

Assignment 2

ITS 413

Internet Technology and Application

Title

Throughput Performance of TCP

By

Group 3

Ms. Watanee Jearanaiwongkul 5122780273

Mr. Chatchapol Lertritkornchai 5122791296

Mr. Phuchong Kheawchaoom 5322040576

Submitted to

Dr. Steven Gordon

Due

Monday 14th February 2011

Table of Content

Title	Pages
Introduction	3
Process: Set up	4
Process: Coding	5 – 6
Result	7 – 10
Graph Result	11 – 14
Discussion	15

Introduction

This assignment asks us to test for the throughput of TCP. The experiment included many alteration of the code, such as length of information, buffer size, transfer rate, delay rate and etc. The experiments also consider being in many different situation, which are Single TCP Session and Multiple TCP Session. The experiments include the usage of iPerf. We are using Ubuntu as our Operating system for the testing process. Since the results show up to be quite different, so we have to do the experiment about 2 – 3 times per test, and get the average result out of it.

Process: Set up

We are using the code of iPerf and altered the information to suite in each situation.

1. We have to shut down the internet, because we are using only 2 computers to connect together.
2. We connect 2 computers using crossing line pluck into the LAN line of each computer.
3. Set up the address of each computer using the command, "sudo ifconfig eth0 100.10.10.1 netmask 255.255.255.0 up"(server), on one of the computer and change the address "100.10.10.2"(client) for the other computer.
4. We then set on of the computer to be the server, by the code, "iperf -s", and we set the other computer to be client by the code, "iperf -c SERVER IP". This is the basic code for server and client. The code will also be change depend on the experiment that we have to measure.
5. We are using the code, which I will tell later on to test for the other process.
6. We do the experiment 2 – 3 times for the correct result and jot down the average of the test.

After we get all the information, we plot down the graph look for the trend and understand it.

Process: Coding

For each experiment, there is different coding, because it has to change according to the information that we want to find.

First category is Application and Protocol Parameter.

First we want to test for the buffer/window size (-w), so the input of client is "iperf -c SERVER IP -w WINDOW SIZE -t TIME". And the server has to be "iperf -s -w WINDOW SIZE -t TIME." In this experiment we will change the buffer size only, so that if the client changes then the server has to stay the same. For example the server's buffer size say the same as change the client's buffer size until it doesn't change, then we change the server's buffer size and start testing with the same number for client again.

Second we want to test for the Length of data written/read by application (-l), so the input of client is "iperf -c SERVER IP -l AMOUNT in K (i.e. 4K) -t TIME". And the server has to be "iperf -s -t TIME". In this experiment we want to see the length of data that is being transfer. We change only the amount in K.

Second category is Network and Link Conditions.

First we want to test for the Link data Rate. So we have to insert the code in for the client only, and the server just stays the same. For this experiment, there are 3 areas that can be altered. "sudo tc qdisc add eth0 root tbf rate SENDING RATE (kbit) latency MAXIMUM TIME(ms) burst SET OF BIT ALLOW TO BE SENT(kb)." When you finish with the experiment you have to type in "sudo tc qdisc del eth0 root", this will reset the code that you have input earlier.

Second we want to find the Delay of data that being transfer. So we input in the information to change the state of client. We input in the code as "sudo tc qdisc add dev eth0 root netem delay RATE (ms) ± TIME (ms)". The code that client have to put afterward is "iperf -c SERVER IP -t TIME". And the code for server is "iperf -s -t TIME". The information that you want to input as the delay time is the average including the ± sign of the time such as if the rate is 100ms and the ± time of 10ms, which make the range to be 90 to 110 ms. When you already finish with the coding, you have to put in the code "sudo tc qdisc del dev eth0 root", this will reset the coding that you have insert earlier.

Third we want to find the packet drop rate, so the code that you have change for the server is "sudo iptables -A INPUT -m statistic -mode random -probability AMOUNT -j DROP" so after we change that you have to put in the basic coding, for client is "iperf -c SERVER IP -t TIME". And for server is "iperf -s -t TIME".

Third category is Multiple TCP Sessions.

First we want to find the Session with 2 TCP, so the code for client is “iperf -c SERVER IP -t TIME -P AMOUNT OF TCP (in this case is 2)”, and the server’s code is “iperf -s -t TIME” The output will be 3 lines, which are input of 2 clients (from the same terminal) and sum line, which show the total bandwidth of 2 clients adding together. For this experiment, you have to do 3 or more times to find the trend and result. There isn’t any more alteration of the code.

