ITS 413 Internet Technologies and Applications

Assignment: Phase 3Report

By:

Juntima	Nawilaijaroen	5222781403
Nicharee	Boonfaeng	5222782237
Kampanat	Sarasri	5222790511

Date:

March 28, 2012

By submitting this report all members of the group listed above agree that each member has contributed approximately equal amounts to designing and performing experiments, as well as to preparing this report. All members agree that this report accurately reflects the experiments conducted by the group members, and is their own work (not works of other groups).

Sirindhorn International Institute of Technology

Thammasat University

Aims

- To be able to apply knowledge that was taught in class and laboratory to setup, configure and control a performance of sending and receiving data through a network.
- To see a performance of TCP and UDP and can indicate which one is better or worse by analyse a result of experiments
- To see a performance of TCP and UDP when some factors were controlled.

Network Diagram

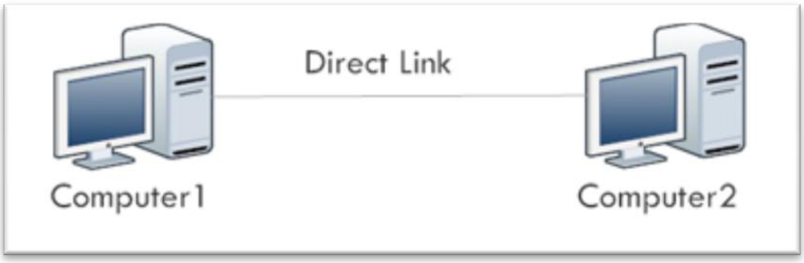

Figure 1 – Direct link between 2 computers by using cross wire.

Figure 1 shows network topology which was using in experiment 1. It is a connection between 2 computers by using cross wire. The specification of computer and wire were described in detail in experiment 1 section.

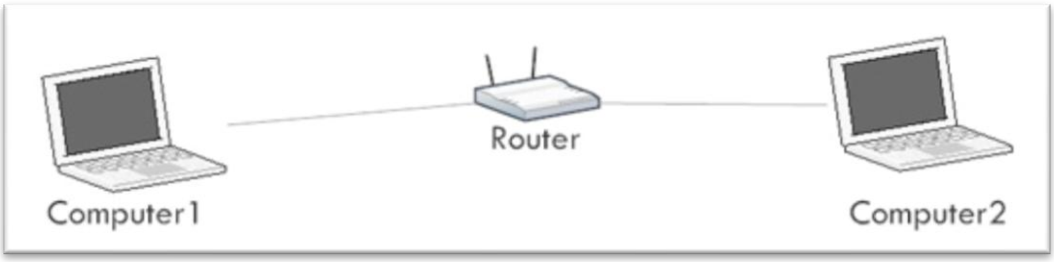

Figure 2 – Direct link between 2 laptop computers by using router.

Figure 2 shows network topology which was using in experiment no.2 - 4. Due to the limitation of equipment, we couldn't find a cross wire to use. So we've router and connect two computers to router by Unshielded Twisted Pair (UTP) instead of a cross wire.

Equipment Specifications

Name	Computer1 (experiment 1 - server)
OS	Ubuntu 11.10
Memory	4.00 GB
Processor	Intel(R) Core(TM) i5 CPU 760 @ 2.80 GHz 2.80 GHz
Network Interface	Realtek PCIe GBE Family Controller
Network Interface Driver	7.49.927.2011

Name	Computer2 (experiment 1 - client)
OS	Ubuntu 11.10
Memory	4.00 GB
Processor	Intel(R) Core(TM) i5 CPU 760 @ 2.80 GHz 2.80 GHz
Network Interface	Realtek PCIe GBE Family Controller
Network Interface Driver	7.49.927.2011

Name	Computer1 (experiment 2-4 - server)
OS	Ubuntu 11.10
Memory	4.00 GB DDR2
Processor	Intel(R) Core(TM) i3 CPU M330 @ 2.13 GHz 2.13 GHz
Network Interface	Atheros AR9285 Wireless network Adapter Marvell Yukon 88E8059 PCI-E Gigabit Ethernet Controller
Network Interface Driver	8.0.0.238 11.22.4.3

Name	Computer2 (experiment 2-4 - client)
OS	Ubuntu 11.10
Memory	4.00 GB
Processor	Intel(R) Core(TM) Core2Duo CPU M330 @ 2.26 GHz 2.26 GHz
Network Interface	Atheros AR9285 Wireless network Adapter Marvell Yukon 88E8059 PCI-E Gigabit Ethernet Controller
Network Interface Driver	8.0.0.238 11.22.4.3

Name	D-Link , DSL - 2730U
Device Interface	RJ-11 ADSL Port
Wireless LAN	802.11 b/g/n
Frequency Range	2.4 GHz – 2.484 GHz

Parameters

Name	Default Value
Bandwidth	1 Mb/s
Window size	16 Kb
Drop rate	0%

Experiments

The experiment was divided into 4 parts. Each part was set in different scenario and different equipment. The method and scenario of each experiment will be described in detail in next section.

