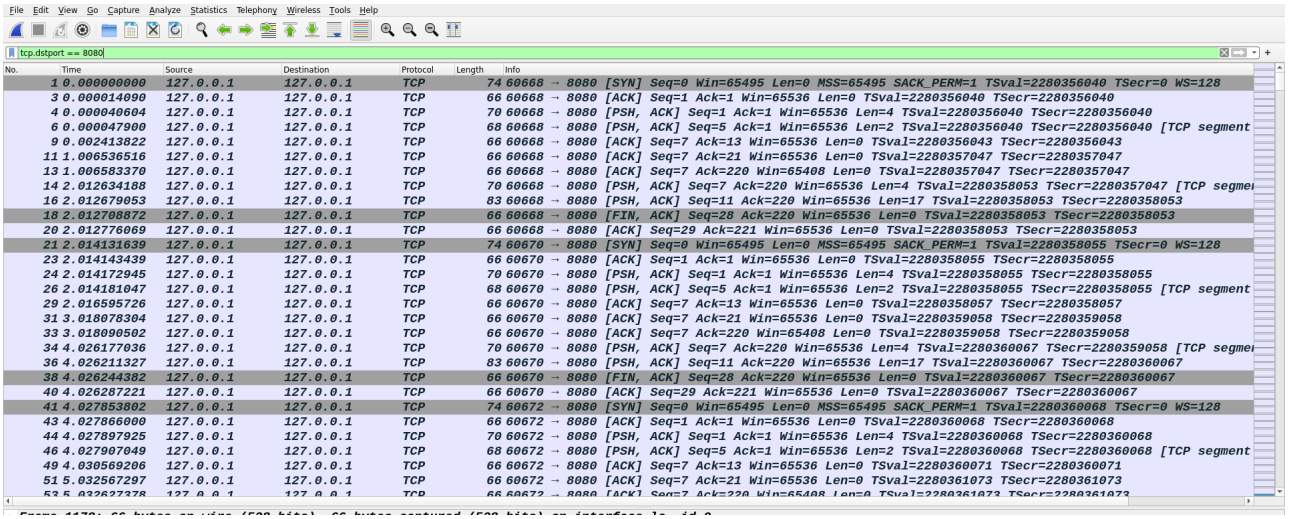
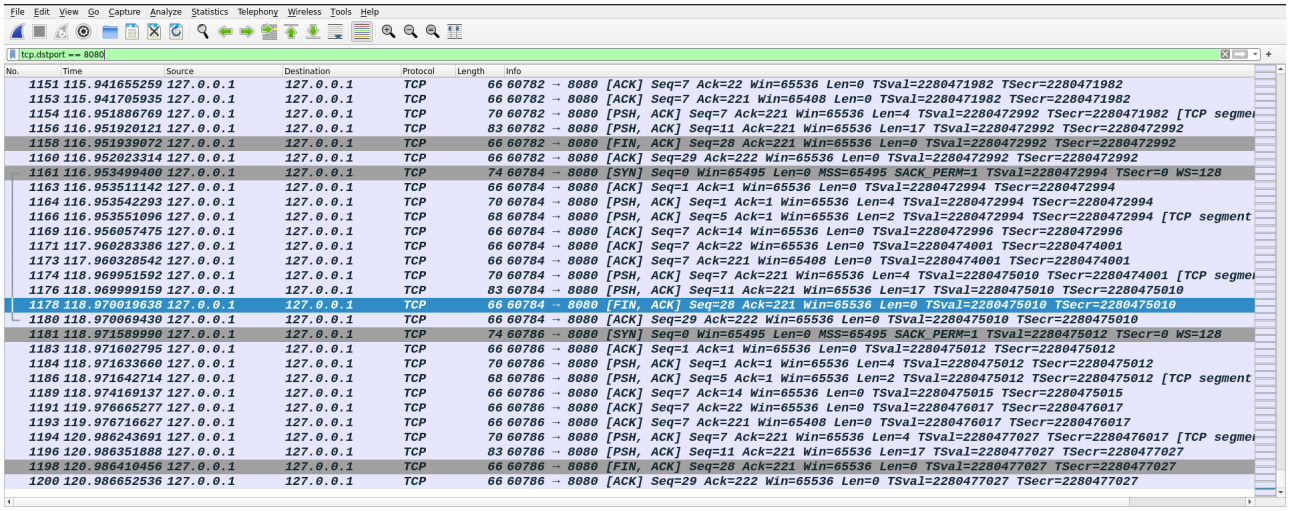
CSE232 | Assignment 2

