

# ITS 413 Internet Technologies and Applications

---

## *Assignment: Phase 3: Performance Analysis of Transport Protocols over Wireless LAN and Ethernet*

By:

Phattawat Samervongs 5222792426

Chanakan Horkaew 5222800179

Ittipat Nimawarodom 5222792236

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

## *Specifications:*

### Server

Dell N5010

WLAN: Dell wireless 1501 b/g/n Half Mini Card

LAN: Realtek RTL8102E Family Fast Ethernet NIC

OS: Ubuntu

### Client

ASUS A42J

WLAN: Atheros AR9285 Wireless Network Adapter

LAN: Jmicron PCI Express Gigabit Ethernet Adapter

OS: Ubuntu

### Router

WRT54GL Wireless Broadband Router 802.11b/g up to 54Mbps with version Backfire 10.03.1 with the Broadcom Linux 2.4 kernel

### Cable

Cross over cable 100mbps

### Software

-Iperf

-Iptables

-TC

## *Default Parameters:*

Parameters	Default values
RTS/CTS Threshold	250 Bytes
Channel	3
MTU	1500
Protocal used	TCP and UDP
LAN CABLE	CAT5E
Distance	1M

*Network Diagram:*

**Wired Network**



**Wireless Network**



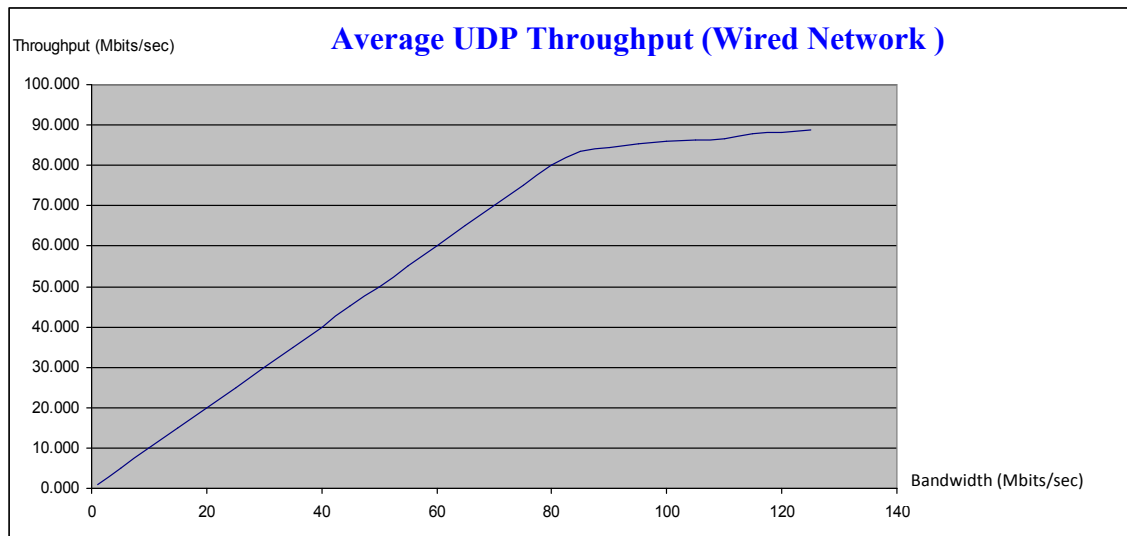


This is what we got from the experiment.

### Wired Network

**TCP:** The average throughput is 94.5 Mb/s because of we can't specify the sending rate in TCP protocol so all the values that we got are very close to each others

