Computer Networks

Practice Session 2

Team members:

- Ahmad Ghalawinji
- Bashar Awada
- Nadine Zaatary

1. Analysing network performances

1.1. CSMA/CD

1.2. First Evidence of Collisions

PC1: **ifconfig igb0 147.171.108.102/24 up**

PC2: **ifconfig igb0 147.171.108.155/24 up**

PC3: **ifconfig igb0 147.171.108.160/24 up**

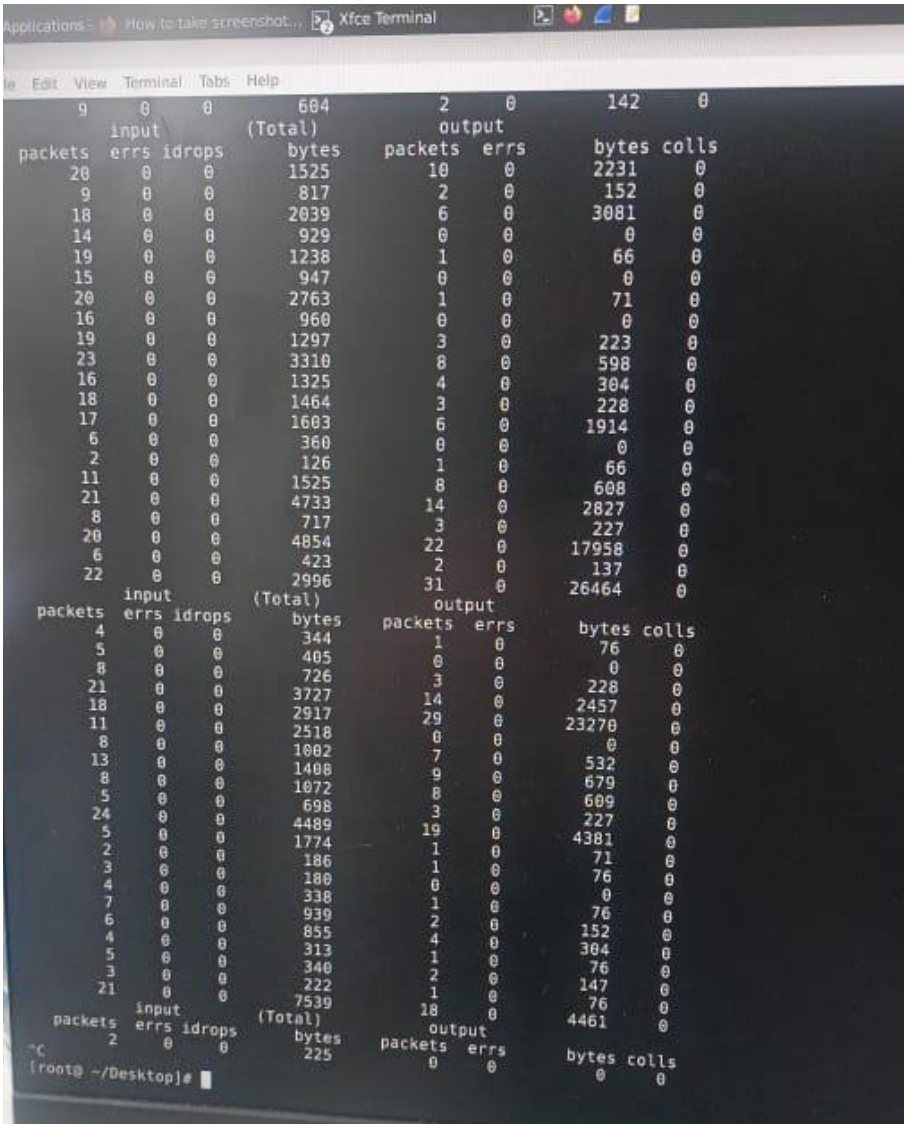
PC4: **ifconfig igb0 147.171.108.170/24 up**

Ran the command: Minicom

In minicom terminal, we ran the following command: int 1 speed-duplex 100-half

Q1. The information can only go in one direction, and the device cannot transmit and receive simultaneously. For now, no collision occurred because we are only working with one computer. Also, we switched to half duplex in order to consider this switch as a hub.