Second we want to find the session with 3 TCP, so we use the same code as above and the result will be 4 lines, which are 3 lines of clients, and 1 more line of summation bandwidth. The code for other part is also the same as the 2 TCP.

And last category is Single and Multiple TCP Sessions in Presence of UDP Sessions.

In this experiment, it will be 1 computer being both TCP and UDP, so that means one computer has to open 2 terminals.

First test is 1 TCP and 1 UDP. So the code for client of the TCP is “iperf -c SERVER IP -t TIME”, and for UDP is “iperf -u -c SERVER IP -t TIME”. For the server, TCP coding is “iperf -s -t TIME”, and for UDP is “iperf -u -s -t TIME”. This experiment will test both UDP and TCP at the same time. If it is possible, you have to press for both UDP and TCP at the same time as much as possible and the result will show in each one of them represent TCP or UDP.

Second test is 2 TCP and 2 UDP. The code for server is the same, but the code for client is as following. UDP: “iperf -u -c SERVER IP -t TIME -P AMOUNT (in this case is 2)”. TCP: “iperf -c SERVER IP -t TIME -P AMOUNT (2)”. The result will be in total of 6 lines, 3 on each. 2 lines of each represent the result, and 1 for the total bandwidth of the 2.

EXTRA

BDP or in the other word is “Bandwidth Delay Product”, which can be calculated by Bandwidth * Round Time Trip. We have learn that the Operating system accept our information that we insert such as 20K and *2, so the number is actually 40K. We input in the code as “sudo tc qdisc add dev eth0 root netem delay RATE (ms) ± TIME (ms)”. The code that client have to put afterward is “iperf -c SERVER IP -t TIME”. And the code for server is “iperf -s -w BUFFER SIZE”. From this experiment, we will be able to observe the result of the Bandwidth for each window size.

Result

First category is Application and Protocol Parameter.

1.1 The buffer size with the amount of server at the default rate that we have been calculated is at 85.3 Kbytes.

Server(Kbyte)	Client(Kbyte)	Transfer(Mbyte)	Throughput(Mb/s)	-w
85.3	2	364	76.4	1000
85.3	3.91	360	75.4	2000
85.3	5.86	449	92.7	3000
85.3	9.77	449	94.1	5000
85.3	19.5	449	94.1	10000
85.3	39.1	450	94.1	20000
85.3	58.6	449	94.2	30000
85.3	78.1	450	94.2	40000
85.3	97.7	449	94.2	50000
85.3	117	450	94.2	60000
85.3	137	450	94.2	70000
85.3	156	450	94.2	80000

1.2 The buffer size with the amount of client at the default rate that we have been calculated is at 16 Kbytes.

Client(Kbyte)	Server(Kbyte)	Transfer(Mbyte)	Throughput(Mb/s)	-w
16	1.95	230	48.3	1000
16	3.91	350	59.5	2000
16	5.86	390	59.6	3000
16	7.81	415	86.4	4000
16	9.77	447	93.8	5000
16	19.5	449	94.1	10000
16	39.1	449	94.2	20000
16	58.6	449	94.2	30000
16	78.1	449	94.2	40000
16	97.7	449	94.2	50000
16	117	449	94.2	60000

2. The Length of data written/read by application.

Length	Time	Bandwidth
2	40	94.2
4	40	94.2
6	40	94.2
8	40	94.2
10	40	94.2
12	40	94.2
14	40	94.2
16	40	94.2
18	40	94.2
20	40	94.2
22	40	94.2

Second category is Network and Link Conditions.

1. Link Data rate

Rate(Kbit) Latency(ms)	Throughput(Kb/s)	Throughput(Kb/s)	Throughput(Kb/s)
	50	100	150
30	28.8	26.8	25.7
50	46.2	48.2	46.2
80	76	77	77
100	95.6	96.1	96.1
300	278	269	279
500	476	480	475
1000	956	930	951
2000	1905	1895	1906
3000	2820	2720	2615
5000	4788	4750	4795
8000	7486	7642	7550
10000	9560	9620	9610

2. Delay of Data

Time	Throughput (Mb/s)
100	85.2
200	69.3
300	43.3
400	29.2
500	22.2

3. Packet Drop rate

Drop	Throughput (Mb/s)
1%	66.7
2%	62.5
3%	40.3
4%	10.7
5%	4.77

Third category is Multiple TCP Sessions.

