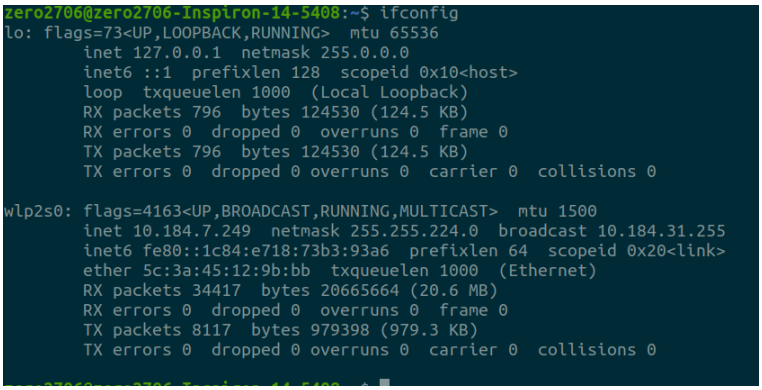
COL334 Assignment 1

Name :- Ujjwal Mehta

Entry No. :- 2020CS10401

In this assignment we will be testing some basic networking tools, capturing and reviewing network packets using **wireshark**. Before moving to the main tasks we will answer initial writeup questions using commands like **ifconfig**, **ping**, **traceroute** and **nslookup**.**It should be noted that I'm using ubuntu linux on which I'm using these commands**.

1. For finding IP address of the machine, I used the **ifconfig** command and here as we can see in the screenshot the inet value associated with the wlp2s0 interface(IITD Wifi here) is giving the ip address value as **10.184.7.249**. Also this ip address value for my machine was changing upon using different internet hosts(like it was different when I was using mobile hotspot)



2. For finding the ip addresses for www.google.com and www.facebook.com we will use the nslookup command. As shown in the below image we can see that the ip address associated for www.google.com is **142.250.206.100** and for www.facebook.com is **157.240.16.35**. The resuls are shown in the below image.

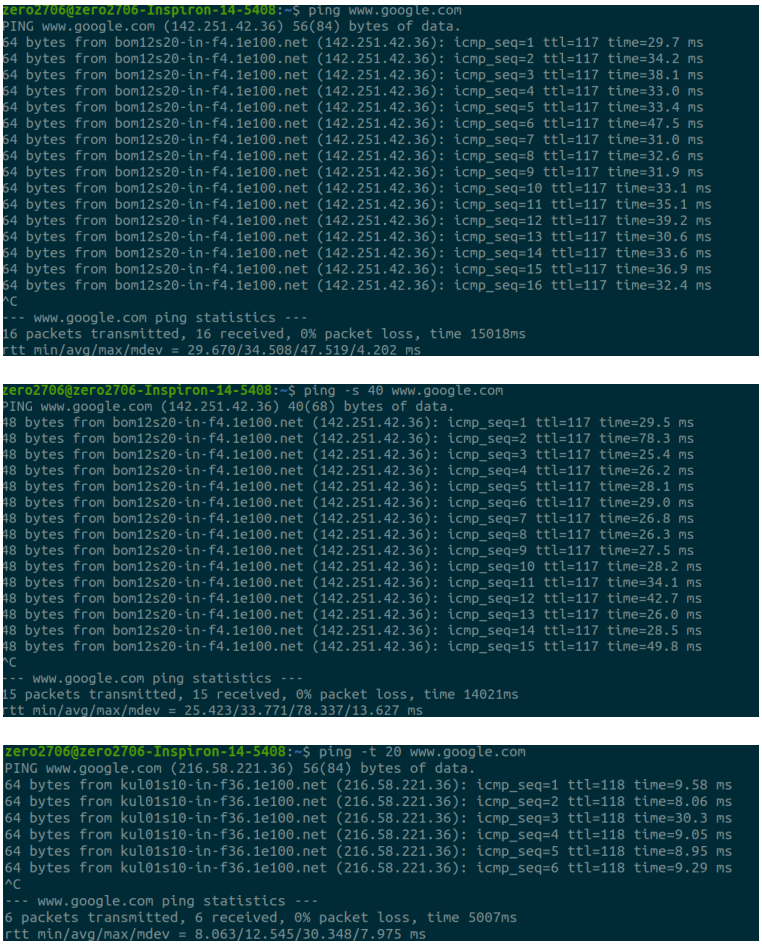
```
zero2706@zero2706-Inspiron-14-5408:~$ nslookup www.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.206.100
Name:   www.google.com
Address: 2404:6800:4002:806::2004

zero2706@zero2706-Inspiron-14-5408:~$ nslookup www.facebook.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 157.240.16.35
Name:   star-mini.c10r.facebook.com
Address: 2a03:2880:f12f:183:face:b00c:0:25de
```