PC1: **- - netstat -w 2 igb0 - -**



- PC1 and PC2: **udptarget -k**
- PC2: **udpmt -p 13000 147.171.108.102**

Q2. After running netstat on PC1 we have 0 collisions as shown below.

[root@ ~/Desktop]# netstat -w 2 igb0

input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
16482	0	0	24916949	35	0	28177	0
16225	0	0	24563196	0	0	0	0

16182	0	0	24419058	3	0	282	0
16369	0	0	24775528	0	0	0	0
16478	0	0	24941933	1	0	71	0
15979	0	0	24169412	3	0	198	0
16372	0	0	24774267	3	0	228	0
16303	0	0	24669951	5	0	375	0
16358	0	0	24751620	1	0	76	0
16268	0	0	24600017	18	0	2864	0
15990	0	0	24197441	3	0	208	0
17031	0	0	25710073	1	0	130	0
16088	0	0	24343025	4	0	304	0
16328	0	0	24684912	6	0	456	0
15940	0	0	24097459	7	0	523	0
16081	0	0	24319613	6	0	436	0
16240	0	0	24520503	0	0	0	0
16473	0	0	24845691	3	0	218	0
16225	0	0	24469770	10	0	3526	0
16342	0	0	24644402	32	0	25043	0
16324	0	0	24629096	5	0	360	0
input		(Total)		output			

- PC1: **udpmt -p 13000 147.171.108.155**

[root@ ~/Desktop]# netstat -w 2 igb0

input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
10053	59	0	14962429	6888	59	10428432	1647
8772	59	0	13255462	6993	59	10575897	1654
9605	56	0	14530402	6641	56	10053026	1633
8966	56	0	13540826	7321	56	11063641	1634
8869	51	0	13349948	7426	51	11240142	1635
9521	61	0	14404690	6736	61	10198304	1714
9401	61	0	14220155	6937	61	10501170	1620
8396	52	0	12613043	8125	52	12286627	1702
9369	58	0	14159213	6614	58	10004945	1724

9298	55	0	14027013	7032	55	10603619	1690
8878	60	0	13331877	7463	60	11280622	1639
9646	56	0	14522594	6702	56	10094436	1619
9564	56	0	14447400	6828	56	10323332	1677
8849	57	0	13388701	7306	57	11058388	1737
8710	55	0	13107696	7578	55	11470326	1673
9194	56	0	13908137	7713	56	11675968	1654
10268	63	0	15531286	6010	63	9096244	1703
8792	58	0	13217609	7525	58	11385752	1660
10080	61	0	15246685	5778	61	8745006	1655
8088	53	0	12233654	8547	53	12935890	1679
9798	64	0	14819848	5793	64	8769088	1791

Q3. We are sending and receiving packets at the same time and over the same media so collision will occur. CSMA/CD detects or listens to see if the shared channel for transmission is busy or not, and defers communications until the channel is free. Collisions are detected using collision detection technology, which detects broadcasts from other stations. When a collision is detected, the station stops transmitting, sends a jam signal, and then waits a random time interval before retransmitting.