Report

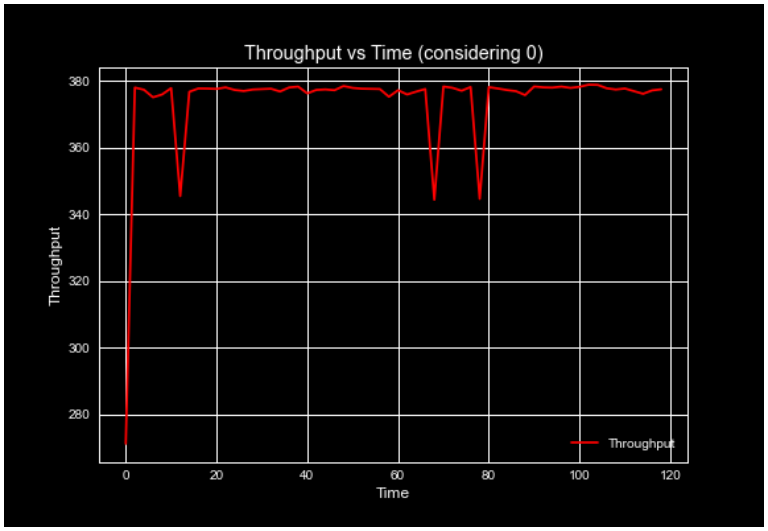
Question 1

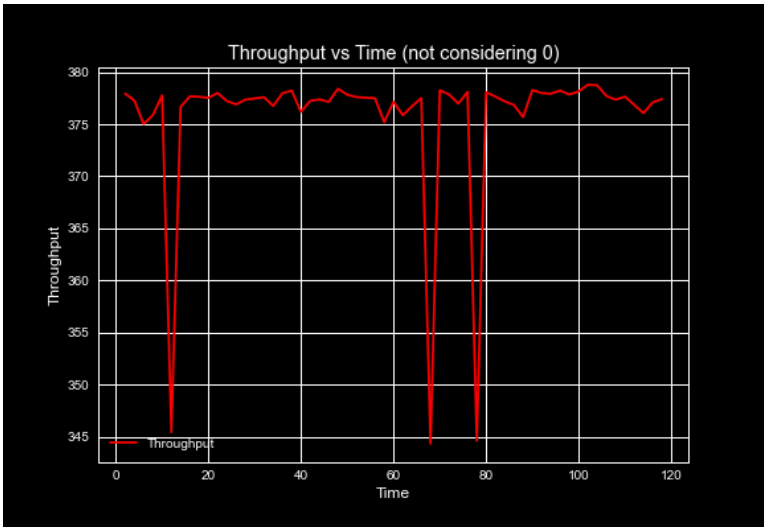
- The client-socket program implemented in the earlier assignment was run for 2 mins by an external script.
- The packets generated were observed using wireshark.
- Those packets were filtered and their data was exported in a csv file.
- To filter the data, the command tcp.dstport == 8080 was used.
- My socket server was running at the port 8080.
- The data looked like:





- The images shown above were captured at the beginning and the end of the 120 sec duration
- Two graphs were plotted using the data generated in the csv file(one with starting from origin and one not starting from origin, rather starting from 2).
- The points on the x axis were approximated to contain the average throughput generated
- The graphs looked like:

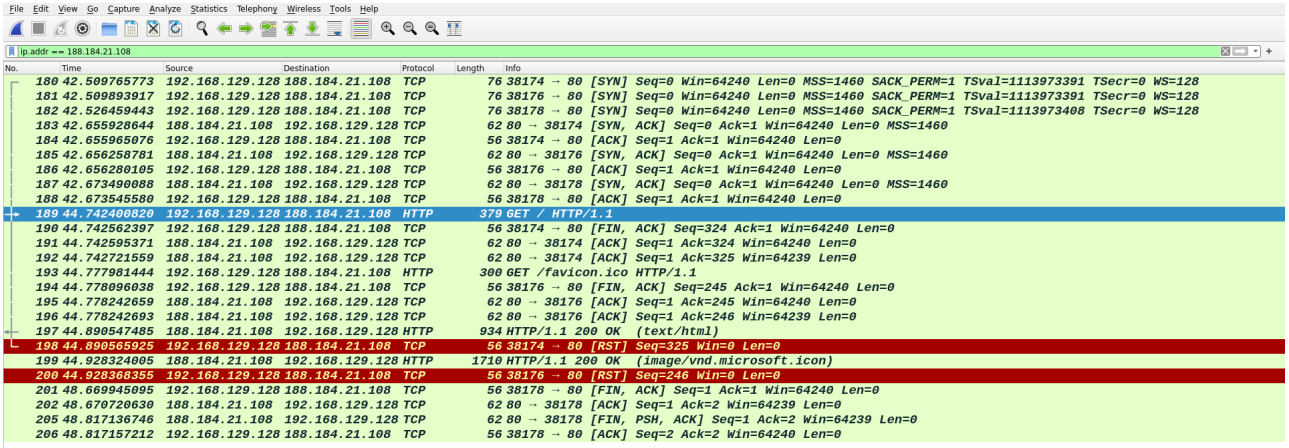




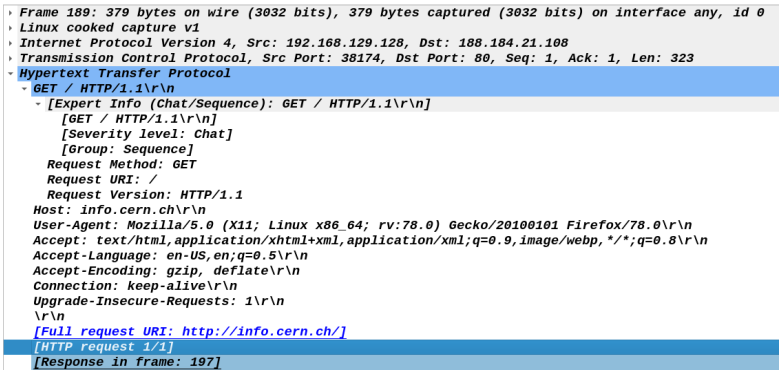
- The data, code to run the files for 2 mins and the code to generate the graph are also present in the folder under the following names:
    - Data Accumulated: ServerSideThroughputData.csv
    - File to run the client repeatedly for 120 secs: runClient.c
    - File to generate graph: createGraph.ipynb

Question 2

- Communication between my machine and the web server hosting info.cern.ch as captured on wireshark.
- The filter used was: ip.addr == 188.184.21.108

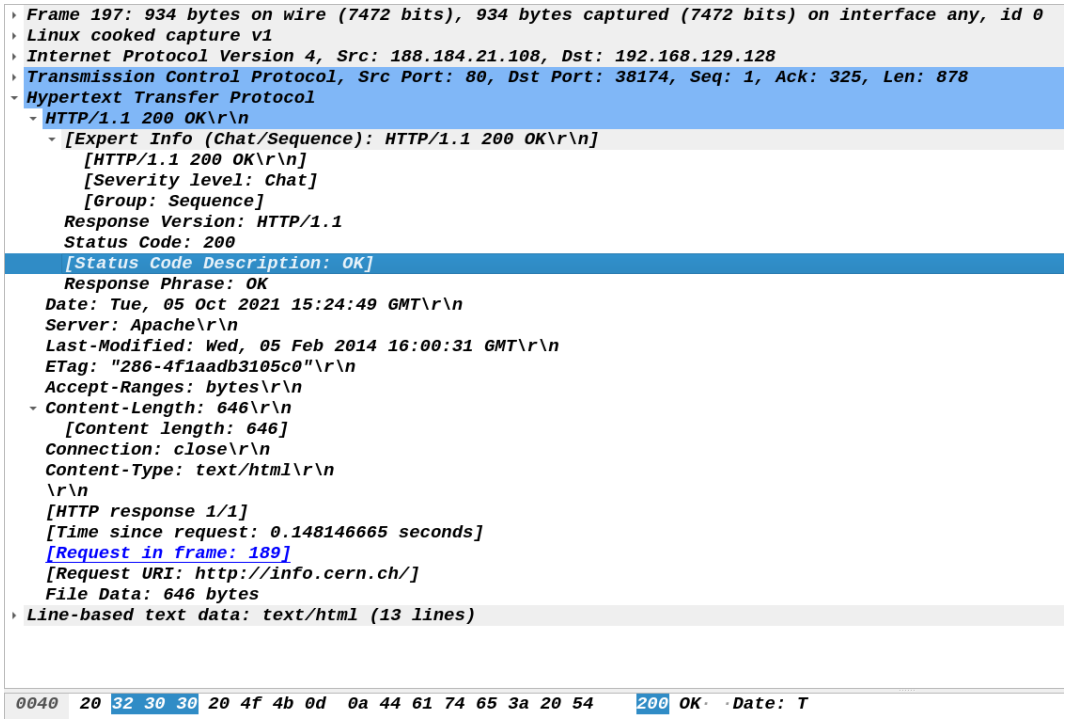


- Filter for getting the http captures: ip.addr == 188.184.21.108 && http
- Request Message for retrieving the webpage:



  - HTTP Packet Type: Request
  - HTTP Request Type: GET
  - User Agent Type: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
  - HTTP Request Packet URL: /
    - Request URI: /
    - Full Request URI: http://info.cern.ch/
  - Name and Version of Webserver: Request packet does not carry the information regarding the sender's webserver, this is only contained in the response packet.