Experiment 1 – TCP vs. UDP

Experiment 1 focused on the different between User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) in actual data transferring major. Multiple sizes of data from 10MB to 1000MB were transferred between two computer to measure the performance of UDP and TCP. We've run a multiple test to see whether those factor affect UDP and TCP performance or not and how. The other matrix which we didn't interest in is bandwidth and window size. These factors were set to default.

Parameters

1. Data size - 10MB ~ 1000MB
2. Time – 10s
3. Protocol – TCP and UDP

Experiment Scenario and Procedure

- For TCP.

  - Set up 2 computers with the same configuration and operation system. Make sure that it is running on the same hardware and setting.
  - Set up and configure IPv4 address on a PCs, using command

```
On 1st host  -ifconfig eth0 1.1.1.1 netmask 255.255.255.0
On 2nd host  -ifconfig eth0 1.1.1.2 netmask 255.255.255.0
```

  - Use ping command to ensure that the network connection is working properly.
  - Make sure there is no other application running in the background process.
  - Setup and configure server on computer1 by using iperf command

```
-iperf -s
```

  - Setup and configure computer2 and start an experiment by using iperf command

```
iperf -c 1.1.1.1 -t 10s
```

  - Monitor the result on the server and client and make sure there is no problem on network connection and data transferring.
  - Run the real test and collect data when there is no problem on network connection and data transferring by running the same command on server and running bash script on windows size

```
#!/bin/bash
counter=10
m='S'
f=1
for i in {1..5}
  do
    for i in {1..13}
       do
          echo "================= Time: "${counter}${m}"=========="
          >> output${f}.txt
          iperf -c 1.1.1.1 -t ${counter}
          >> output${f}.txt
          sleep 5s
          let counter=counter+10
      done
   echo "File "${f}" successfully generated"
   let counter=10
   let  f=f+1
done
```

- For UDP

  - Set up 2 computers with the same configuration and operation system. Make sure that it is running on the same hardware and setting.
  - Set up and configure IPv4 address on a PCs, using command

```
On 1st host  -ifconfig eth0 1.1.1.1 netmask 255.255.255.0
On 2nd host  -ifconfig eth0 1.1.1.2 netmask 255.255.255.0
```

  - Use ping command to ensure that the network connection is working properly.
  - Make sure that there is no other application running in the background process.
  - Setup and configure server on computer1 by using iperf command

```
-iperf -s
```

- Setup and configure computer2 and start an experiment by using iperf command

```
iperf -c 1.1.1.1 -t 10s
```

- Monitor the result on the server and client and make sure there is no problem on network connection and data transferring.
- Run the real test and collect data when there is no problem on network connection and data transferring by running the same command on server and running bash script on windows size

```
#!/bin/bash
counter=10
m='S'
f=1
for i in {1..5}
  do
    for i in {1..13}
       do
         echo "================= Time: "${counter}${m}"=========="
         >> output${f}.txt
         iperf -c 1.1.1.1 -t ${counter}
         >> output${f}.txt
         sleep 5s
         let counter=counter+10
      done
   echo "File "${f}" successfully generated"
   let counter=10
   let  f=f+1
done
```