Here we can also get the ip addresses by trying over different dns servers which is as shown in the below image. (we can see that we get different result from the previous ones because when the dns server to which we request for the ip address changes then it results in ip address which that dns server finds)

```
zero2706@zero2706-Inspiron-14-5408:~$ nslookup www.google.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.206.100
Name:   www.google.com
Address: 2404:6800:4002:82b::2004

zero2706@zero2706-Inspiron-14-5408:~$ nslookup www.google.com 1.1.1.1
Server:         1.1.1.1
Address:        1.1.1.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.183.164
Name:   www.google.com
Address: 2404:6800:4009:824::2004

zero2706@zero2706-Inspiron-14-5408:~$ nslookup www.facebook.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 157.240.16.35
Name:   star-mini.c10r.facebook.com
Address: 2a03:2880:f12f:83:face:b00c:0:25de

zero2706@zero2706-Inspiron-14-5408:~$ nslookup www.facebook.com 1.1.1.1
Server:         1.1.1.1
Address:        1.1.1.1#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 157.240.16.35
Name:   star-mini.c10r.facebook.com
Address: 2a03:2880:f12f:83:face:b00c:0:25de
```

3. We can ping to www.google.com using the ping command and we can change

the packets sizes to be sent as well as their time to live using various command paramters (like for changing size we can use -s flag and for changing ttl we can use -t). The result are as shown in image below.



Here we can see that the actual size of the packet that is sent is greater then the size given as parameters because the size of the headers in the packets gets increased.

4. We can get the path followed by the network packets while exchanging queries from a particular server by using **traceroute** command. We can force the traceroute to use IPv4 if we add the -4 flag in the traceroute command. Also the path we get depends on the internet network that we are using. (The first image below is using IITD wifi and the second is using my phone's hotspot).

```
zero2706@zero2706-Inspiron-14-5408:~$ traceroute www.iitd.ac.in
traceroute to www.iitd.ac.in (10.10.211.212), 30 hops max, 60 byte packets
 1  10.184.0.14 (10.184.0.14)  5.271 ms  5.265 ms  5.251 ms
 2  10.254.236.18 (10.254.236.18)  5.198 ms 10.254.236.10 (10.254.236.10)  5.217 ms  6.072 ms
 3  www.iitd.ac.in (10.10.211.212)  5.160 ms  5.148 ms  5.135 ms
zero2706@zero2706-Inspiron-14-5408:~$ traceroute www.google.com
traceroute to www.google.com (216.58.221.36), 30 hops max, 60 byte packets
 1  10.184.0.14 (10.184.0.14)  11.117 ms  11.082 ms  11.068 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

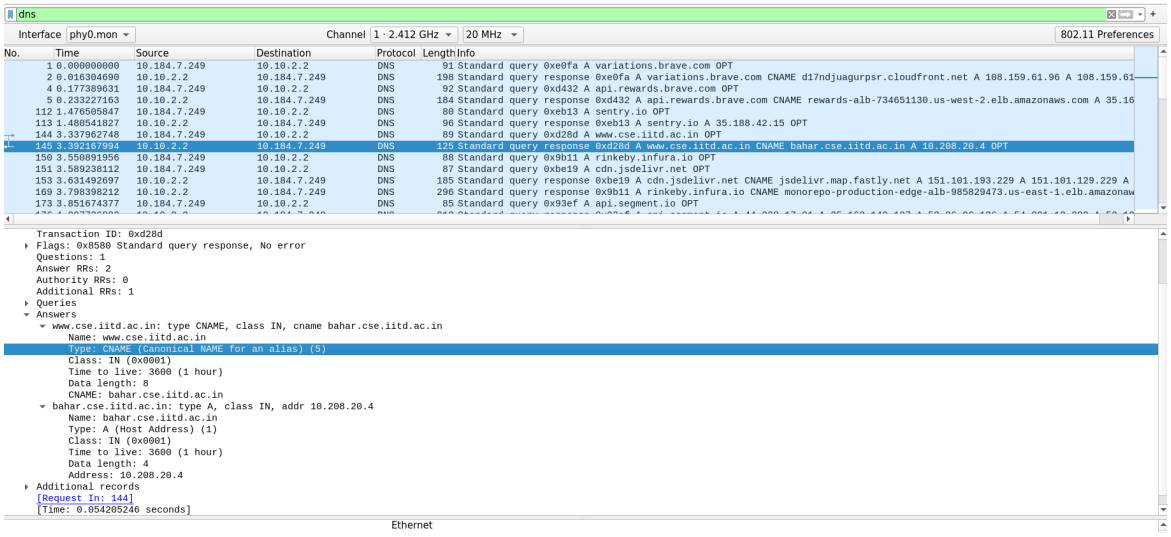
```
zero2706@zero2706-Inspiron-14-5408:~$ traceroute www.iitd.ac.in
traceroute to www.iitd.ac.in (10.10.211.212), 30 hops max, 60 byte packets
 1  _gateway (192.168.143.141)  8.515 ms  8.475 ms  8.460 ms
 2  10.184.0.14 (10.184.0.14)  9.646 ms  9.631 ms  9.617 ms
 3  10.254.236.18 (10.254.236.18)  8.520 ms  9.587 ms 10.254.236.10 (10.254.236.10)  10.389 ms
 4  www.iitd.ac.in (10.10.211.212)  10.311 ms  10.297 ms  10.283 ms
zero2706@zero2706-Inspiron-14-5408:~$ traceroute www.google.com
traceroute to www.google.com (142.250.206.100), 30 hops max, 60 byte packets
 1  _gateway (192.168.143.141)  2.208 ms  2.159 ms  2.380 ms
 2  10.184.0.14 (10.184.0.14)  10.204 ms  10.191 ms  10.179 ms
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Now will do the packet analysis on wireshark and tasks given in the assignment.