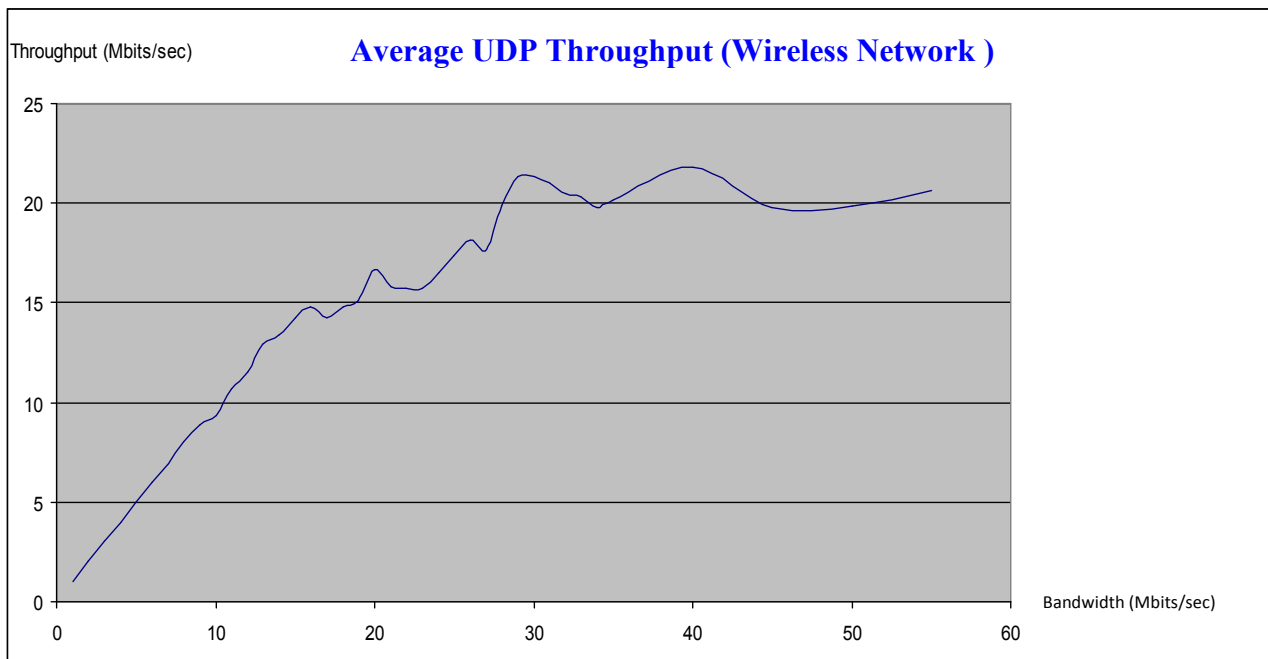
### **UDP:**



## Wireless Network

**TCP:** The average throughput is 14.4 Mb/s because of we can't specify the sending rate in TCP protocol so all the values that we got are very close to each others

**UDP:**



**Summary:** TCP and UDP wired network have constant throughput as the bandwidth increases the throughput is also. However after the bandwidth reached to 95 Mb/s, the throughput seems to reach the highest. For wireless network it similar to wired network but the throughput is more vary.

## 1.2 Delay

Afterwards we apply delay command to the terminal by using this command:

```
Sudo tc qdisc add dev XXXX root netem delay YYms
```

Where XXXX = Network devices for ex. eth0

YY = Delay time for ex. 100ms

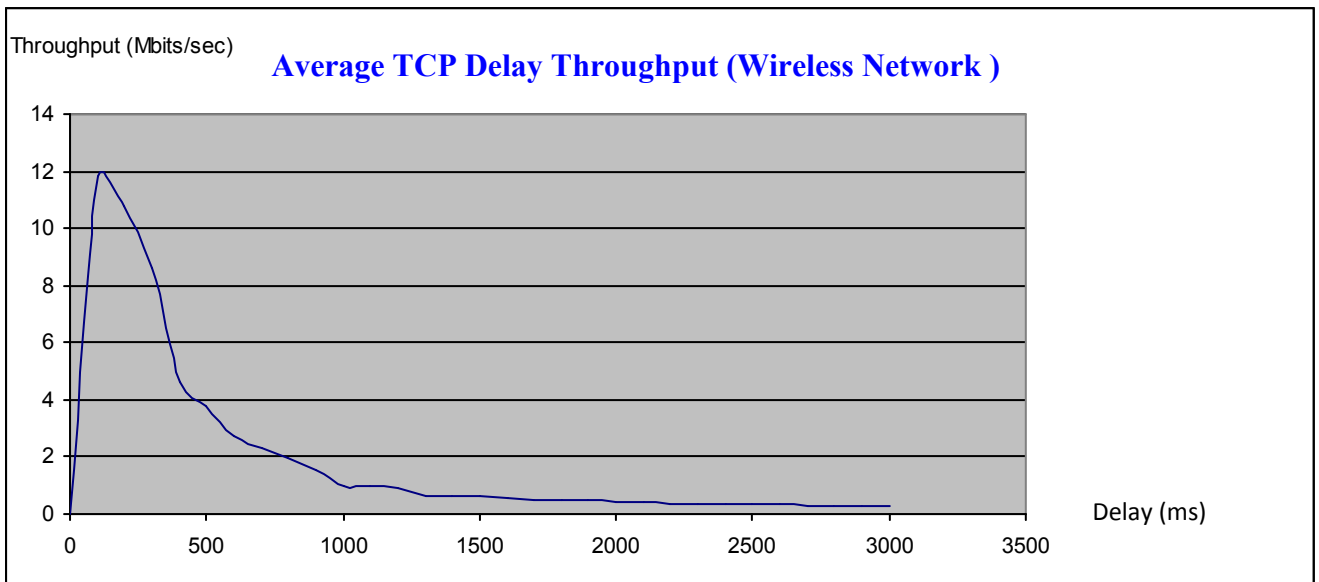
### Parameters

Parameters	Default values
Sending rate	54.2Mb/s
Delay	Increase 100ms every times

This is the result from the test.