Result

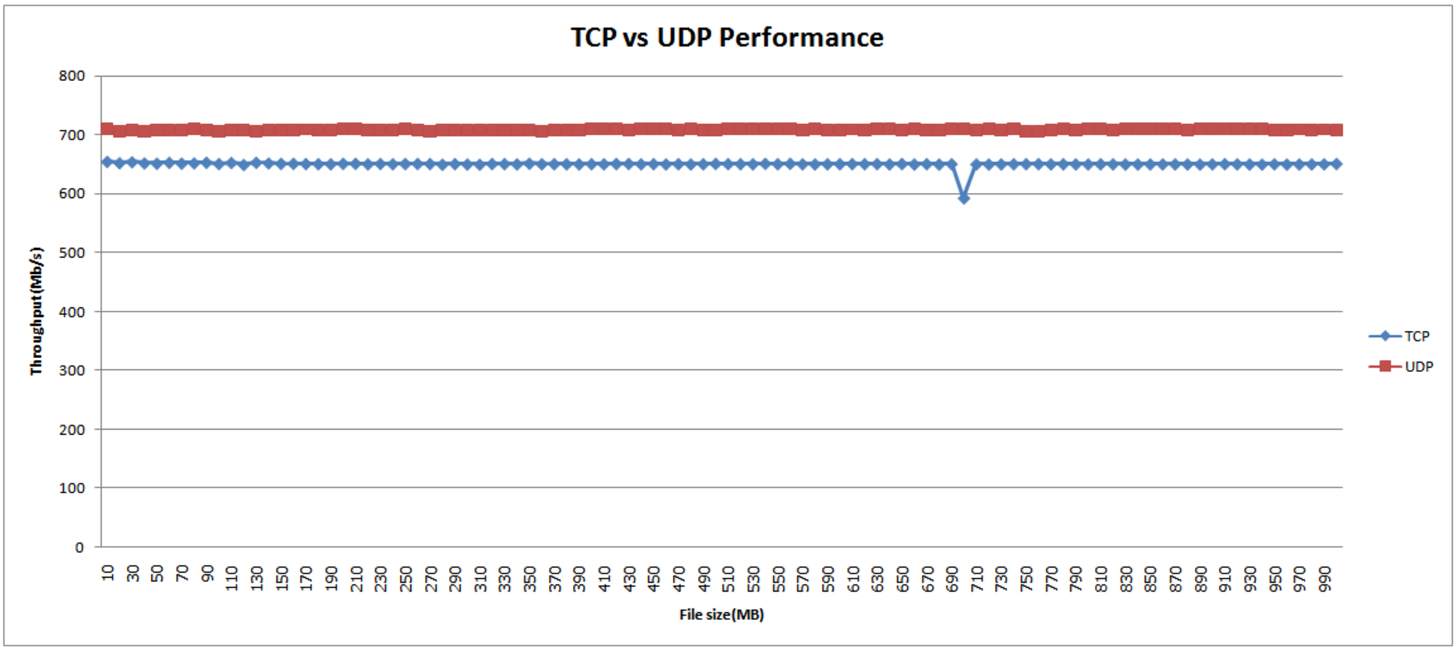

Figure 1 – Throughput of TCP compare to UDP

Note – Table of result can be seen in Appendix section

Conclusion

Figure 1 shows the comparison of throughput between TCP and UDP. From this, we can conclude that UDP throughput is obviously higher than TCP.

While we have same number of file size, we can see that UDP has better performance than TCP. Since UDP has smaller header than TCP, therefore time required in sending is smaller. Also, error correction function and congestion function in TCP caused time required in sending become bigger and performance got lower.

Experiment 2 – Impact of packet drop on TCP performance

The aim of this experiment is to measure the performance of Transmission Control Protocol (TCP) when some of data loss during data transmission. Variance percentage of data being drop is set from 1% to 100% by using **iptable** before data being sent between two computers.

Parameters

1. Drop rate - 0.5% - 100%
2. Time – 60s
3. Protocol – TCP

Experiment Scenario and Procedure

- For TCP.

    - Set up 2 computers with the same configuration and operation system. Make sure that it is running on the same hardware and setting.
    - Set up and configure IPv4 address on a PCs, using command

```
On 1st host   -ifconfig eth0 1.1.1.1 netmask 255.255.255.0
On 2nd host  -ifconfig eth0 1.1.1.2 netmask 255.255.255.0
```

- Use ping command to ensure that the network connection is working properly.
- Make sure there is no other application running in the background process.
- Setup and configure server on host-1(Server) by using following command
    - Computer1 run the server

```
-iperf -s
```

To start the server with TCP protocol and using default configuration.

- Computer2 setting % of packet drop.

```
- sudo iptables -A INPUT -m statistic --mode random --probability 0.05  -j DROPD
```

To create a rule for **iptable** to randomly drop packet(s) by limited drop rate to 5%.

- Setup and configure a client and do a pretest testing the on Computer2(client) by using iperf command

```
iperf -c 1.1.1.1  -t 60s
```

- Monitor the result on the server and client and make sure there have no problem on network connection and data transferring.
- Conduct an experiment by initially set drop rate to 5% and increasing by 5% at each time of experiment until drop rate reach 100% and collect data when there is no problem on network connection and data transferring.