1.3. How to measure effective throughput

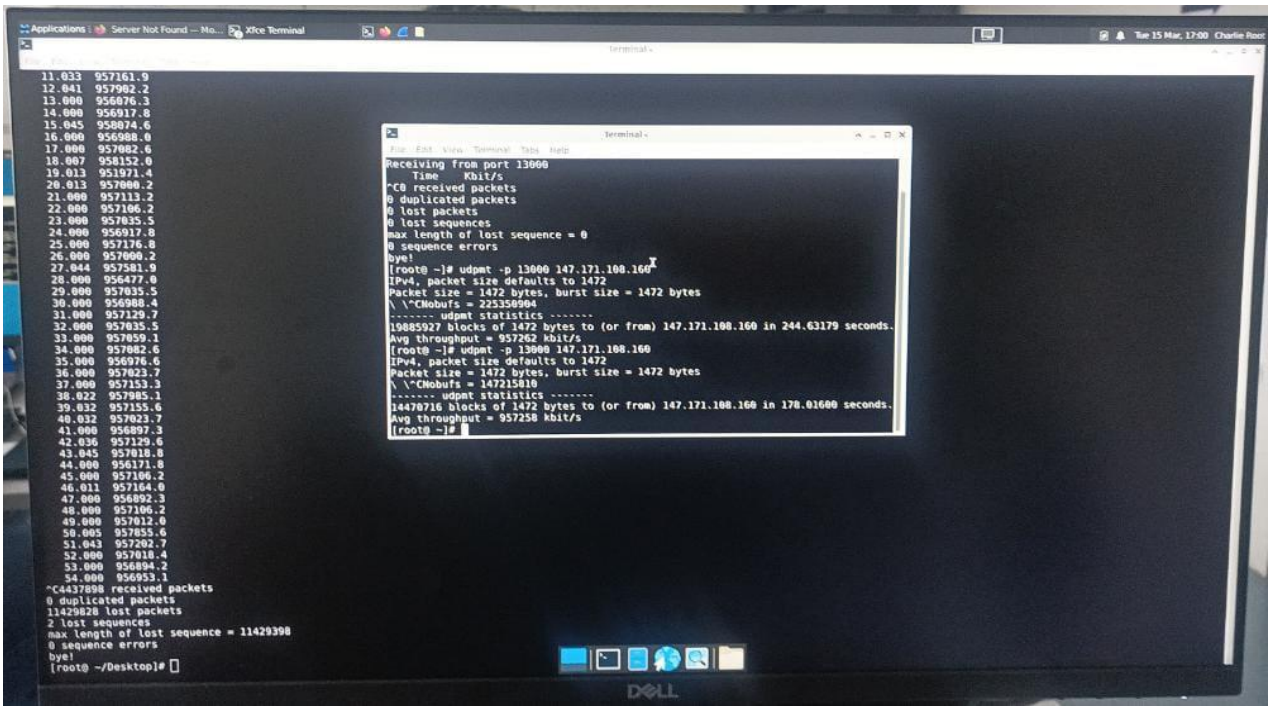
1.4. Effective throughput for a single stream

Q4.

Size = 1472 bytes

Delay = 178.016000 seconds

Average throughput = 957258 kbit/s



Q5.

60 bytes

[root@ ~]# udptarget -k

IPv6 (and IPv4) protocol

Receiving

from port 13000

Time Kbit/s

0.801 38122.1

1.801 47616.0

2.801 47619.8

3.801 47617.0

4.801 47619.4

5.801 47617.4

6.801 47620.8

7.801 47616.0

8.801 47617.9

9.801 47618.9

10.801 47619.4

11.801 47617.4

1470 bytes

[root@ ~]# udptarget

IPv6 (and IPv4) protocol

Receiving from port 13000

Time Kbit/s

1.00

0 95714.6

2.000 95704.5

3.000 95797.0

4.000 95702.9

5.000

95692.7

6.000 95797.0

7.000 95714.6

8.000 95692.7

9.000 95701.3

78315

received packets

0 duplicated packets

7519 lost packets

1 lost sequences

max length of lost

sequence = 7519

0 sequence errors

1480 bytes

[root@ ~]# udptarget -k

IPv6 (and IPv4)

protocol

Receiving from port 13000

```
    Time    Kbit/s
Sequence error : expected = 18848,
received = 6100
    1.000   94176.0
    2.000   94174.2
    3.000   94144.0
    4.000
94160.0
    5.000   94256.0
Sequence error : expected = 48170, received = 134
    6.000
94176.0
    7.000   94144.0
    8.000   94080.0
    9.000   94144.0
   10.000   94176.0

11.000   94256.0
67470 received packets
0 duplicated packets
78192 lost packets
3 lost
sequences
max length of lost sequence = 48036
2 sequence errors
```

2880 bytes

```
[root@ ~]# udptarget -k -q
IPv6 (and IPv4) protocol
Receiving from port 13000
        Time    Kbit/s
        0.988   22717.4
        1.988   24261.1
        2.988   23708.2
```