- Response Message:

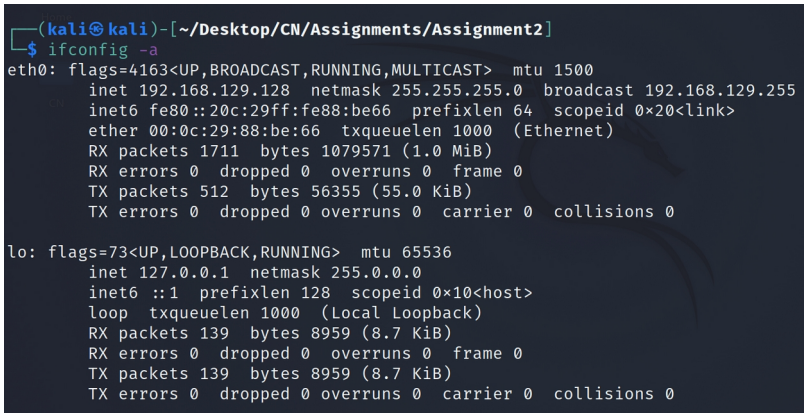


  - HTTP Packet Type: Response
  - HTTP Response Code: 200
  - HTTP Response Description:
    - Status Code Description: OK
    - Response Phrase: OK
    - Response Code 200 means success, that is the resource has been fetched and is transmitted in the message body.

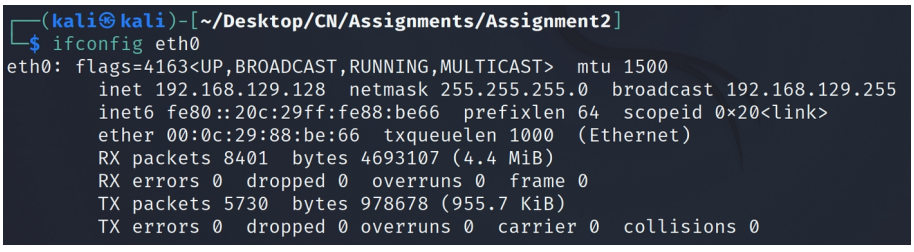
Question 3

a) IP address using ifconfig

- Ifconfig is used to configure the kernel-resident network interfaces.
- It is used at boot time to setup interfaces as necessary.
- After that, it is usually only needed when debugging or when system tuning is needed.
- To look at all the interfaces, both up and down, I can use -a flag:



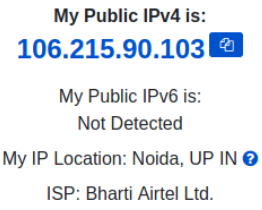
- The above shows that I have 2 interfaces:
  i. eth0
  ii. lo
- The lo or the loopback interface is for the local host, so we can ignore that.
- The other, that is eth0 is the one through which my machine connects with the internet.
- So we can have a closer look at it by giving its name as parameter to the ifconfig command:



- Through the above, we can see that the ip addresses are as follows:
  i. IPv4 address: 192.168.129.128
  ii. IPv6 address: fe80::20c:29ff:fe88:be66

b) IP address using whatismyip.com

- The following screenshot was taken after visiting the above website:



- We can clearly see that the IP addresses obtained by ifconfig and the one obtained by visiting the above website are different.
- This is because the IP address shown on the above website is my public/external IP address, whereas the IP address obtained from ifcongfig is my private/internal IP address.
- External IP addresses are usually assigned to router or modem instead of being assigned to each and every machine in that network, where as internal IP address is assigned to each machine.

Question 4

a) Command to send a packet with mtu 3000

- The command used is: ping -c 1 -s 3000 -M do www.google.com
- The flags used are as follows:
  i. -c 1: This makes sure that only 1 packet is sent
  ii. -s 3000: This makes sure that the size of packet is 3000
  iii. -M do: This makes sure that the packet is not fragmented.
- Command in action:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ping -c 1 -s 3000 -M do www.google.com
PING www.google.com (142.250.71.4) 3000(3028) bytes of data.
ping: local error: message too long, mtu=1500

--- www.google.com ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

- Clearly, the test failed.
- This is because, even though we sent the packet with mtu 3000, it was not trasmitted successfully as the network itself has some limit which is less than 3000, thus showing us the error ‘message too long’.
- Even though our machine allowed us to send the packet, the network itself didn’t allow such big a packet to transmit, thus resulting in the error along with the packet loss.

b) The command to display all active tcp connection with pids.