Result

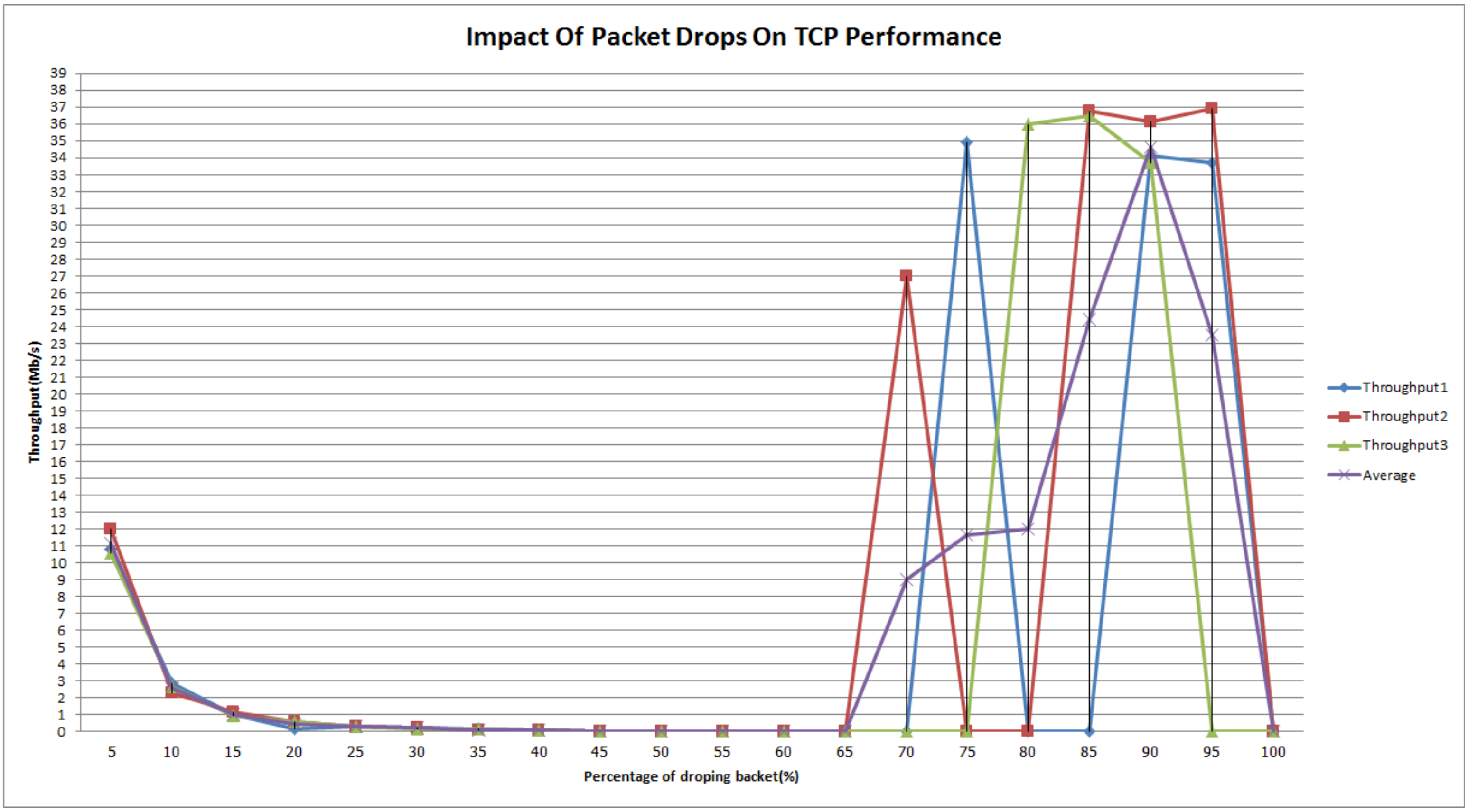

Figure 3 – The throughput of TCP when packets were dropped.