1. DNS Task

In this task we will analyse the packet transfer from wireshark over DNS protocol when we try to access `www.cse.iitd.ac.in` after flushing the local DNS cache .

1. We can see from the below wireshark screenshot in the packet details that the protocol for the transfer of DNS packets is **UDP** (user datagram protocol).



2. Upon capturing the network packets we can see that the query DNS packet is 1 (for getting the ip address of www.cse.iitd.ac.in) and for that we have 1 response DNS packet sent to and from our local DNS server when connected to IIT Delhi wifi (Though there are other DNS packet transfer too relating to browser which we can see in the wireshark screenshot)

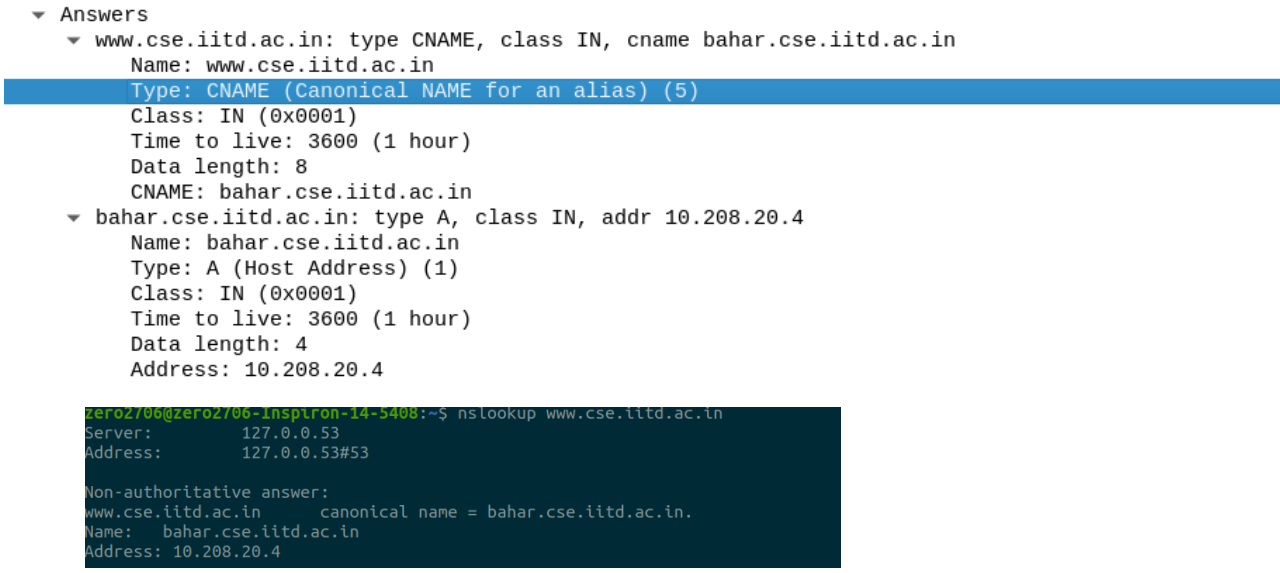
3. Here as we can see in the records of the DNS response that first the record contains the information about the alias hostname (that is bahar.cse.iitd.ac.in) and then their