1. 2 of the TCP at the interval of 60 and 65

Interval	Transfer	Throughput(Mb/s)
60	337	46.7
	342	47.5
sum	679	94.2
65	365	46.7
	371	47.5
sum	735	94.2

2. 3 of the TCP at the interval of 60 and 65

Interval	Transfer	Throughput(Mb/s)
60	227	34.4
	227	34.4
	227	31.4
sum	680	94.2
65	227	34.4
	227	34.4
	227	31.4
sum	680	94.2

And last category is Single and Multiple TCP Sessions in Presence of UDP Sessions.

1. One of the TCP and one of the UDP

	Time(sec)	Throughput(Mb/s)
TCP	60	93.2
UDP	65	1.05
TCP	60	93.1
UDP	65	1.05

2. Two of the TCP and two of the UDP

	Time(sec)	Throughput(Mb/s)
UDP	65	1.05
UDP	65	1.05
TCP	60	45.9
TCP	60	46.3

EXTRA

Since we set the delay to be 80 so the value is 160(they accept as RTT), and we want to find BDP so we use the formula: $[160\text{ms} * 100\text{Mb/s (bandwidth of the computer)}] / 8$ (for turning it to be Byte), and it turn out to be 2000 KB. Dr. Steven wants us to find the value of 1000KB, but we did 2000KB. We are misunderstanding the instruction.

Delay	TCP window size	Throughput (Mb/s)
80 ms	10	0.549
80 ms	20	1.1
80 ms	30	1.58
80 ms	40	2.3
80 ms	50	2.94
80 ms	80	4.66
80 ms	100	5.83
80 ms	200	12.5
80 ms	256	14.3

Result

First category is Application and Protocol Parameter.

1. When the buffer size is increased, the throughput will be increased because the space for holding data in application layer is expanded then the data easily flow from application layer to TCP layer.

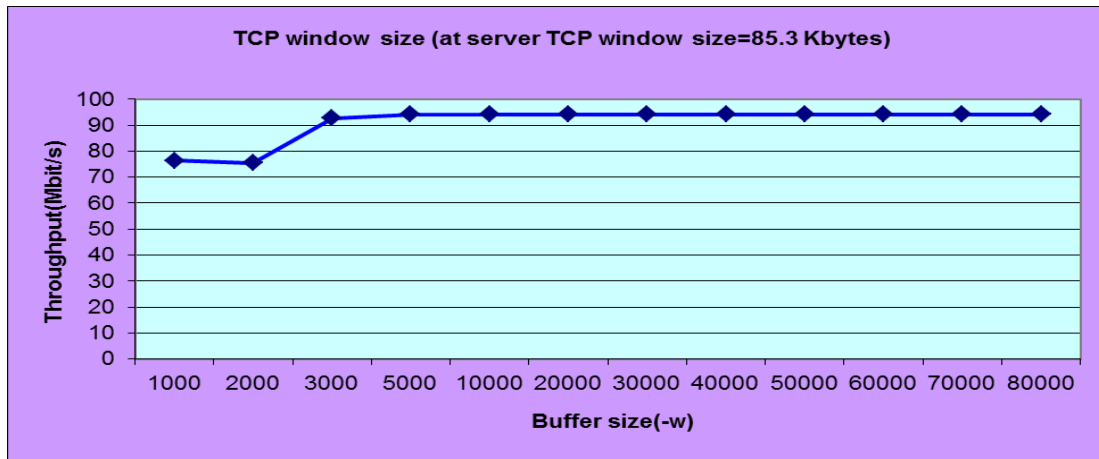


Figure 1: TCP window size (at sever TCP window size=85.3 Kbytes)