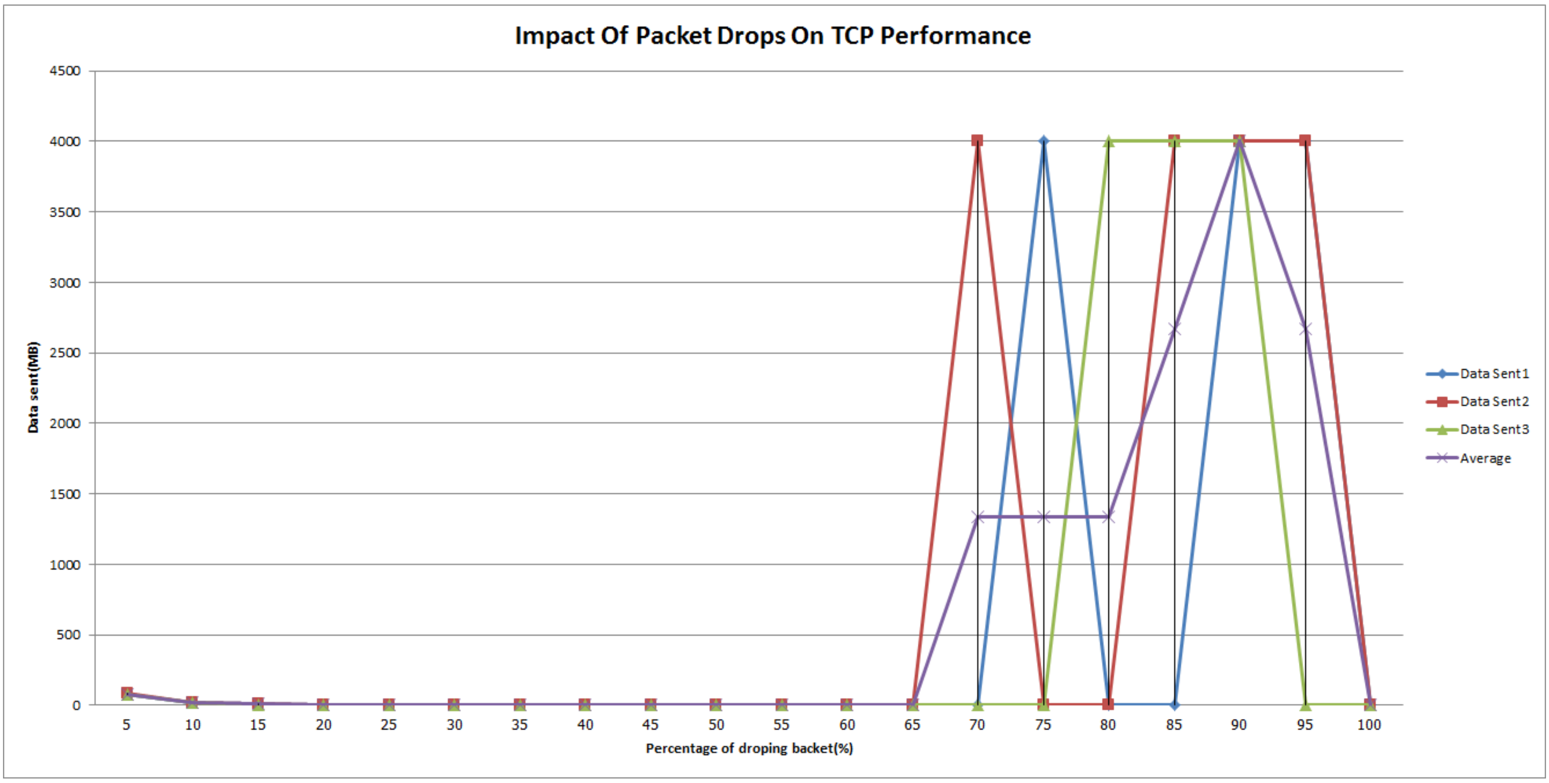

Figure 4 – The throughput of TCP when packets were dropped.

Note – Table of result can be seen in Appendix section

Conclusion

This experiment shows that we've dropped all packets that were sent to the server by different drop rate in each time of running the experiment.

In figure 3, the graph shows that drop rate effect throughput of TCP. The bigger the drop rate, the lower the throughput.

In figure 4 shows that if the drop rate is increasing, the number of packet that reached the server is lower even we have same file size and conduct the experiment with same length of time.

However, drop packet function that occurred in server side caused server was not able to send ACK back to client to tell that server received all packets. This problem effected to the amount of packet loss. The amount of packet loss will get higher. Congestion control in client side will reduce its speed in sending packet of data to the server to reduce the number of packet loss. It assumes that server's buffer is full which result in increasing of time sending a single file that got higher.

The result of an experiment after drop rate has reached 65 percent has gone wrong due to the problem of terminal. We couldn't extract a real value of data sent there's a calculation exception occurred in a terminal.

Experiment 3 – Impact of Window Size on TCP Performance

This experiment mentioned on the impact of TCP window size on Transmission Control Protocol (TCP) performance. The experiment conducted by sending 100MB of data between two computers by using various sizes of TCP window. The experiment was conducted 3 times to ensure the result. Therefore, we will use the average of those 3 results to create performance graph. To focus on how TCP windows size effect the performance, the other parameter was set to default.

Parameters

1. File size - 100MB
2. Window size – 1Kb – 256 Kb
3. Protocol – TCP

Experiment Scenario and Procedure

- Set up 2 computers with the same configuration and operation system. Make sure that it is running on the same hardware and setting.
- Set up and configure IPv4 address on a PCs, using command

```
On 1st host   -ifconfig eth0 1.1.1.1 netmask 255.255.255.0
On 2nd host  -ifconfig eth0 1.1.1.2 netmask 255.255.255.0
```

- Use ping command to ensure that the network connection is working properly.
- Make sure there is no other application running in the background process.
- Setup and configure server on computer1(server) by using **iperf** command to setup the server with the 1k TCP windows size.

```
-iperf -s
```

- Setup and configure a client computer and conduct an experiment on computer2(client) by using iperf command to send data to the server with 100M of data size.

```
iperf -c 1.1.1.1  -n 100M
```

- Monitor the result on the server and client and make sure there have no problem on network connection and data transferring.
- Conduct an experiment by initially set the window size to 1KB and increasing by 1KB until window size is more than 30KB. Then, increased the window size by 10KB at each time of experiment until 100KB. Finally, increased the window size by 20KB until the window size reached its limitation, 256KB. Data will be collected when there is no problem on network connection and data transferring.

