

CSE232 | Assignment 2

Report

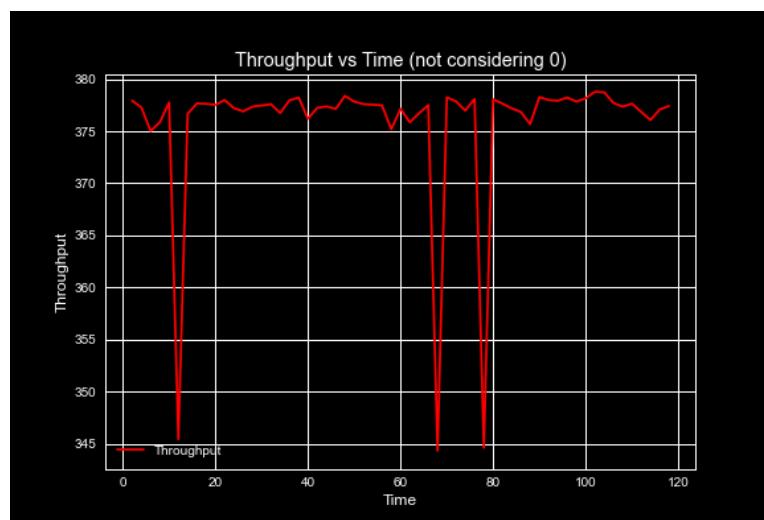
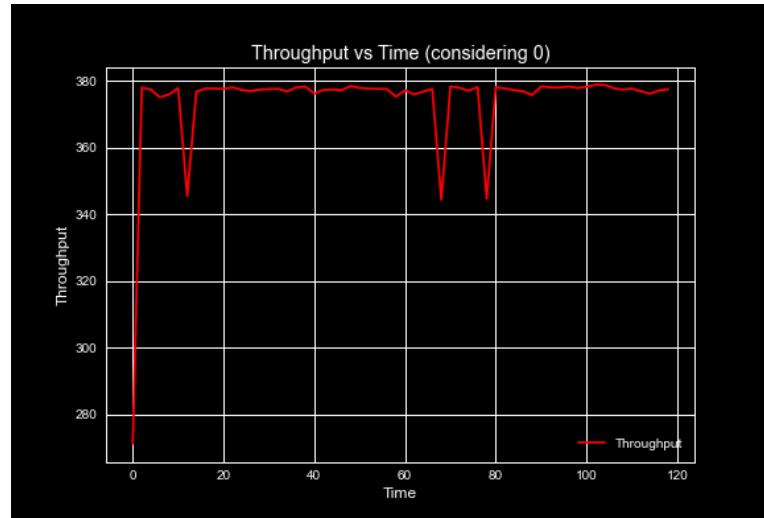
Question 1