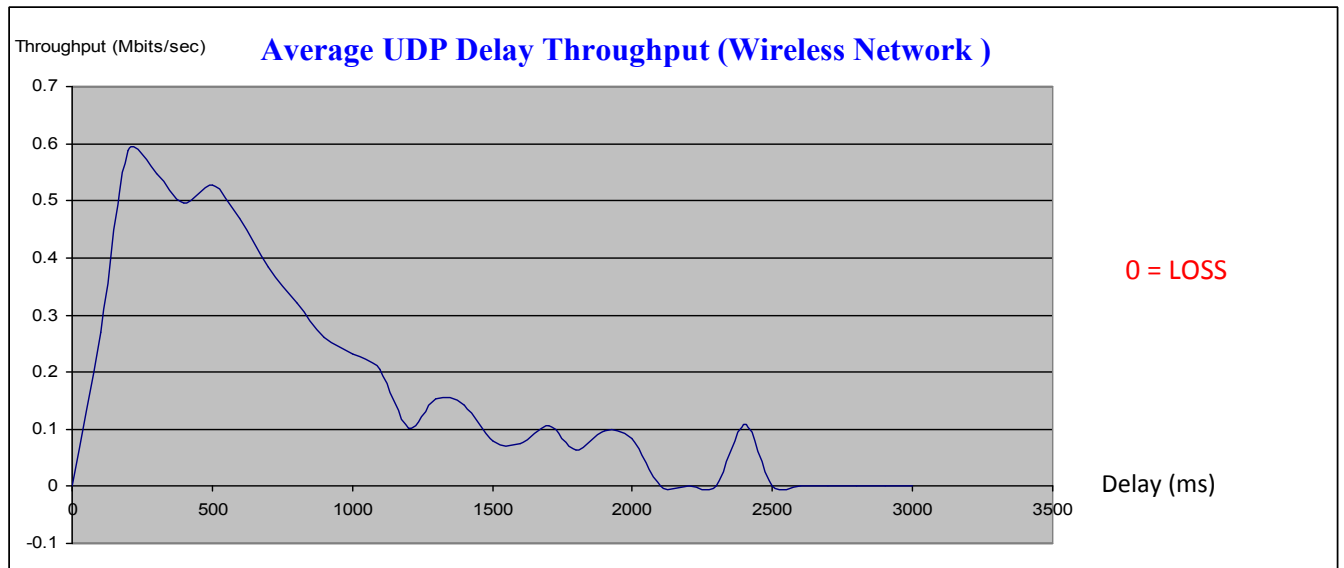
### Wireless Network

**TCP:**





## UDP:



**Summary:** After we apply delay to the experiment. For TCP graph we apply delay time reached to 3000ms. We still get the data but For UDP after we apply delay, it seems to have some packets lost so for both TCP and UDP we can conclude that as many as delay applies, the less throughput we get or even loss for UDP.

## 1.3 Jitter

Then we also apply jitter to the experiment by using this command:

```
Sudo tc qdisc add dev XXXX root netem delay YYms ZZms
```