Result

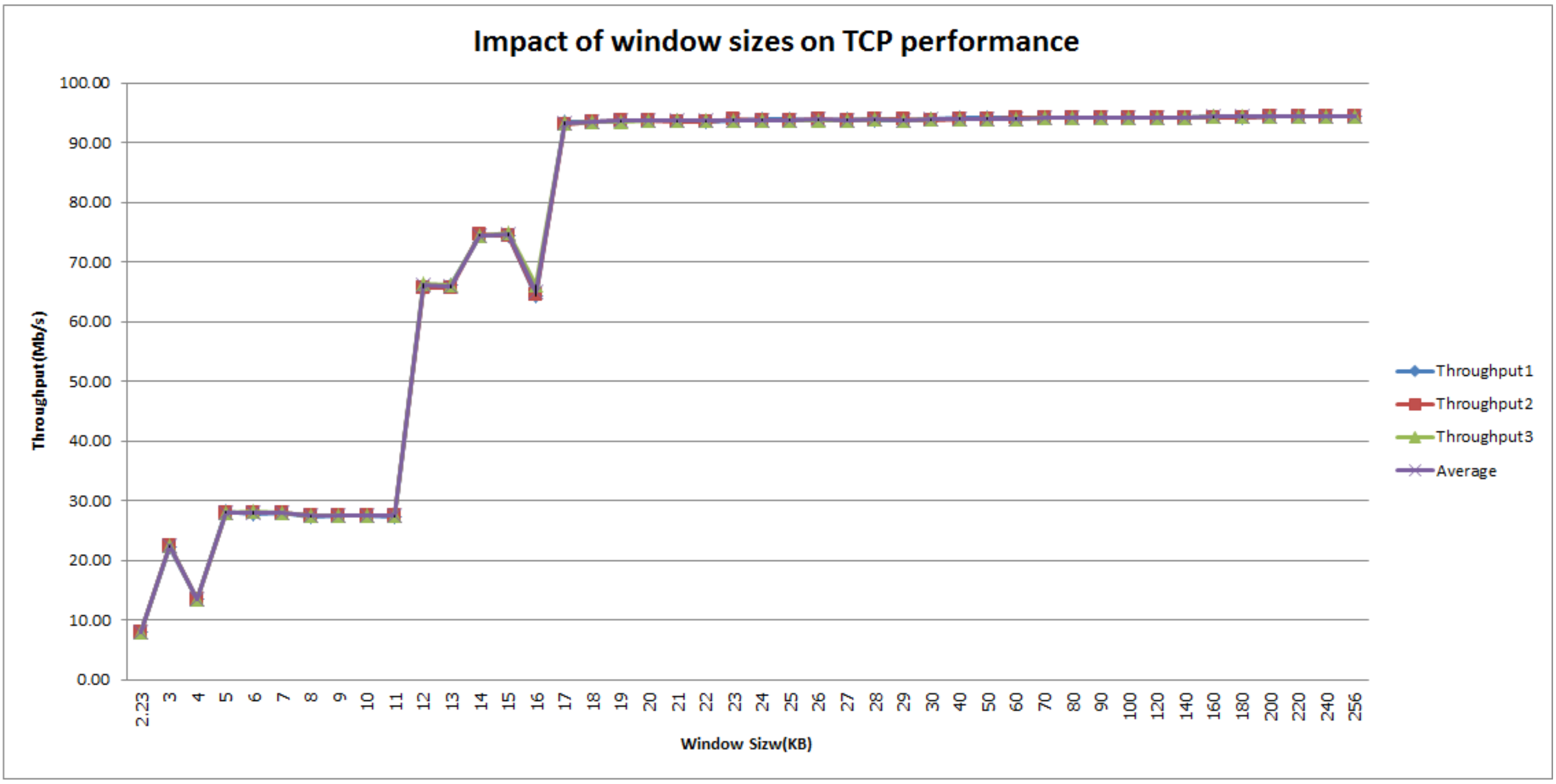

Figure 5 – Throughput of TCP performance by increasing window's size

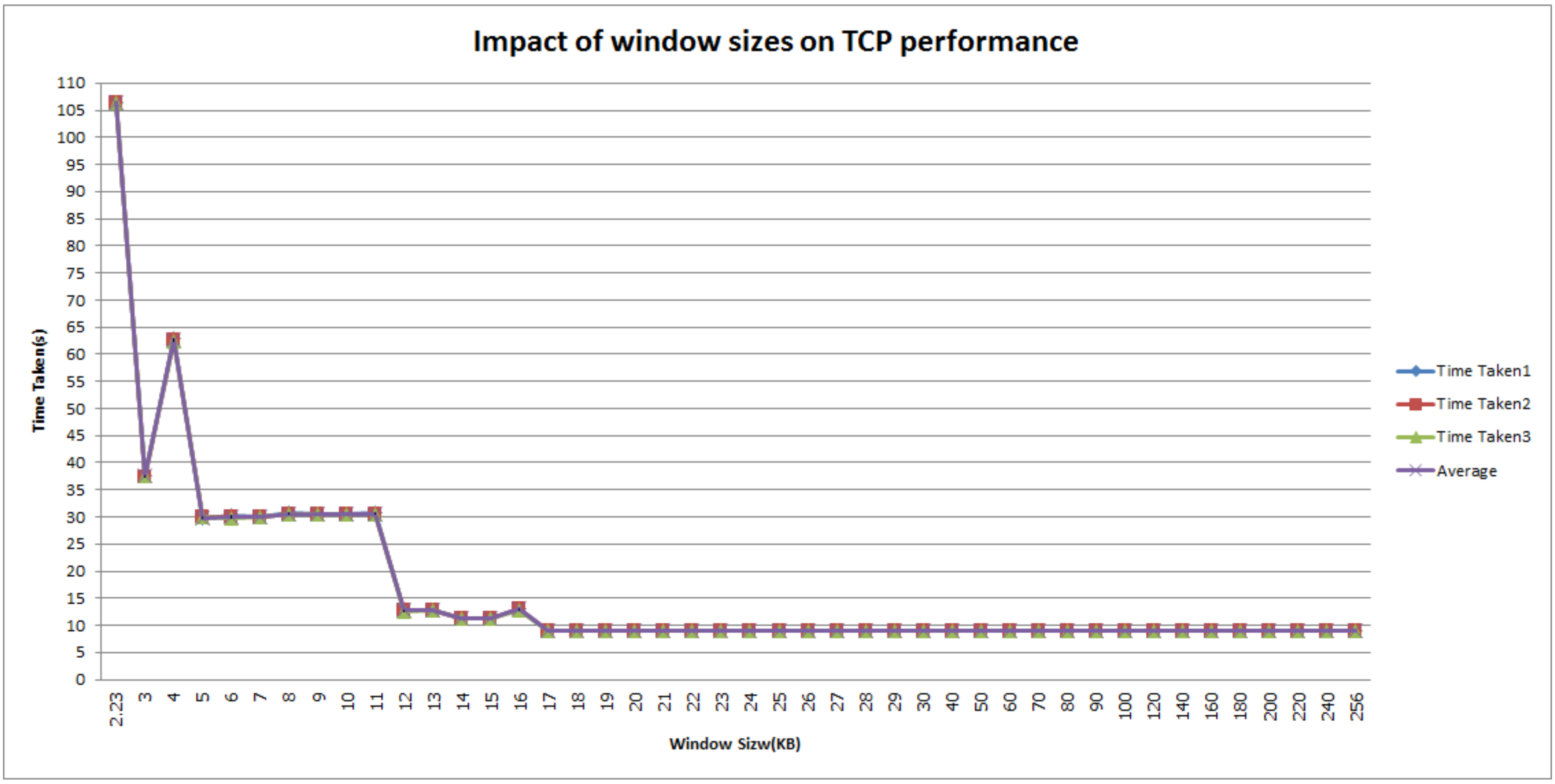

Figure 6 – Time taken in sending same file size but different window's size

Conclusion

According to figure 5, when window size was increased the throughput was getting higher almost exponentially (when there's no interference). By increasing window size, we could increase number of packet at each time of sending and receiving file. A single file could be transfer faster.

Disadvantage of increasing window size occurred when the window size is too big, packet size that was sent at each time might be too big. There might be a chance that buffer is not enough, congestion will be occurred. And follow by packet loss.

Experiment 4 – TCP and UDP Session

This experiment aims to study on fairness of UDP and TCP performance. The value of UDP bandwidth was set to 1 Mb/s.

Before we created a UDP session, we've run a simple test to see the throughput of TCP session. The result shows throughput of TCP session is equal to 94.0 Mb/s. Later on, we introduced a UDP session with bandwidth of 1Mb/s. Then, we've increased number of UDP session from 1 to 10 sessions.

From the scenario mentioned above, we would like to see how fairness of UDP on TCP is when there're multiple sessions sending on the same wire. According to the theory, we assumed that the TCP throughput would decrease only 1Mb/s (no. of throughput equals to 93 Mb/s.) The experiment was conducted to see whether there's a change of TCP throughput from an ideal throughput or not.