- 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:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	127.0.0.1	127.0.0.1	TCP	74	60668 - 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2280356040 TSecr=0 WS=128
3	0.0000140999	127.0.0.1	127.0.0.1	TCP	66	60668 - 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2280356040 TSecr=2280356040
4	0.000040604	127.0.0.1	127.0.0.1	TCP	70	60668 - 8080 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4 TSval=2280356040 TSecr=2280356040
6	0.000047900	127.0.0.1	127.0.0.1	TCP	68	60668 - 8080 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=2 TSval=2280356040 TSecr=2280356040 [TCP segment
9	0.002413822	127.0.0.1	127.0.0.1	TCP	66	60668 - 8080 [ACK] Seq=7 Ack=13 Win=65536 Len=0 TSval=2280356043 TSecr=2280356043
11	1.006536516	127.0.0.1	127.0.0.1	TCP	66	60668 - 8080 [ACK] Seq=7 Ack=21 Win=65536 Len=0 TSval=2280357047 TSecr=2280357047
13	1.006583370	127.0.0.1	127.0.0.1	TCP	66	60668 - 8080 [ACK] Seq=7 Ack=220 Win=65408 Len=0 TSval=2280357047 TSecr=2280357047
14	2.012634188	127.0.0.1	127.0.0.1	TCP	70	60668 - 8080 [PSH, ACK] Seq=7 Ack=220 Win=65536 Len=4 TSval=2280358053 TSecr=2280358053 [TCP segment
16	2.012679053	127.0.0.1	127.0.0.1	TCP	83	60668 - 8080 [PSH, ACK] Seq=11 Ack=220 Win=65536 Len=17 TSval=2280358053 TSecr=2280358053
18	2.012708872	127.0.0.1	127.0.0.1	TCP	66	60668 - 8080 [FIN, ACK] Seq=20 Ack=220 Win=65536 Len=0 TSval=2280358053 TSecr=2280358053
20	2.012776069	127.0.0.1	127.0.0.1	TCP	66	60668 - 8080 [ACK] Seq=29 Ack=221 Win=65536 Len=0 TSval=2280358053 TSecr=2280358053
21	2.014131639	127.0.0.1	127.0.0.1	TCP	74	60670 - 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2280358055 TSecr=0 WS=128
23	2.014143439	127.0.0.1	127.0.0.1	TCP	66	60670 - 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2280358055 TSecr=2280358055
24	2.014172945	127.0.0.1	127.0.0.1	TCP	70	60670 - 8080 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4 TSval=2280358055 TSecr=2280358055
26	2.014181047	127.0.0.1	127.0.0.1	TCP	68	60670 - 8080 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=2 TSval=2280358055 TSecr=2280358055 [TCP segment
29	2.016595726	127.0.0.1	127.0.0.1	TCP	66	60670 - 8080 [ACK] Seq=7 Ack=13 Win=65536 Len=0 TSval=2280358057 TSecr=2280358057
31	3.018078304	127.0.0.1	127.0.0.1	TCP	66	60670 - 8080 [ACK] Seq=7 Ack=21 Win=65536 Len=0 TSval=2280359058 TSecr=2280359058
33	3.018090502	127.0.0.1	127.0.0.1	TCP	66	60670 - 8080 [ACK] Seq=7 Ack=220 Win=65536 Len=4 TSval=2280359058 TSecr=2280359058
34	4.026177036	127.0.0.1	127.0.0.1	TCP	70	60670 - 8080 [PSH, ACK] Seq=7 Ack=220 Win=65536 Len=4 TSval=2280360067 TSecr=2280359058 [TCP segment
36	4.026211327	127.0.0.1	127.0.0.1	TCP	83	60670 - 8080 [PSH, ACK] Seq=11 Ack=220 Win=65536 Len=17 TSval=2280360067 TSecr=2280360067
38	4.026244382	127.0.0.1	127.0.0.1	TCP	66	60670 - 8080 [FIN, ACK] Seq=20 Ack=220 Win=65536 Len=0 TSval=2280360067 TSecr=2280360067
40	4.0262687221	127.0.0.1	127.0.0.1	TCP	66	60670 - 8080 [ACK] Seq=29 Ack=221 Win=65536 Len=0 TSval=2280360067 TSecr=2280360067
41	4.027853892	127.0.0.1	127.0.0.1	TCP	74	60672 - 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2280360068 TSecr=0 WS=128
43	4.027866008	127.0.0.1	127.0.0.1	TCP	66	60672 - 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2280360068 TSecr=2280360068
44	4.027897925	127.0.0.1	127.0.0.1	TCP	70	60672 - 8080 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4 TSval=2280360068 TSecr=2280360068
46	4.027907049	127.0.0.1	127.0.0.1	TCP	68	60672 - 8080 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=2 TSval=2280360068 TSecr=2280360068 [TCP segment
49	4.030569206	127.0.0.1	127.0.0.1	TCP	66	60672 - 8080 [ACK] Seq=7 Ack=13 Win=65536 Len=0 TSval=2280360971 TSecr=2280360971
51	5.032567297	127.0.0.1	127.0.0.1	TCP	66	60672 - 8080 [ACK] Seq=7 Ack=21 Win=65536 Len=0 TSval=2280361073 TSecr=2280361073
52	4.026272728	127.0.0.1	127.0.0.1	TCP	66	60672 - 8080 [ACK] Seq=7 Ack=220 Win=65408 Len=0 TSval=2280361073 TSecr=2280361073