- The command being used is: netstat -atp
- The flags used are as follows:
  i. -a: Show both listening and non-listening sockets.
  ii. -t: Show the tcp connections
  iii. -p: Show the PID and name of the program to which each socket belongs.
- The command in action:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ netstat -atp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.129.128:37714   webafs706.cern.ch:http  TIME_WAIT   -
tcp        0      0 192.168.129.128:36248   server-13-35-191-:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:39558   del11s12-in-f14.1:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:37776   del11s09-in-f3.1e:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:58850   del03s06-in-f4.1e:https ESTABLISHED 2070/x-www-browser
tcp        0      1 192.168.129.128:51556   maa05s05-in-f2.1e:https SYN_SENT    2070/x-www-browser
tcp        0      0 192.168.129.128:43134   maa03s46-in-f3.1e1:http ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:46216   ws-in-f94.1e100.n:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:44796   maa05s19-in-f14.1:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:39556   del11s12-in-f14.1:https TIME_WAIT   -
tcp        0      0 192.168.129.128:43130   maa03s46-in-f3.1e1:http ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:36246   server-13-35-191-:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:43718   maa05s20-in-f3.1e:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:39884   maa05s18-in-f22.1:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:53280   117.18.237.29:http      ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:43158   maa03s46-in-f3.1e1:http ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:39560   del11s12-in-f14.1:https TIME_WAIT   -
tcp        0      0 192.168.129.128:58318   del11s05-in-f10.1:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:36936   ec2-35-162-54-217:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:58318   del11s05-in-f10.1:https ESTABLISHED 2070/x-www-browser
tcp        0      0 192.168.129.128:36936   ec2-35-162-54-217:https ESTABLISHED 2070/x-www-browser
```

Question 5

a) Authoritative result in nslookup

- For this question, I will be getting the information pertaining to www.facebook.com (question done after the long period when the site was down)
- If we directly do nslookup for facebook.com, we would most definitely get the result from some cache on the pathway.
- This is because facebook.com is a pretty common website and the probability of its IP Address being cached somewhere on the path is pretty high.

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ nslookup facebook.com
Server:         192.168.129.2
Address:        192.168.129.2#53

Non-authoritative answer:
Name:   facebook.com
Address: 157.240.198.35
Name:   facebook.com
Address: 2a03:2880:f12f:83:face:b00c:0:25de
```

- As shown in the above image, we are getting non-authoritative results.
- Since we want to get info from authoritative servers, we need to find one of the authoritative nameservers for the website.
- This can be done by adding the flag -type=ns as shown below.

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ nslookup -type=ns facebook.com
Server:         192.168.129.2
Address:        192.168.129.2#53

Non-authoritative answer:
facebook.com    nameserver = d.ns.facebook.com.
facebook.com    nameserver = c.ns.facebook.com.
facebook.com    nameserver = b.ns.facebook.com.
facebook.com    nameserver = a.ns.facebook.com.

Authoritative answers can be found from:
d.ns.facebook.com       has AAAA address 2a03:2880:f1fd:c:face:b00c:0:35
d.ns.facebook.com       internet address = 185.89.219.12
c.ns.facebook.com       has AAAA address 2a03:2880:f1fc:c:face:b00c:0:35
c.ns.facebook.com       internet address = 185.89.218.12
b.ns.facebook.com       has AAAA address 2a03:2880:f0fd:c:face:b00c:0:35
b.ns.facebook.com       internet address = 129.134.31.12
a.ns.facebook.com       has AAAA address 2a03:2880:f0fc:c:face:b00c:0:35
a.ns.facebook.com       internet address = 129.134.30.12
```

- Let us call nslookup on one of the authoritative webservers:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ nslookup a.ns.facebook.com
Server:         192.168.129.2
Address:        192.168.129.2#53

Non-authoritative answer:
Name:   a.ns.facebook.com
Address: 129.134.30.12
Name:   a.ns.facebook.com
Address: 2a03:2880:f0fc:c:face:b00c:0:35
```

- Now since we have the authoritative nameserver of the website with us, let us call facebook.com directly from the above nameserver, thus getting the authoritative response as desired.

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ nslookup facebook.com a.ns.facebook.com
Server:         a.ns.facebook.com
Address:        129.134.30.12#53

Name:   facebook.com
Address: 157.240.239.35
Name:   facebook.com
Address: 2a03:2880:f144:181:face:b00c:0:25de
```

b) Finding ttl for www.facebook.com

- We can directly find the ttl(time to live) using the nslookup command.
- For this we have to use the -debug flag, and we would be done.

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ nslookup -debug facebook.com
Server:         192.168.129.2
Address:        192.168.129.2#53

------------
    QUESTIONS:
        facebook.com, type = A, class = IN
    ANSWERS:
    ->  facebook.com
        internet address = 157.240.16.35
        ttl = 5
    AUTHORITY RECORDS:
    ADDITIONAL RECORDS:
------------
Non-authoritative answer:
Name:   facebook.com
Address: 157.240.16.35
------------
    QUESTIONS:
        facebook.com, type = AAAA, class = IN
    ANSWERS:
    ->  facebook.com
        has AAAA address 2a03:2880:f12f:83:face:b00c:0:25de
        ttl = 5
    AUTHORITY RECORDS:
    ADDITIONAL RECORDS:
------------
Name:   facebook.com
Address: 2a03:2880:f12f:83:face:b00c:0:25de
```