Parameters

1. TCP Bandwidth Size :- couldn't control
2. UDP Bandwidth Size :- 1 Mb/s for each session
3. No. of TCP Session – 1
4. No. of UDP Session – 0 – 10 sessions
5. Time – 60s
6. Protocol – TCP and UDP

Experiment Scenario and Procedure

- Set up 2 computers with the same configuration and operation system. Make sure that it is running on the same hardware and setting.
- Set up and configure IPv4 address on a PCs, using command

```
On 1st host   -ifconfig eth0 1.1.1.1 netmask 255.255.255.0
On 2nd host  -ifconfig eth0 1.1.1.2 netmask 255.255.255.0
```

- Use ping command to ensure that the network connection is working properly.
- Make sure there is no other application running in the background process.
- Setup and configure Computer1 and Computer2 to be a server by using iperf command

```
-iperf -s
```

- Setup and configure a client and do a pretest testing the on host-2(client) by using iperf command

```
- iperf –c 1.1.1.1 –t 60 on Computer1 – 1st Terminal
- iperf –c 1.1.1.1 –u –t 60 –b 1M on  Computer1 – 2nd Terminal
```

- Monitor the result on the server and client and make sure there have no problem on network connection and data transferring.
- Conduct an experiment by a method that was already describe above and collect data when there is no problem on network connection and data transferring.
- The experiment was to be conducted 10 times. By each time of experiment, we've open one more terminal to run one more UDP session.