is information about the ip address of the server of www.cse.iitd.ac.in which is **10.208.20.4** hence the number of DNS server involved is 1 which is the local DNS server.

4. The DNS server which replies with the actual ip address is our local DNS server since it is the only DNS server.

5. As we can see that all of our DNS servers are responding to fetch ip address of www.cse.iitd.ac.in

6. The resource records involved in getting the ip address are given in the below image. Here **the TTL is 3600, the type is CNAME(for resolving the alias hostname) and A (for host address), name is www.cse.iitd.ac.in and the value of ip address obtained is 10.208.20.4 which is the same answer that we get from nslookup command(image below of terminal)**.



2. Iperf Task

In this task we will capture the communication between iperf3 client and remote server using iperf3 command.

1. Here in the communication between iperf3 client and remote server, the number of UDP packets exchanged is equal to **2529** which can be displayed using the capture file properties in wireshark.
2. The bulk data is sent from the remote server to the client and the average size of packet is **566 bytes**.
3. Here we can calculate the throughtput using the total bytes transferred in wireshark for the communication and divide it by the total time taken for exchange which is the time stamp difference of last and first packet. Now after looking at the wireshark file we can see that

$=> Throughput = (bytestransferred) \div (timelast - timefirst)$

$=> Throughput = (566 \times 2527 + 46 \times 2) \div (10.248)$

=> Throughput = 139.575 kbytes per second

As we can see in the below image that this throughput value matches with the one given by statistics of wireshark though it is slightly more then the one displayed in terminal (128 Kbytes per second) because header data also gets included in the transfer.

Time

First packet:	2022-08-28 13:23:29
Last packet:	2022-08-28 13:23:40
Elapsed:	00:00:10

Capture

Hardware:	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz (with SSE4.2)
OS:	Linux 5.13.0-27-generic
Application:	Dumpcap (Wireshark) 3.6.5 (Git v3.6.5 packaged as 3.6.5-1~ubuntu20.04.0+wiresharkdevstable)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
wlp2s0	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	2529	2529 (100.0%)	—
Time span, s	10.248	10.248	—
Average pps	246.8	246.8	—
Average packet size, B	566	566	—
Bytes	1430374	1430374 (100.0%)	0
Average bytes/s	139 k	139 k	—
Average bits/s	1,116 k	1,116 k	—

```
Reverse mode, remote host ping.online.net is sending
[  5] local 10.184.7.249 port 34948 connected to 62.210.18.40 port 5208
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datag
rams
[  5]   0.00-1.00   sec   128 KBytes  1.05 Mbits/sec  10215.768 ms  0/251 (0%)
[  5]   1.00-2.00   sec   128 KBytes  1.05 Mbits/sec  0.020 ms  0/250 (0%)
[  5]   2.00-3.00   sec   128 KBytes  1.05 Mbits/sec  0.110 ms  0/250 (0%)
[  5]   3.00-4.00   sec   128 KBytes  1.05 Mbits/sec  0.055 ms  0/250 (0%)
[  5]   4.00-5.00   sec   128 KBytes  1.05 Mbits/sec  0.092 ms  0/250 (0%)
[  5]   5.00-6.00   sec   128 KBytes  1.05 Mbits/sec  1.520 ms  0/250 (0%)
[  5]   6.00-7.00   sec   128 KBytes  1.05 Mbits/sec  0.029 ms  0/251 (0%)
[  5]   7.00-8.00   sec   128 KBytes  1.05 Mbits/sec  0.040 ms  0/250 (0%)
[  5]   8.00-9.00   sec   128 KBytes  1.05 Mbits/sec  0.023 ms  0/250 (0%)
[  5]   9.00-10.00  sec   128 KBytes  1.05 Mbits/sec  0.107 ms  0/250 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datag
rams
[  5]   0.00-10.00  sec  1.28 MBytes  1.07 Mbits/sec  0.000 ms  0/2502 (0%)  sen
der
[  5]   0.00-10.00  sec  1.25 MBytes  1.05 Mbits/sec  0.107 ms  0/2502 (0%)  rec
eiver

iperf Done.
zero2706@zero2706-Inspiron-14-5408:~$
```

3. HTTP Task

In this task we will be analysing the given HTTP packets given in http2-h2c.pcap file.