- As evident from the above image, the ttl or the time to live for www.facebook.com on the local dns is 5 seconds. The entry would expire after this much time.
- Please note that the ttl for both IPv4 and IPv6 is 5 seconds.
- The same can be found out using the dig command as follows:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig facebook.com

; <<>> DiG 9.16.15-Debian <<>> facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26006
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;facebook.com.                  IN      A

;; ANSWER SECTION:
facebook.com.           5       IN      A       157.240.198.35

;; Query time: 56 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 13:48:20 EDT 2021
;; MSG SIZE  rcvd: 57
```

- As it is evident, the dig command also states that the ttl is 5 seconds.

Question 6

a) Running traceroute to iiith.ac.in

- Before running the traceroute command, certain rules were added to the system's firewall for ICMP time exceeded packets.
- Also flag -I was used for using ICMP ECHO for probes.
- The following is the traceroute:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ sudo traceroute -I www.iiith.ac.in
traceroute to www.iiith.ac.in (196.12.53.50), 30 hops max, 60 byte packets
 1  192.168.129.2 (192.168.129.2)  0.214 ms  0.109 ms  0.099 ms
 2  dsldevice.lan (192.168.1.1)  1.656 ms  1.764 ms  1.955 ms
 3  abts-north-static-236.220.160.122.airtelbroadband.in (122.160.220.236)  6.097 ms  6.035 ms  5.995 ms
 4  125.18.20.101 (125.18.20.101)  5.928 ms  5.883 ms  5.831 ms
 5  116.119.61.117 (116.119.61.117)  44.984 ms  44.946 ms  45.271 ms
 6  49.44.220.188 (49.44.220.188)  45.727 ms  45.710 ms  45.622 ms
 7  * * *
 8  115.242.184.26.static.jio.com (115.242.184.26)  44.629 ms  44.780 ms  44.696 ms
 9  196.12.34.76 (196.12.34.76)  47.951 ms  48.233 ms  48.155 ms
10  196.12.53.50 (196.12.53.50)  53.121 ms  53.568 ms  53.561 ms
```

- Pings for intermediate hosts(including the final one) with their average latencies are as follows(for computing the average, 10 packets were sent each time):

```
--- 192.168.129.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9201ms
rtt min/avg/max/mdev = 0.368/0.561/0.665/0.081 ms
```

- 192.168.129.2: 0.561 ms

```
--- 192.168.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9036ms
rtt min/avg/max/mdev = 3.286/28.260/94.118/33.274 ms
```

- 192.168.1.1: 28.260 ms

```
--- 122.160.220.236 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9036ms
rtt min/avg/max/mdev = 6.762/34.856/113.783/36.373 ms
```

- 122.160.220.236: 34.856 ms

```
--- 125.18.20.101 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9035ms
rtt min/avg/max/mdev = 6.554/29.027/81.607/26.714 ms
```

- 125.18.20.101: 29.027 ms

```
--- 116.119.61.117 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9060ms
rtt min/avg/max/mdev = 45.900/48.657/58.364/3.558 ms
```

- 116.119.61.117: 48.657 ms

```
--- 49.44.220.188 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9031ms
rtt min/avg/max/mdev = 44.599/70.159/121.400/22.817 ms
```

- 49.44.220.188: 70.159 ms
- The next one was full of stars only, so ignoring that one

```
--- 115.242.184.26 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9042ms
rtt min/avg/max/mdev = 46.071/64.257/135.109/31.614 ms
```

- 115.242.184.26: 64.257 ms

```
--- 196.12.34.76 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9030ms
rtt min/avg/max/mdev = 48.516/55.070/83.944/10.728 ms
```

- 196.12.34.76: 55.070 ms

```
--- 196.12.53.50 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9035ms
rtt min/avg/max/mdev = 53.737/61.841/94.495/12.410 ms
```

- 196.12.53.50: 61.841 ms
- The above one is the last hop and is the IP address for www.iiith.ac.in

b) Ping messages to www.iiit.ac.in

```
--- www.iiit.ac.in ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99658ms
rtt min/avg/max/mdev = 53.282/67.744/189.571/30.110 ms
```

- As it can be seen from the image above, the average latency for pinging www.iiit.ac.in is 67.744 ms

c) Comparing sum of intermediate ping latencies and average latency of pinging www.iiit.ac.in