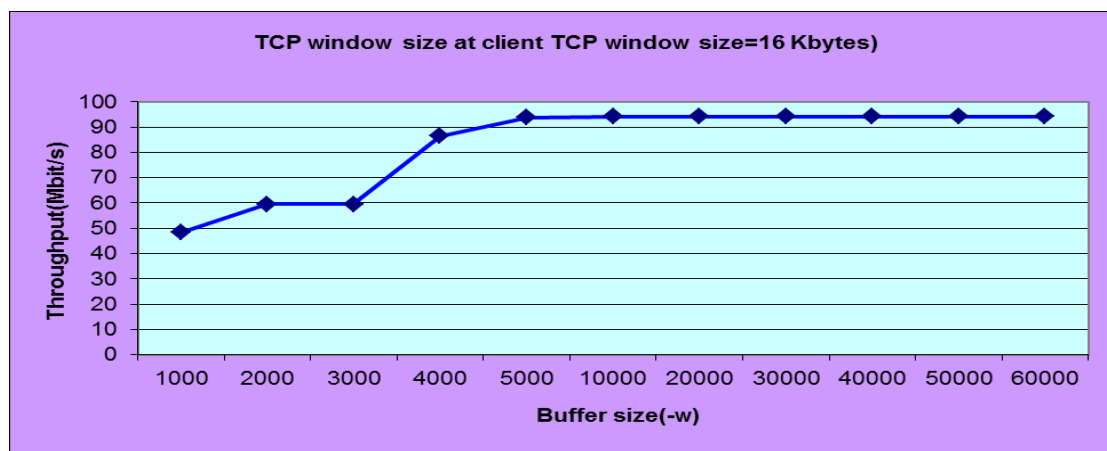


Figure 2: TCP window size(at client TCP window size = 16 Kbytes)

From Figure 1 and Figure 2, the result is shown that the buffer size of sever is more effect than buffer size of client because the default value of server's buffer size is relatively larger than the default value of client's buffer size. Thus, the throughput is more decreased when compares between the reduce server's buffer size and the reduce client buffer size.

- The number is a straight line, because throughput doesn't cause any effect by read or write.

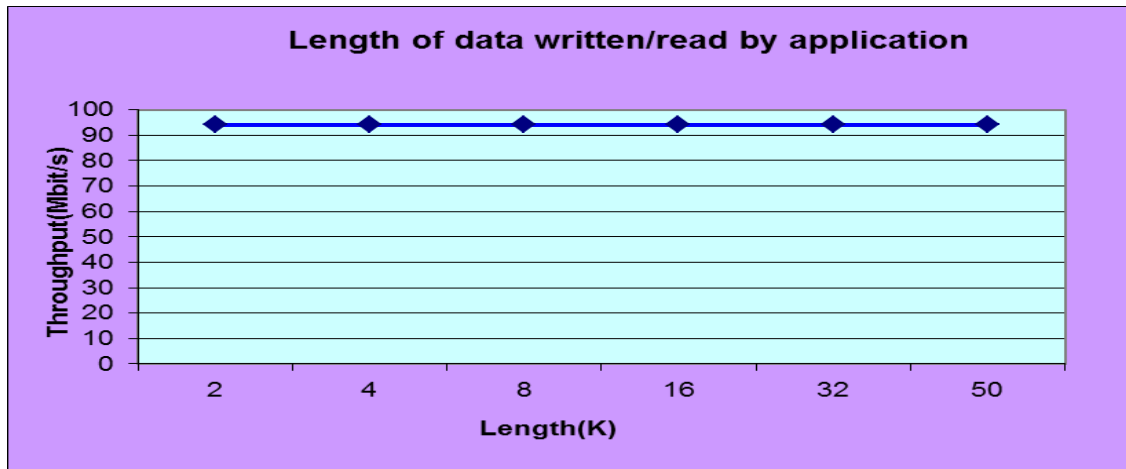


Figure 3: Length of data written/read by application

From Figure 3, the throughput trend to be constant due to the bandwidth delay product is quite small so the written or read in buffer by application are not effect to throughput.

Second category is Network and Link Conditions

- In Figure 4, the throughput is directly control by sending rate in –tc command so the whatever the value of latency is changed, It does not effect to throughput.