3.988 23293.4

4.988 21081.6

5.988 22256.6

6.988 18478.1

7.988 24768.0

8.988 23823.4

9.988 21542.4

10683 received packets

196 duplicated packets

5492717 lost packets

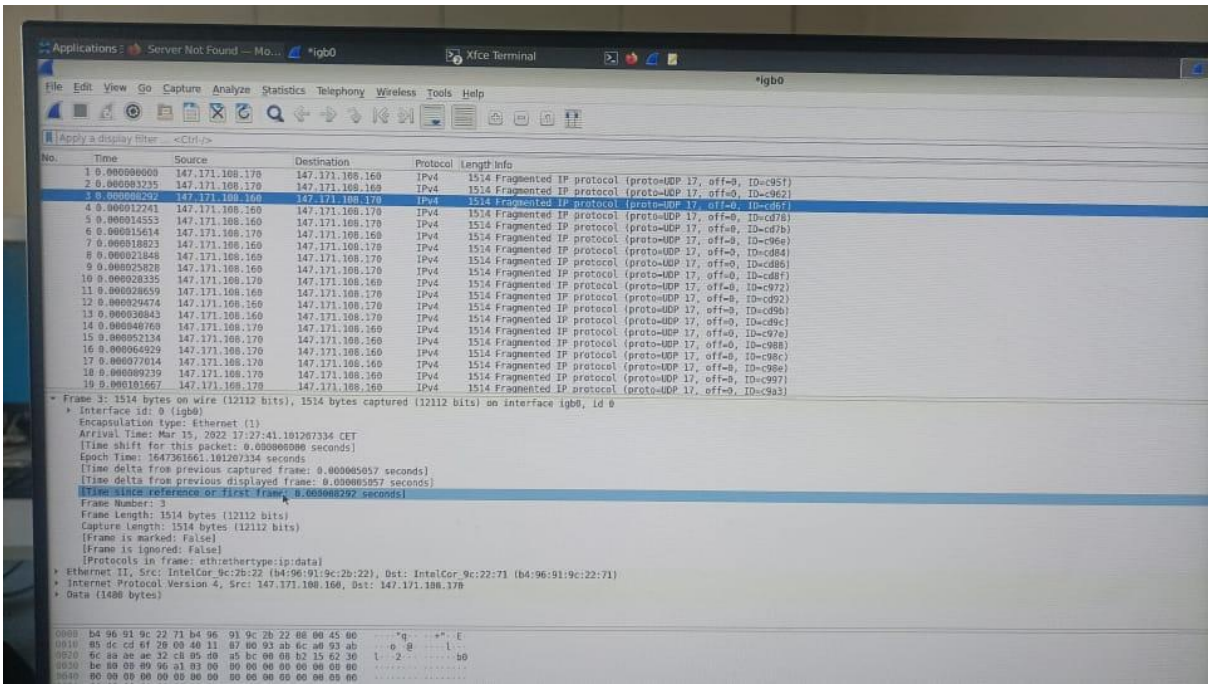
4905 lost sequences

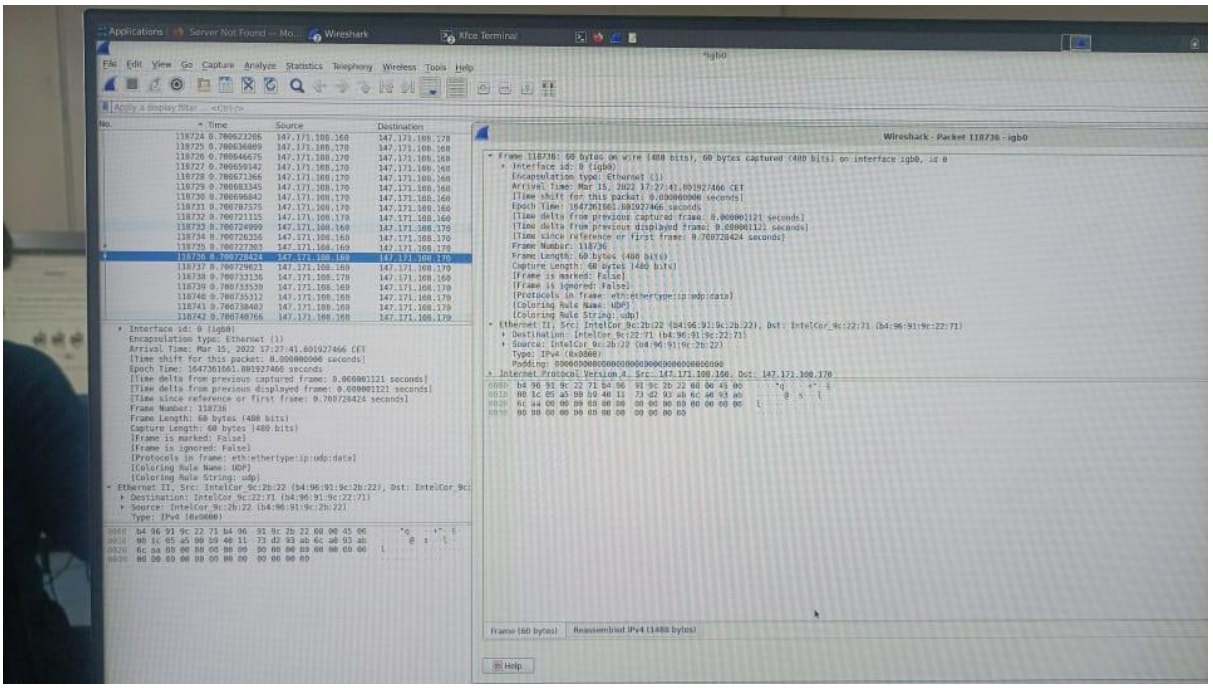
max length of lost sequence = 23816

0 sequence errors

We notice that when the packet size is too small, 60 bytes, the throughput is small, and when we use throughput 1470, the packet is perfectly fragmented into two maximum length packets, allowing the transmission to fully use the network's capacity.

Q6.





Increasing the packet size to 1480 bytes adds a reminder that another transmission with a small chunk is required, lowering the throughput.

For 1460 bytes, we notice that the size of one packet is being divided into two almost equal packets to be sent so we achieve good throughput.

For 1480 bytes, the packet is fragmented into one big packet and another small one.

1.5. Effective throughput with multiple streams and a switch

PC1: **ifconfig igb0 192.168.0.2/24 up**

PC2: **ifconfig igb0 192.168.0.1/24 up**

PC3: **ifconfig igb0 192.168.0.3/24 up**

PC4: **ifconfig igb0 192.168.0.4/24 up**

Q7. The average throughput on PC2 is 960907 kbits/s in 7.73305 seconds. And the average throughput on PC4 is 960920 kbits/s in 7.72524 seconds. Because there is no relationship between the connections, we see that the throughput is reasonably close.