Result

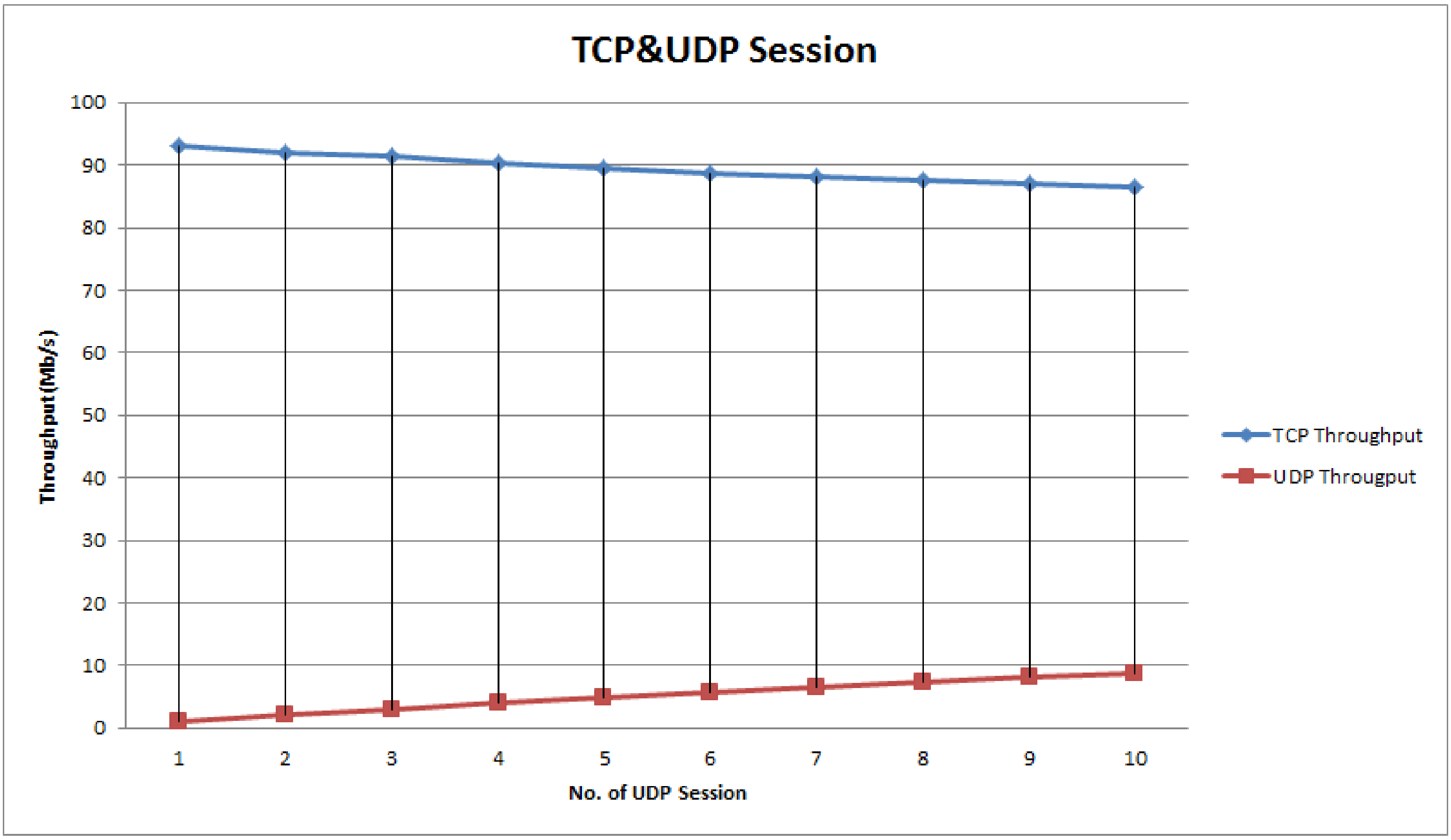

Figure 7 – Throughput of TCP and UDP session.

Conclusion

According figure 7, the TCP throughput has decreased from 94.0 Mb/s (when there's no UDP session) to ~86.0 Mb/s (10 UDP sessions were sending across the same wire). By this result, we assume that there's fairness between TCP and UDP performance since the TCP throughput decreased by 1 Mb/s when we open one more UDP session that limited bandwidth size to 1 Mb/s.