Where XXXX = Network devices for ex. eth0

YY = Delay time for ex. 100ms

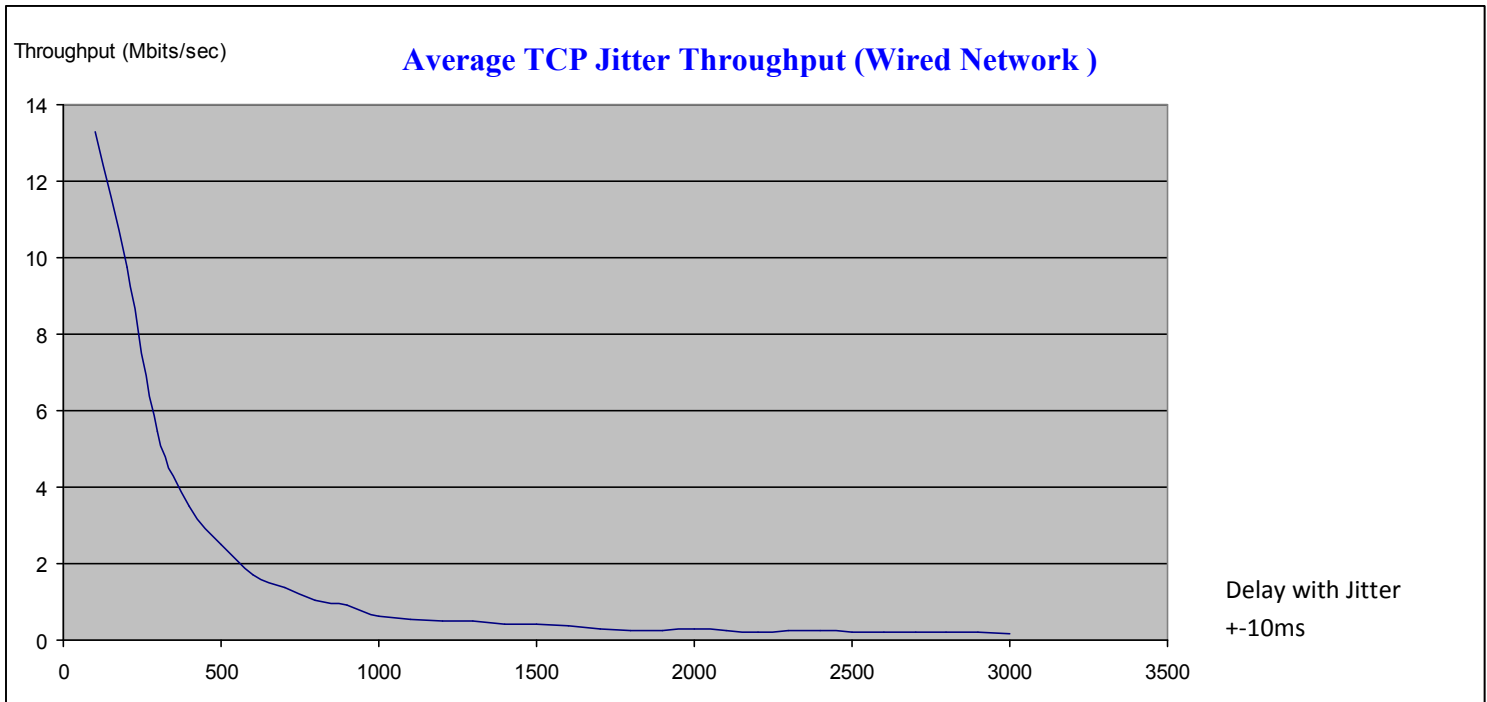
ZZ = Jitter time for ex. 10ms

## Parameters

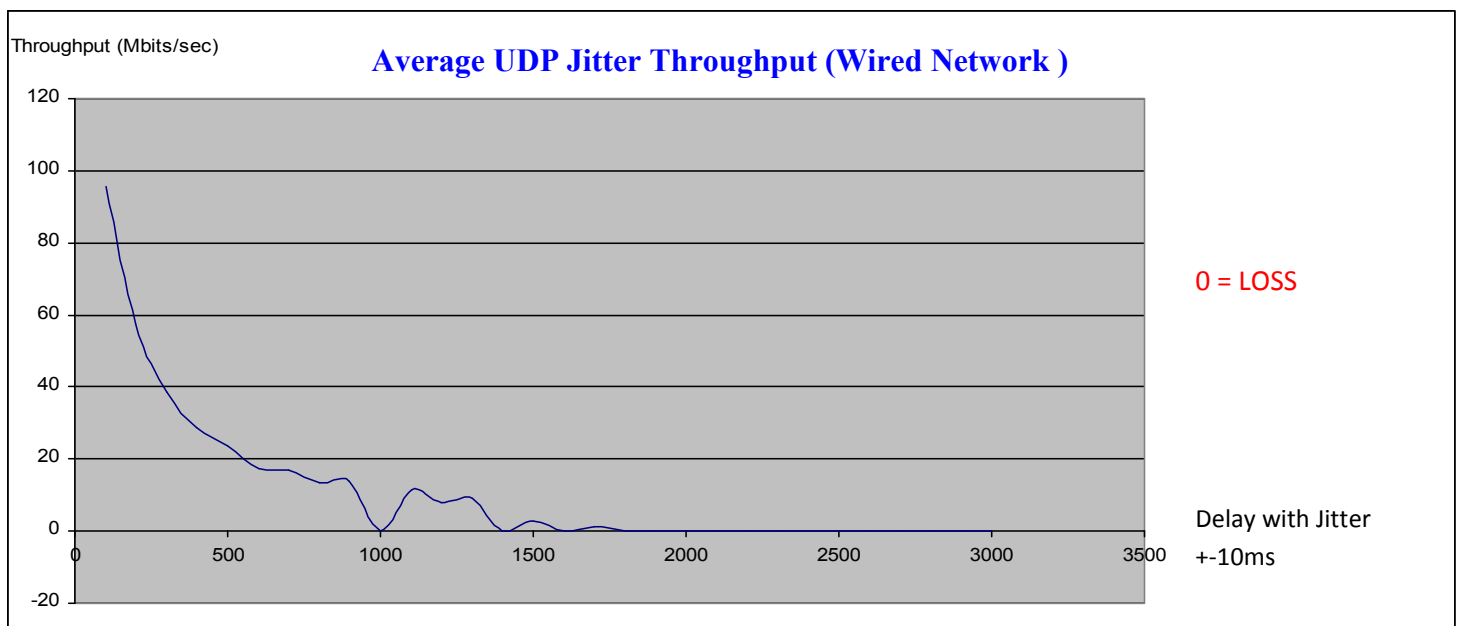
Parameters	Default values
Sending rate	54.2Mb/s
Delay	Increase 100ms every times
Constant Jitter	+/-10ms