- Sum of intermediate ping latencies:
  - 0.561 + 28.260 + 34.856 + 29.027 + 48.657 + 70.159 + 64.257 + 55.070 = 330.847 ms
- Average latency obtained in part b:
  - 67.744 ms
- Before making any kind of comparison, it is important to note that in the interest of time, the average latencies of the intermediate hosts has been calculated over 10 entries, while the final latency has been calculated over 100 entries.
- As compared to 100, 10 is a very small size and even 1 very high or very low value can affect the average considerably.
- Keeping the above points in mind, we can safely make the statement that sum of ping latencies of intermediate hosts would always be greater than the average latency of the final host.
- This is because, when we use the ping command, we are in actuality calculating the RTT from our machine to the server whose IP address has been entered.
- So for each of the intermediate host, we are again and again adding the time of travel which is being added only once for the final host.
- Along with that when we calculate the time for each host, it is more or less in the sense that our packet reached the said host, it was processed there and the corresponding ack was sent to us.
- But while pinging the final host, even though these all hosts are encountered, still they are encountered in some sort of pipeline which considerably reduces their time of processing.
- In the above manner only the node which is making the bottleneck matters the most while other processing times for the packets are dealt with in parallel with the bottleneck one.
- As a result, the sum of intermediate host latencies is lesser than the average final host frequency.

d) Comparing the maximum intermediate host ping latency with the final host ping latency.

- Maximum Intermediate Host ping latency:
  - 70.159 ms
- Average final host latency:
  - 67.744 ms
- Keeping in mind the point on averages discussed in the previous part, it would be safe to say that both the latencies are more or less similar.
- Again a few outlying values would affect the average a lot but it can be seen that if we

repeat the above experimentation multiple times, the latencies would be alike.

- This is because the one with the maximum latency acts as the bottleneck for the path.
- It is the host where maximum amount of time is spent by the traveling packets as the other hosts are in some sort of a pipeline, therefore the time spent there does not make a huge impact on the final latency.

e) Reverse DNS

- For reverse dns, dig command with -x flag is used.
- The following are the results with their host names and aliases:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 192.168.129.2

; <<>> DiG 9.16.15-Debian <<>> -x 192.168.129.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 30520
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;2.129.168.192.in-addr.arpa.    IN      PTR

;; AUTHORITY SECTION:
168.192.in-addr.arpa.   5       IN      SOA     prisoner.iana.org. hostmaster.root-servers.org. 1 604800 60 604
800 604800

;; Query time: 196 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:40:44 EDT 2021
;; MSG SIZE  rcvd: 132
```

- 192.168.129.2: prisoner.iana.org, hostmaster.root-servers.org

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 192.168.1.1

; <<>> DiG 9.16.15-Debian <<>> -x 192.168.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16015
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;1.1.168.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
1.1.168.192.in-addr.arpa. 5     IN      PTR     dsldevice.lan.

;; Query time: 64 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:41:23 EDT 2021
;; MSG SIZE  rcvd: 69
```

- 192.168.1.1: dsldevice.lan

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 122.160.220.236

; <<>> DiG 9.16.15-Debian <<>> -x 122.160.220.236
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47806
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 4096
;; QUESTION SECTION:
;236.220.160.122.in-addr.arpa.  IN      PTR

;; ANSWER SECTION:
236.220.160.122.in-addr.arpa. 5 IN      PTR     abts-north-static-236.220.160.122.airtelbroadband.in.

;; Query time: 8 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:41:56 EDT 2021
;; MSG SIZE  rcvd: 123
```

- 122.160.220.236: abts-north-static-236.220.160.122.airtelbroadband.in

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 125.18.20.101

; <<>> DiG 9.16.15-Debian <<>> -x 125.18.20.101
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 30722
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;101.20.18.125.in-addr.arpa.    IN      PTR

;; AUTHORITY SECTION:
20.18.125.in-addr.arpa. 5       IN      SOA     infoblox.localdomain. root.dnsdel.mantraonline.com. 2014022210 3600 3600 1209600 3600

;; Query time: 16 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:42:45 EDT 2021
;; MSG SIZE  rcvd: 139
```

- 125.18.20.101: infoblox.localdomain, root.dnsdel.mantraonline.com

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 116.119.61.117

; <<>> DiG 9.16.15-Debian <<>> -x 116.119.61.117
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 53727
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;117.61.119.116.in-addr.arpa.   IN      PTR

;; AUTHORITY SECTION:
116.in-addr.arpa.       5       IN      SOA     ns.apnic.net. read-txt-record-of-zone-first-dns-admin.apnic.net. 306093384 7200 1800 604800 3600

;; Query time: 235 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:43:47 EDT 2021
;; MSG SIZE  rcvd: 144
```

- 116.119.61.117: ns.apnic.net, read-txt-record-of-zone-first-dns-admin.apnic.net

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 49.44.220.188

; <<>> DiG 9.16.15-Debian <<>> -x 49.44.220.188
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 52969
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;188.220.44.49.in-addr.arpa.    IN      PTR

;; AUTHORITY SECTION:
44.49.in-addr.arpa.     5       IN      SOA     ns1.44.49.in-addr.arpa. jiodns.admins\@.ril.com. 2021092711 10800 900 1209600 86400

;; Query time: 2035 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:44:33 EDT 2021
;; MSG SIZE  rcvd: 117
```

- 49.44.220.188: ns1.44.49.in-addr.arpa jiodns.admins\@.ril.com
- For the 7th entry, only *** were available, so the corresponding host name couldn’t be found out.