Q8. All PCs sent packets to PC1 at rates of 18721283 kbit/s, 964446 kbit/s, and 962914 kbit/s for PC2. PC3, and PC4 relatively. Because of collisions, we observe a significant variation in throughput.

1.6. Effective throughput with multiple streams and a switch

Q9. For the experiment depicted in Figure 3: The average throughput on PC2 is 81240 kbits/s in 7.36591 seconds. And the average throughput on PC4 is 65222 kbits/s in 5.51824 seconds. Figure 4 depicts an experiment: PC2, PC3, and PC4 all transmitted packets to PC1 at 34972 kbit/s, 39618 kbit/s, and 35766 kbit/s, respectively. We see that the throughput is significantly lower than with a switch.

Q10. We saw no collisions and had higher throughputs in the three divergent scenarios. This is because when one machine sends to several machines, it only sends once at a time. So PC1 sends to PC2, and then to PC3, and finally to PC4. As a result, this is a better mechanism than three converging streams.

2. Virtual Networks - VLANs

2.1. Switch administration how-to

2.2. Basic switch configuration

show running-config

```
hostname "HP ProCurve Switch 6108"
cdp run
snmp-server community "public" Unrestricted
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-8
   ip address 192.168.0.254 255.255.255.0
```

2.3. VLAN configuration

2.3.1. Layer 1 VLANs / ports VLANs

Q11. PC2 and PC3 cannot ping each other because they are connected to different vlans.

Q12. check it by pinging to a non existing host

2.4. Tagged VLANs

Q13.

Q14.