## Wired Network

### TCP:

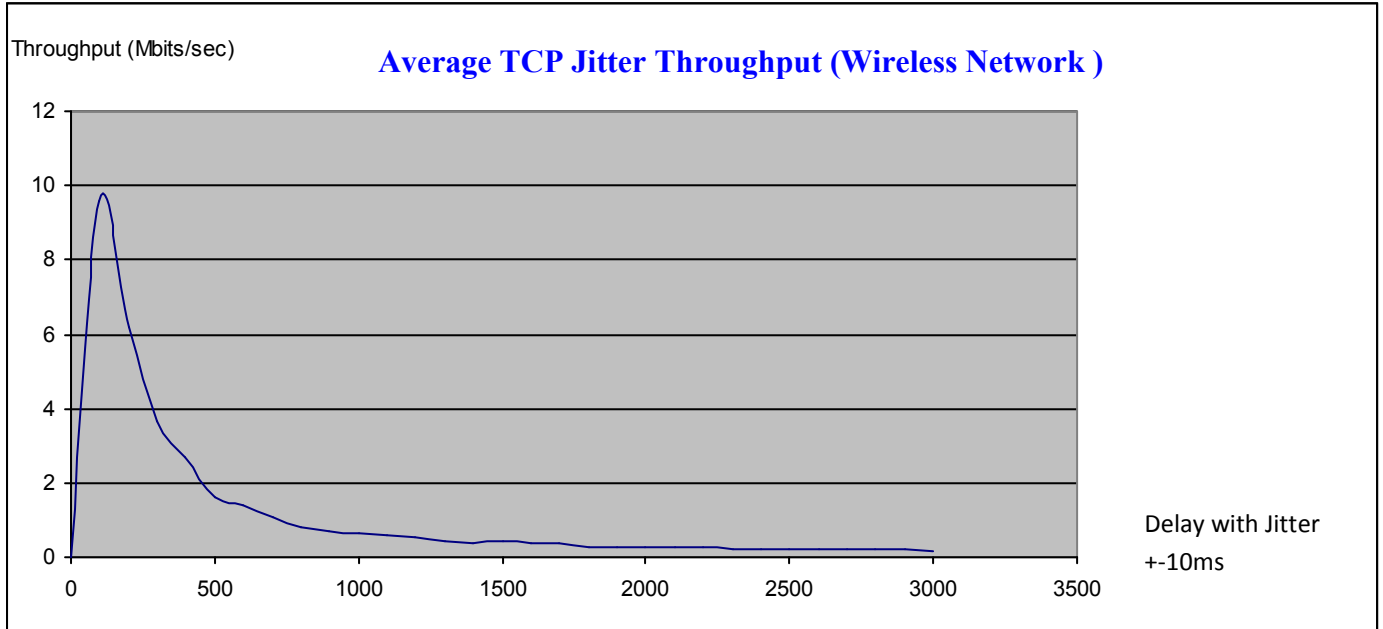


### UDP:

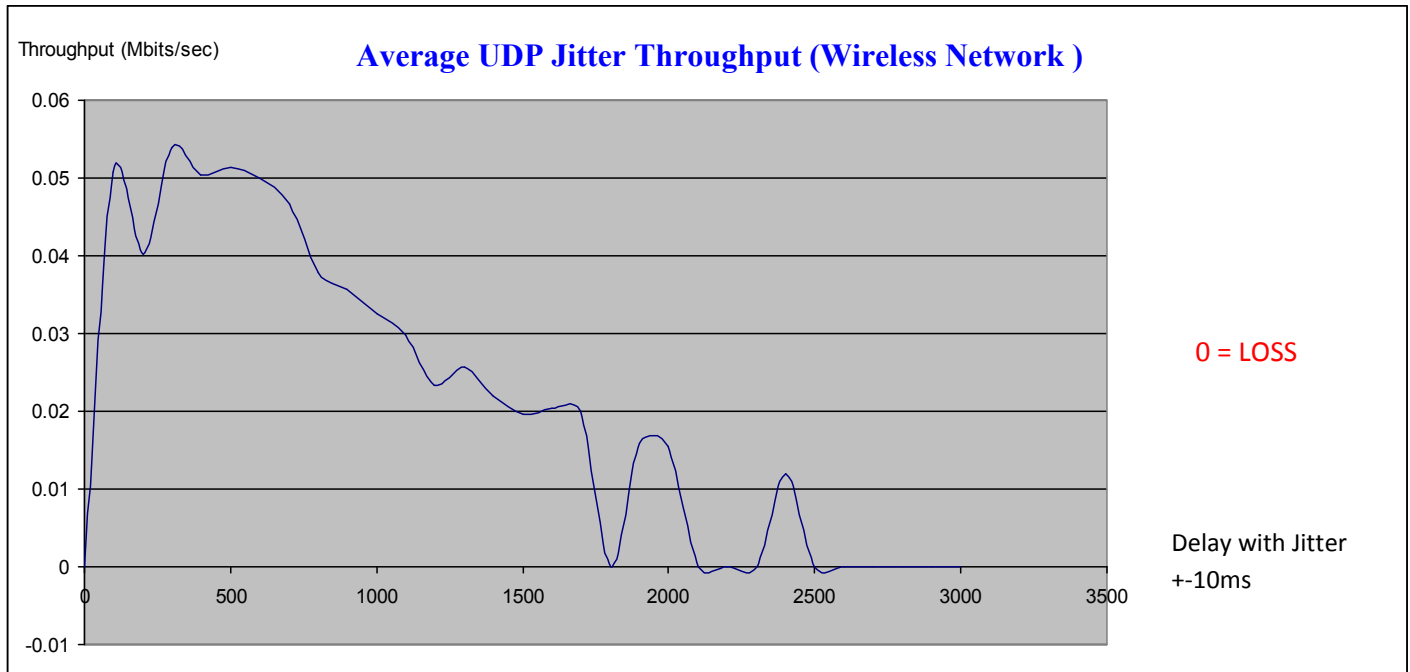


## Wireless Network

**TCP:**



**UDP:**



**Summary:** As the graphs show for TCP both wired and wireless network eventhought we apply jitter to the experiment but we still get the packets but for UDP after we apply there is some packets lost.

## **Conclusion**

Firstly, 1.1 TCP and UDP throughput we can see the real throughput from both TCP and UDP on both wired and wireless network. After we apply 1.2 Delay, it impacts the throughput on both TCP and UDP then we also test 1.3 the jitter and it also impact on the throughput so we can conclude that both delay and jitter are factors that can impact on throughput whether the protocol are TCP or UDP.

## Task 5 : Impact on Bandwidth Delay Product on TCP Throughput

### 1. Aim :

This task aims to investigate the factor that will be impact on TCP Throughput. BDP contain with three parameters involve which are Advertised Window, Data rate and RTT (Round-Trip-Time).

From the equation :

$$\text{awnd} \geq \text{rate} * \text{RTT}$$

BDP refer to  $(\text{rate} * \text{RTT})$

### 2. Topology : Wire, Wireless

### 3. Parameters :

- **Delay** – define on both sender and receiver to 8ms.
- **Receiver Window Size** – Keep increasing from 83Kb to 175Kb.