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 115.242.184.26

; <<>> DiG 9.16.15-Debian <<>> -x 115.242.184.26
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9851
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;26.184.242.115.in-addr.arpa.   IN      PTR

;; ANSWER SECTION:
26.184.242.115.in-addr.arpa. 5  IN      PTR     115.242.184.26.static.jio.com.

;; Query time: 19 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:45:14 EDT 2021
;; MSG SIZE  rcvd: 99
```

- 115.242.184.26: 115.242.184.26.static.jio.com

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 196.12.34.76

; <<>> DiG 9.16.15-Debian <<>> -x 196.12.34.76
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 1443
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;76.34.12.196.in-addr.arpa.     IN      PTR

;; AUTHORITY SECTION:
34.12.196.in-addr.arpa. 5       IN      SOA     ns.stph.net. hyd.itsm.stpi.in.34.12.196.in-addr.arpa. 2016090101 3600 300 360000 14400

;; Query time: 547 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:45:43 EDT 2021
;; MSG SIZE  rcvd: 118
```

- 192.12.34.76: ns.stph.net, hyd.itsm.stpi.in.34.12.196.in-addr.arpa

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ dig -x 196.12.53.50

; <<>> DiG 9.16.15-Debian <<>> -x 196.12.53.50
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 28052
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 4096
;; QUESTION SECTION:
;50.53.12.196.in-addr.arpa.     IN      PTR

;; AUTHORITY SECTION:
53.12.196.in-addr.arpa. 5       IN      SOA     ns.stph.net. hyd.itsm.stpi.in. 2017100601 3600 300 360000 14400

;; Query time: 384 msec
;; SERVER: 192.168.129.2#53(192.168.129.2)
;; WHEN: Tue Oct 05 10:46:18 EDT 2021
;; MSG SIZE  rcvd: 117
```

- 196.12.53.50: ns.stph.net, hyd.itsm.stpi.in
- The last one is the final hop or the final destination.

Question 7

- In this question, I have sent 10 packets for each command to eliminate any kind of redundancy throughout.
- 127.0.0.1 refers to the loopback ip address.
- In other words it is the localhost.
- So when we are sending the ping, we are actually sending it to the localhost itself.
- The following image shows that the ping command is working

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ping -c 10 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.067 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.073 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.070 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.068 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.205 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.075 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.089 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.072 ms

--- 127.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9223ms
rtt min/avg/max/mdev = 0.053/0.085/0.205/0.040 ms
```

- To make the ping command fail, I can do that in 2 ways:
  1. Increasing the size of packet to be sent via ping by a lot.
     - This results in 100% packet loss as since the size is too large, either the response to be received exceeds the waiting time or the server itself is not able to carry such a big packet, and the server marks it as lost packet.

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ping -c 10 -s 100000 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 100000(100028) bytes of data.

--- 127.0.0.1 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9222ms
```

     - This solution would work for any ip address and is not specific to the one in question.
  2. Another, and more reliable solution is to deactivate the loopback interface.
     - Since 127.0.0.1 is the loopback address, if we deactivate the loopback interface, the ping would not be sent, thus resulting in it's failure.
     - Please note that this solution, unlike the first one, works only for the IP in question and not any other one.
     - Initial ifconfig state:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.129.128  netmask 255.255.255.0  broadcast 192.168.129.255
        inet6 fe80::20c:29ff:fe88:be66  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:88:be:66  txqueuelen 1000  (Ethernet)
        RX packets 913  bytes 915682 (894.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132  bytes 15038 (14.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 119  bytes 7959 (7.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 119  bytes 7959 (7.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Disabling the loopback interface:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ sudo ifconfig lo down
```

- ifconfig after disabling the loopback interface:

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.129.128  netmask 255.255.255.0  broadcast 192.168.129.255
        inet6 fe80::20c:29ff:fe88:be66  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:88:be:66  txqueuelen 1000  (Ethernet)
        RX packets 1616  bytes 1073589 (1.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 510  bytes 55971 (54.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Now sending the ping

```
┌──(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ping -c 10 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9218ms
```