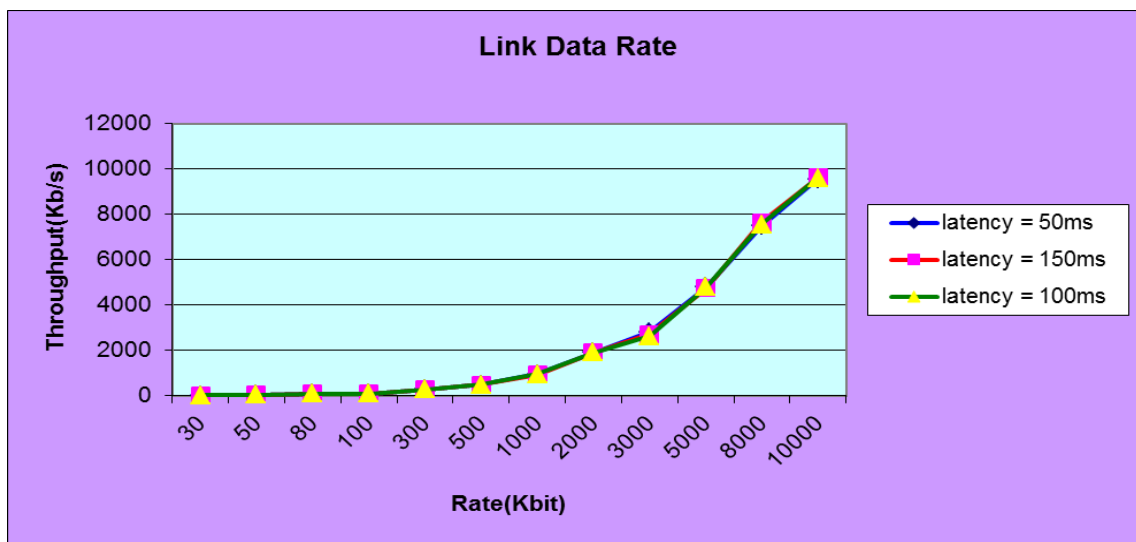


Figure 4: Link Data Rate

- Figure 5 is shown that time delay is directly effect to throughput. When the time delay is increasing, the throughput will be decreasing like a linear fashion. It occur from the TCP have the timer to count time for ack. If the client did not receive ACK back from server in time, it will retransmit until receive ACK. Thus, the throughput is dropped.

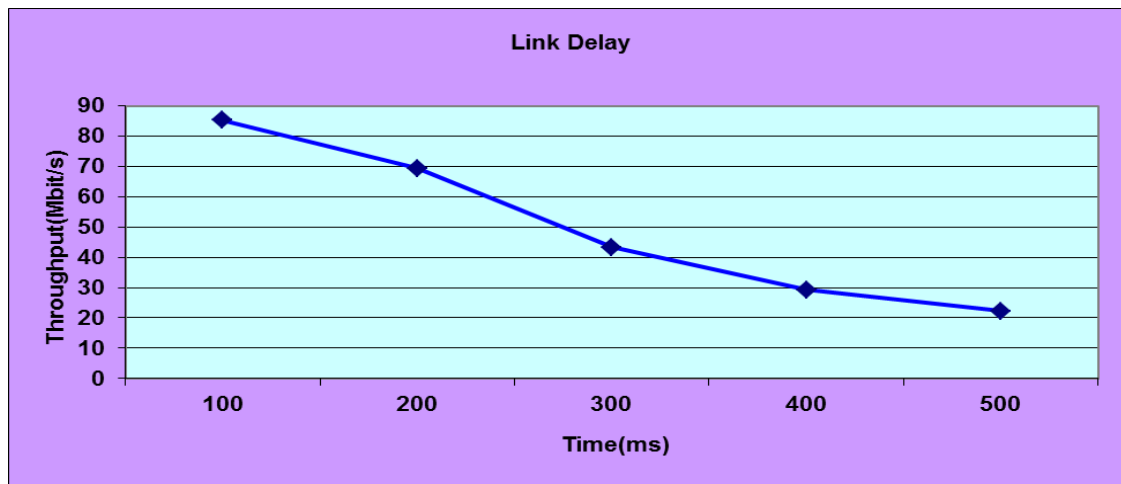
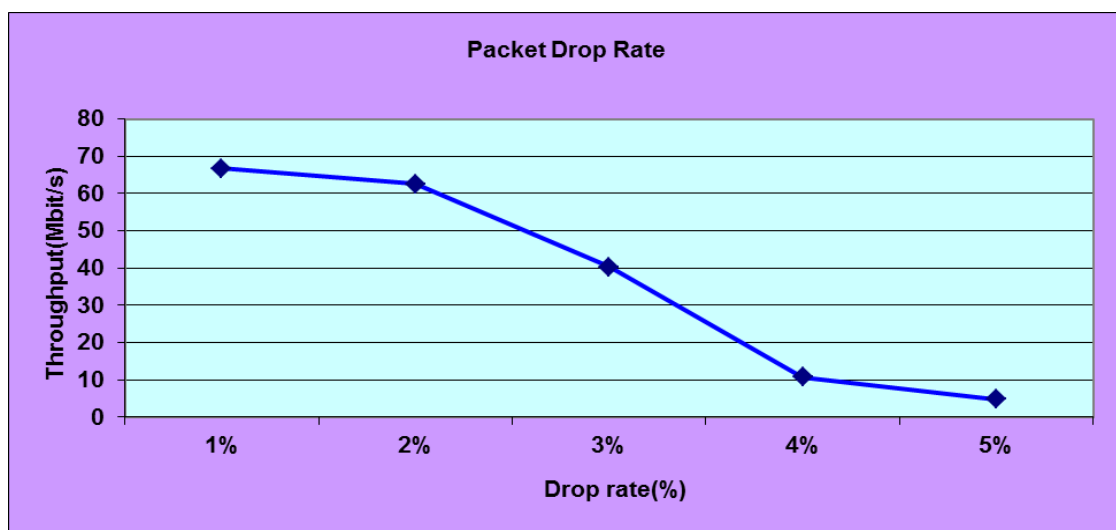