### 4. Methodology :

Before conduct the experiment, we have to setup some command on both receiver and sender as follow.

First, disable offloading on both sender and receiver to prevent TCP/IP function to shift the LAN card.

```
sudo ethtool -K eth0 gso off
sudo ethtool -K eth0 gro off
sudo ethtool -K eth0 tso off
```

Second, Set the TCP congestion control algorithm to **Reno**. On sender:

```
sudo sysctl net.ipv4.tcp_congestion_control=reno
```

Third, turn off auto-tuning of the TCP receive buffer size. On the receiver:

```
sudo sysctl net.ipv4.tcp_moderate_rcvbuf=0
```

Fourth, adjust the Receiver Window Size. On receiver

```
sudo sysctl net.ipv4.tcp_rmem="4096 100000 6000000"
```

Fifth, turn off the auto tuning capability. On receiver

```
sudo sysctl net.ipv4.tcp_moderate_rcvbuf=0
```

Now, the default Receiver Window Size changed to 100000 (97500 actually). We can test it by launch a server on the receiver.

Next, we add a delay to 5ms on both sender and receiver.

```
sudo tc qdisc add dev eth0 root netem delay 8ms
```

Then test Round-Trip-Time by use ping command to see time.

```
ping -c 5 192.168.1.122
PING 192.168.1.122 (192.168.1.122) 56(84) bytes of data.
64 bytes from 192.168.1.122: icmp_req=1 ttl=64 time=16.6 ms
64 bytes from 192.168.1.122: icmp_req=2 ttl=64 time=16.6 ms
64 bytes from 192.168.1.122: icmp_req=3 ttl=64 time=16.6 ms
64 bytes from 192.168.1.122: icmp_req=4 ttl=64 time=16.6 ms
64 bytes from 192.168.1.122: icmp_req=5 ttl=64 time=16.6 ms
```

We see the average RTT is 16.6ms. The real 0.6ms RTT plus the extra 8ms delay added by the sender and the extra 8ms delay added by the receiver.

With a 100Mb/s bandwidth and 16ms RTT the BDP is 200KB. We expect to see TCP throughput to be maximum when the Advertised Window is larger than 200KB, while the throughput should be approximately  $awnd/RTT$  when less than 200KB.

When all are set, now we are ready to conduct an experiment.

Begin with launch a server on receiver with option `-w` to specific Receiver Window Size.

```
iperf -s -w 83000
```

Sender connected to server by specific IP of server.

```
iperf -c 192.168.1.122
```

Then, record the throughput and repeat the experiment. When you satisfy stop the server and launch again with different Receiver Window Size.

## 5. Experiment Results :

Bandwidth	RTT	BDP	-w	Buffer Size	Max Adv.Win	Expected Thr	Measure Thr	Accuracy
[Mb/s]	[ms]	[KBytes]	[KBytes]	[KBytes]	[KBytes]	[Mb/s]	[KBytes]	[%]
100	16	200	83	168	126	63	58.1	92
100	16	200	100	200	150	75	68.1	90
100	16	200	125	250	187.5	93.75	84.5	90
100	16	200	150	256	192	96	88.2	91
100	16	200	175	256	192	96	88.4	91

### Wire

Bandwidth	RTT	BDP	-w	Buffer Size	Max Adv.Win	Expected Thr	Measure Thr	Accuracy
[Mb/s]	[ms]	[KBytes]	[KBytes]	[KBytes]	[KBytes]	[Mb/s]	[KBytes]	[%]
54	16	108	83	168	126	54	14.2	26
54	16	108	100	200	150	54	14.4	26
54	16	108	125	250	187.5	54	14.4	26
54	16	108	150	256	192	54	Same as above	Same as above
54	16	108	175	256	192	54	Same as above	Same as above

### Wireless

## 6. Explanation:

TCP perform poorly on wireless that why the throughput is very low. We have tried to set the Receiver Window Size as much as possible for the buffer size, but it cannot exceed 256Kbytes.

## 7. Conclusions:

As a result, the tables above show that when Advertised Window is less than BDP, the throughput is less than we expected, and when Advertised Window Is larger than BDP, the throughput reached maximum.

We knew that when we set `-w` option on server, the kernel will allocate double, and the maximum buffer is 75%. But, we found that the maximum buffer size cannot exceed 256Kbytes.



## Task8

These experiments mainly focus on TCP and UDP session. The experiment will be carry out by opening 2 terminal windows. One will be running server for TCP with port 3003, while the other terminal will be running a server for UDP with port 4004. With the two servers port open on the server computer. We can then proceed to transmit packets from client computer.