No.	Time	Source	Destination	Protocol	Length	Info
1151	115.941655289	127.0.0.1	127.0.0.1	TCP	66 60782	- 8080 [ACK] Seq=7 Ack=22 Win=65536 Len=0 TSval=2280471982 TSecr=2280471982
1152	115.941705935	127.0.0.1	127.0.0.1	TCP	66 60782	- 8080 [ACK] Seq=7 Ack=221 Win=65408 Len=0 TSval=2280471982 TSecr=2280471982
1153	116.951886769	127.0.0.1	127.0.0.1	TCP	70 60782	- 8080 [PSH, ACK] Seq=7 Ack=221 Win=65536 Len=4 TSval=2280472992 TSecr=2280471982 [TCP segment of a connection]
1154	116.951920121	127.0.0.1	127.0.0.1	TCP	83 60782	- 8080 [PSH, ACK] Seq=11 Ack=221 Win=65536 Len=17 TSval=2280472992 TSecr=2280472992
1155	116.951939072	127.0.0.1	127.0.0.1	TCP	66 60782	- 8080 [FIN, ACK] Seq=22 Ack=221 Win=65536 Len=0 TSval=2280472992 TSecr=2280472992
1160	116.952023314	127.0.0.1	127.0.0.1	TCP	66 60782	- 8080 [ACK] Seq=29 Ack=222 Win=65536 Len=0 TSval=2280472992 TSecr=2280472992
1161	116.953499409	127.0.0.1	127.0.0.1	TCP	74 60784	- 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK Perm=1 TSval=2280472994 TSecr=0 WS=128
1163	116.953511142	127.0.0.1	127.0.0.1	TCP	66 60784	- 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2280472994 TSecr=2280472994
1164	116.953542293	127.0.0.1	127.0.0.1	TCP	70 60784	- 8080 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4 TSval=2280472994 TSecr=2280472994
1166	116.953551098	127.0.0.1	127.0.0.1	TCP	68 60784	- 8080 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=2 TSval=2280472994 TSecr=2280472994 [TCP segment of a connection]
1169	116.956057475	127.0.0.1	127.0.0.1	TCP	66 60784	- 8080 [ACK] Seq=7 Ack=14 Win=65536 Len=0 TSval=2280472996 TSecr=2280472996
1171	117.960283388	127.0.0.1	127.0.0.1	TCP	66 60784	- 8080 [ACK] Seq=7 Ack=22 Win=65536 Len=0 TSval=2280474001 TSecr=2280474001
1173	117.960328542	127.0.0.1	127.0.0.1	TCP	66 60784	- 8080 [ACK] Seq=7 Ack=221 Win=65408 Len=0 TSval=2280474001 TSecr=2280474001
1174	118.960995159	127.0.0.1	127.0.0.1	TCP	70 60784	- 8080 [PSH, ACK] Seq=7 Ack=221 Win=65536 Len=4 TSval=2280475010 TSecr=2280475010 [TCP segment of a connection]
1176	118.960999159	127.0.0.1	127.0.0.1	TCP	83 60784	- 8080 [PSH, ACK] Seq=11 Ack=221 Win=65536 Len=17 TSval=2280475010 TSecr=2280475010
1178	118.970069438	127.0.0.1	127.0.0.1	TCP	66 60784	- 8080 [ACK] Seq=29 Ack=222 Win=65536 Len=0 TSval=2280475010 TSecr=2280475010
1181	118.971589990	127.0.0.1	127.0.0.1	TCP	74 60786	- 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK Perm=1 TSval=2280475012 TSecr=0 WS=128
1183	118.971602795	127.0.0.1	127.0.0.1	TCP	66 60786	- 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2280475012 TSecr=2280475012
1184	118.971633660	127.0.0.1	127.0.0.1	TCP	70 60786	- 8080 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=4 TSval=2280475012 TSecr=2280475012
1186	118.971642714	127.0.0.1	127.0.0.1	TCP	68 60786	- 8080 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=2 TSval=2280475012 TSecr=2280475012 [TCP segment of a connection]
1188	118.971669137	127.0.0.1	127.0.0.1	TCP	66 60786	- 8080 [ACK] Seq=7 Ack=14 Win=65536 Len=0 TSval=2280475015 TSecr=2280475015
1191	119.976665277	127.0.0.1	127.0.0.1	TCP	66 60786	- 8080 [ACK] Seq=7 Ack=22 Win=65536 Len=0 TSval=2280476017 TSecr=2280476017
1193	119.976716627	127.0.0.1	127.0.0.1	TCP	66 60786	- 8080 [ACK] Seq=7 Ack=221 Win=65408 Len=0 TSval=2280476017 TSecr=2280476017
1194	120.986243691	127.0.0.1	127.0.0.1	TCP	70 60786	- 8080 [PSH, ACK] Seq=7 Ack=221 Win=65536 Len=4 TSval=2280477027 TSecr=2280477027 [TCP segment of a connection]
1196	120.986351888	127.0.0.1	127.0.0.1	TCP	83 60786	- 8080 [PSH, ACK] Seq=11 Ack=221 Win=65536 Len=17 TSval=2280477027 TSecr=2280477027
1198	120.986464056	127.0.0.1	127.0.0.1	TCP	66 60786	- 8080 [FIN, ACK] Seq=28 Ack=221 Win=65536 Len=0 TSval=2280477027 TSecr=2280477027
1200	120.986655236	127.0.0.1	127.0.0.1	TCP	66 60786	- 8080 [ACK] Seq=29 Ack=222 Win=65536 Len=0 TSval=2280477027 TSecr=2280477027