Figure 5: Link Delay

- Packet Drop rate: rate of dropping increase, causing the data cannot be transfer to and receive from, because the server will drop unnecessary information and I will just retransmit the same information over and over again until it reach the destination.

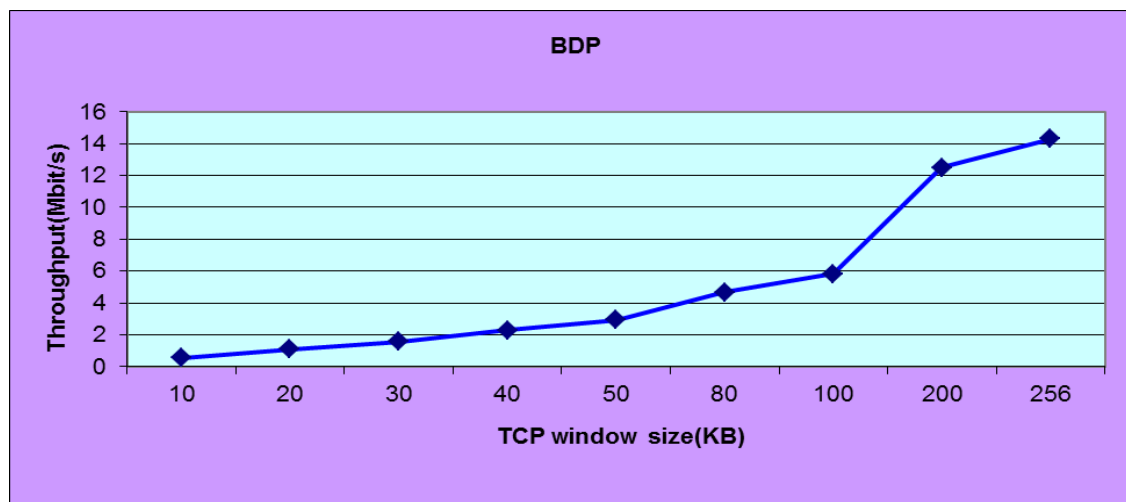


Third category is Multiple TCP Sessions and last category is Single and Multiple TCP Sessions in Presence of UDP Sessions.

From many trail, we got the same result, so we will not plot the graph. The information turn out to be like this, because the amount of summation is the same for all time, so when we did 2 TCP or 3 TCP means it just shared out from that maximum amount. The total amount is 94.2. For the 2 TCP means that the total has to be $94.2/2$ and it doesn't matter if it is divided equally or not. The 3 TCP mean that it will be divided by 3. For the 1 TCP and 1 UDP, The value is about 1 Mb for UDP and the rest is for TCP. For 2 TCP and 2 UDP, the UDP will be approximately 1 Mb per channel, and the rest will be divided to be for TCP equally. As you can see that it assumes that the UDP is more TCP. Overall information for 2 TCP and 2 UDP will be in total of 94.2 Mb/s. From the experiment we got that the UDP will always be the same, which is 1.05 Mb/s.

EXTRA

As the amount of size that can be transfer is increase, so the amounts that is successfully sent to or receive from is increase as well.



Discussion

The result we got helps us to understand more of the relation between each of the input, or rate that can be altered. The results that we got are similarly as other group, there is some part that is not exactly the same as other group, because it could cause from the different type of network and computer usage. Some of the information we got cannot be turn into graph data, because it turns out to be the same summation even though we try 2-3 times with it.