Commands:

Open TCP server with port 3003:

```
iperf -s -p 3003
```

Open UDP server with port 4004:

```
iperf -s -u -p 4004
```

Sending TCP packets to port 3003 for *n* seconds:

```
iperf -c [SERVER IP] -t n -p 3003
```

Sending UDP packets with **100** Mbits/sec bandwidth to port 4004 for *n* seconds in *p* parallel fashion:

```
iperf -c [SERVER IP] -t n -p 4004 -b 100M -P p
```

## Part 1: Start and stop at the same time (60 seconds)

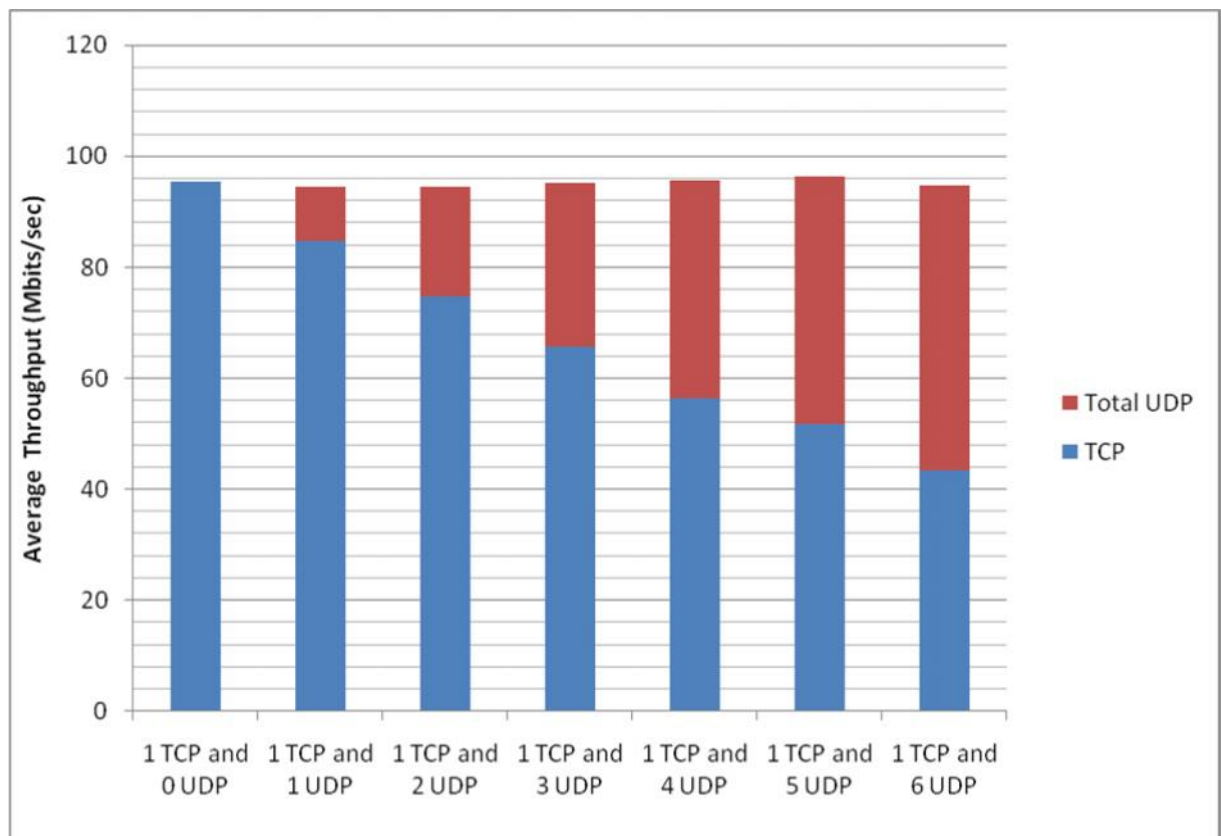
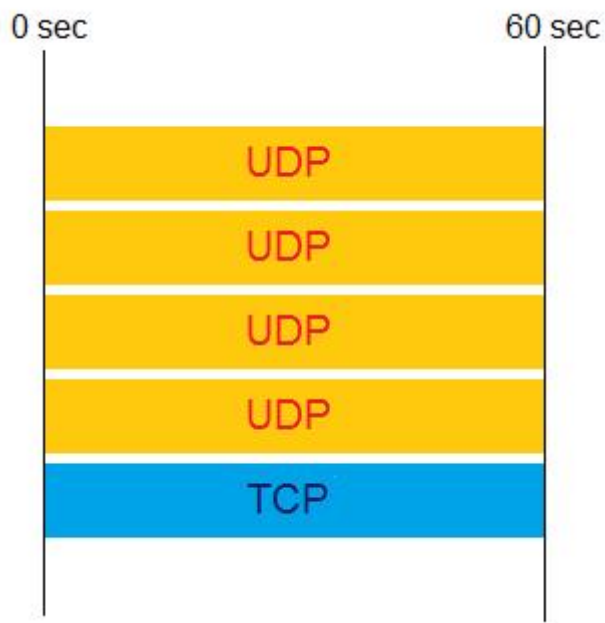
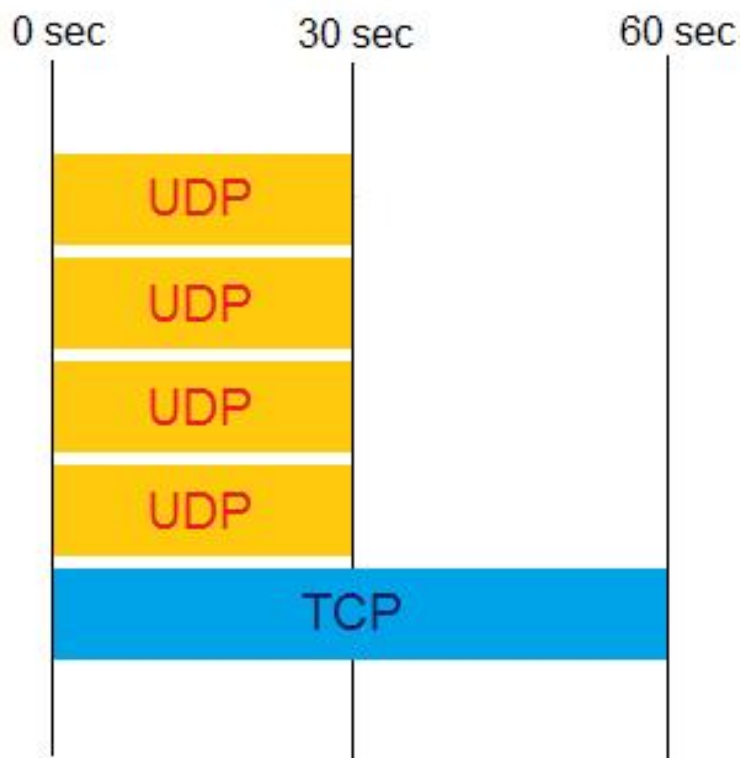