1. Here the total packets are **10** and upon using Statistics analysis on wireshark we can see that the total number of **HTTP/1.1 packets are 2** which are the 1 GET request packet and 1 switching protocol response packet. Similarly the number of **HTTP/2 packets are 9**.
2. The total number of **HTTP/2 packets** that exchanged before the first object is fetched are **4**.
3. The main difference observed between the headers of HTTP/1.1 and HTTP/2 headers is that HTTP/2 pack-

ets contain some additional information like TYPE HEADER and different type of information about the header some of which are absent in HTTP/1.1 packet(like x-backend-header , x-frame-options, etc).

4. PING Task

In this task we are going to exchange IP packets between host and remote server using ping command and examine the packets. **Note** for ping command with packet size of 3500, I was facing a 100% packet loss hence I used 1000 as my packet size to ping.

1. While executing this command the number of IP packets transferred were 10 of which 5 were requests and 5 were responses from the remote server ping-ams1.online.net
2. The size of each ping request sent from host to remote server is **1042 bytes**.
3. The table diagram for each ping request is shown below:-

Ping Request	Fragmented	Time of Sending	Length of Packet	Time of Receiving	Response Fragmented	Length of Data
1	No	0.000 seconds	1042 bytes	0.2392 seconds	No	992 bytes
2	No	1.00183 seconds	1042 bytes	1.2616 seconds	No	992 bytes
3	No	2.00366 seconds	1042 bytes	2.18317 seconds	No	992 bytes
4	No	3.00444 seconds	1042 bytes	3.311102 seconds	No	992 bytes
5	No	4.00610 seconds	1042 bytes	4.2320 seconds	No	992 bytes

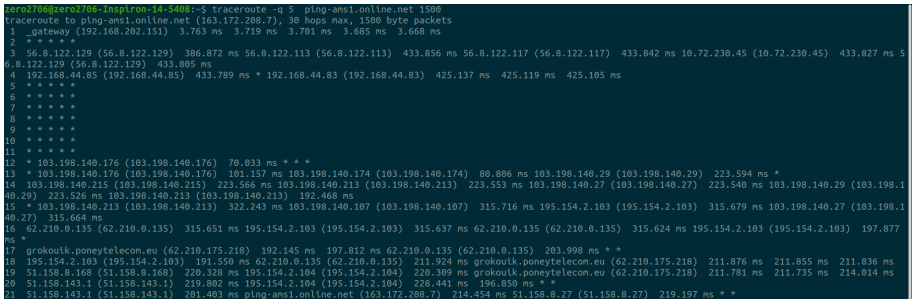
5. TRACEROUTE Task

In this task we are going to analyze the ip packets when trying to connect to ping-ams1.online.net remote server while analyzing the hops via traceroute command.

1. The number of hops that are involved in finding the route from local host to ping-ams1.online.net are **21**.
2. The total number of IP packets exchanged are 167 and out of them the number of packets sent from client

are **112** and the number of packets sent from different hop servers to client are **55**. The table for the different ip packets from various hops and remote server(mentioned as server in the diagram) is shown below(though not all the hop servers are included cause there number was big).

IP Address	Server/Hop/Client	Packets Sent
192.168.202.67	Client	112
192.168.202.151	Hop	5
56.8.122.129	Hop	2
56.8.122.113	Hop	1
56.8.122.117	Hop	1
10.72.230.45	Hop	1
192.168.44.85	Hop	1
192.168.44.83	Hop	3
103.198.140.176	Hop	2
103.198.140.174	Hop	1
163.172.208.7	Server	5



3. Here in all the IP datagrams sent by my client/host, the source port and destination port always changes from packet to packet while the field of Data remains same for each packet. The Data field remains the same because the same packet data is used to trace path by traceroute

while since there are multiple processes going on that's why we have to use multiple ports for packet transfer.