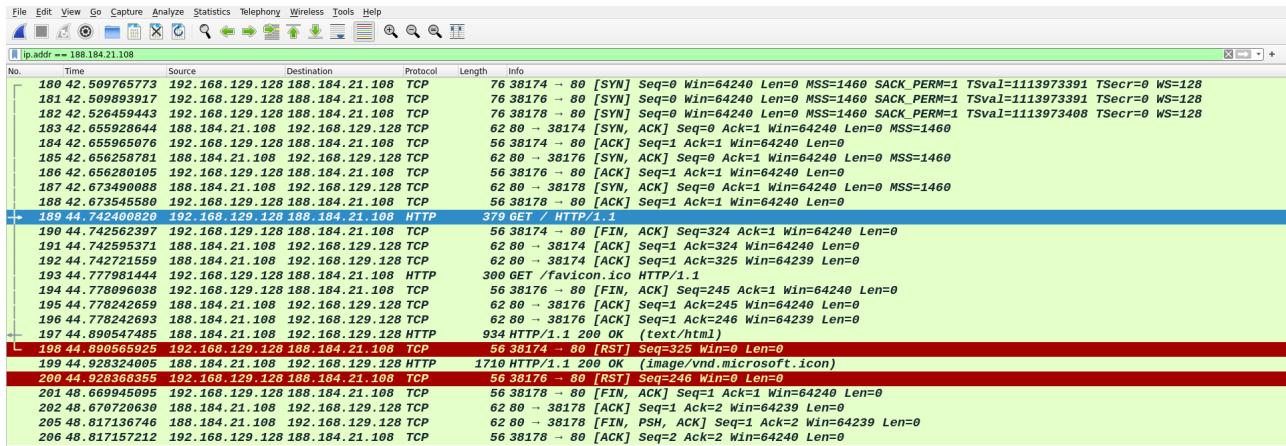
- The images shown above were captured at the beginning and the end of the 120 sec duration
- Two graphs were plotted using the data generated in the csv file(one with starting from origin and one not starting from origin, rather starting from 2).
- The points on the x axis were approximated to contain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

```
Frame 189: 379 bytes on wire (3032 bits), 379 bytes captured (3032 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.129.128, Dst: 188.184.21.108
Transmission Control Protocol, Src Port: 38174, Dst Port: 80, Seq: 1, Ack: 1, Len: 323
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      [GET / HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI:
    Request Version: HTTP/1.1
    Host: info.cern.ch\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
[FULL request URI: http://info.cern.ch/]
[HTTP request 1/1]
[Response in frame: 197]
```

- 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:

```

> Frame 197: 934 bytes on wire (7472 bits), 934 bytes captured (7472 bits) on interface any, id 0
> Linux cooked capture v1
> Internet Protocol Version 4, Src: 188.184.21.108, Dst: 192.168.129.128
> Transmission Control Protocol, Src Port: 80, Dst Port: 38174, Seq: 1, Ack: 325, Len: 878
> Hypertext Transfer Protocol
- HTTP/1.1 200 OK\r\n
  [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
  [HTTP/1.1 200 OK\r\n]
  [Severity level: chat]
  [Group: Sequence]
  Response Version: HTTP/1.1
  Status Code: 200
  [Status Code Description: OK]
  Response Phrase: OK
  Date: Tue, 05 Oct 2021 15:24:49 GMT\r\n
  Server: Apache\r\n
  Last-Modified: Wed, 05 Feb 2014 16:00:31 GMT\r\n
  ETag: "286-4f1aadb3105c0"\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 646\r\n
  [Content length: 646]
  Connection: close\r\n
  Content-Type: text/html\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.148146665 seconds]
  [Request in frame: 189]
  [Request URI: http://info.cern.ch/]
  File Data: 646 bytes
  Line-based text data: text/html (13 lines)

```

0040 20 32 30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 54 200 OK Date: T

- 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:

```
(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ifconfig -a
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 0x20<link>
            ether 00:0c:29:88:be:66  txqueuelen 1000  (Ethernet)
              RX packets 1711  bytes 1079571 (1.0 MiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 512  bytes 56355 (55.0 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 0x10<host>
            loop  txqueuelen 1000  (Local Loopback)
              RX packets 139  bytes 8959 (8.7 KiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 139  bytes 8959 (8.7 KiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

- 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:

```
(kali㉿kali)-[~/Desktop/CN/Assignments/Assignment2]
└─$ ifconfig eth0
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 0x20<link>
            ether 00:0c:29:88:be:66  txqueuelen 1000  (Ethernet)
              RX packets 8401  bytes 4693107 (4.4 MiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 5730  bytes 978678 (955.7 KiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

- 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:

My Public IPv4 is:
106.215.90.103 

My Public IPv6 is:
Not Detected

My IP Location: Noida, UP IN 

ISP: Bharti Airtel Ltd.

- 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 ifconfig 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, whereas 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 transmitted 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.)
```

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	0	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 (about 3.13.00 seconds)
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 web servers:

```
(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

Google
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: 0x0005, 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  46.035 ms  85.995 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:
 - $0.561 + 28.260 + 34.856 + 29.027 + 48.657 + 70.159 + 64.257 + 55.070 = 330.847 \text{ ms}$
- 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: 0x0005, 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: 0x0005, 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: 0x0005, 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: 0x0005, 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: 0x0005, 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: 0x0005, 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: 0x0005, 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 its 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 0x20<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 0x10<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 0x20<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
```