Figure 1.

As you can see in this bar chart, as the number of UDP transmission increase the throughput of TCP transmission decrease gradually. The UDP transmission takes over the bandwidth use by TCP to transmit its

packets. As more session of UDP appear, lesser share of TCP's bandwidth available. Another thing to point out from this chart is that the total throughput always added up to 95 Mbits/sec. From this result, we can interpret that TCP will always have higher throughput compare to individual UDP transmission. This is because TCP have fairness as one of its property to manage its transmission. However UDP have nothing assisting its transmission. Therefore, TCP is given higher priority to access to the resource. As more number of UDP transmits concurrently, their individual throughputs decrease slightly.

### Part 2: Start together, all UDP finish at different time



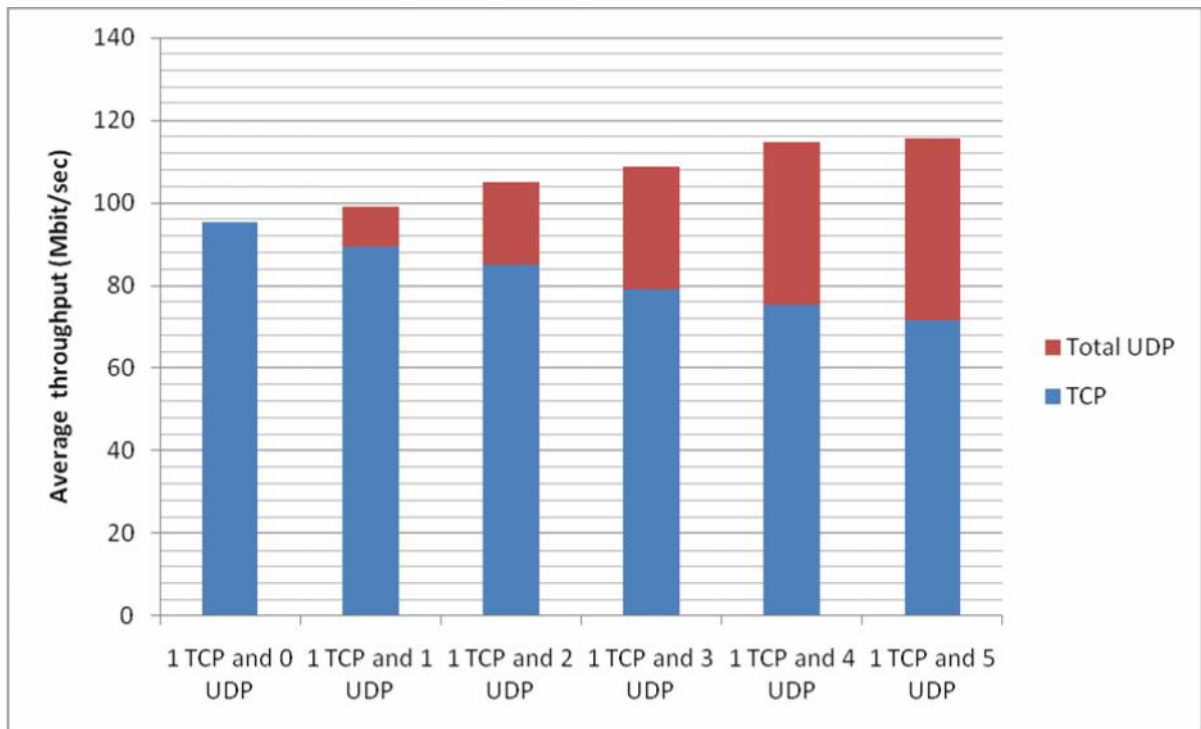


Figure 2.

In part 2, we can see that the UDP transmission does not have as much effect as in part 1. This is because; UDP transmission only lasted for half of the time of the TCP transmission. Therefore, it will take up less bandwidth than in part 1. As soon as the UDP transmission finish, it give back all its bandwidth used during their transmission back to TCP. When this happens, TCP can then transmit with it full bandwidth. In this bar chart, the total throughputs exceed 100 Mbits/sec, which should be the limit speed of the transmission. This is possible because after UDP finish its transmission, it release the resource back to TCP. This allows TCP to transmit back at its full bandwidth, by averaging the first half and second half, TCP average throughput rise.