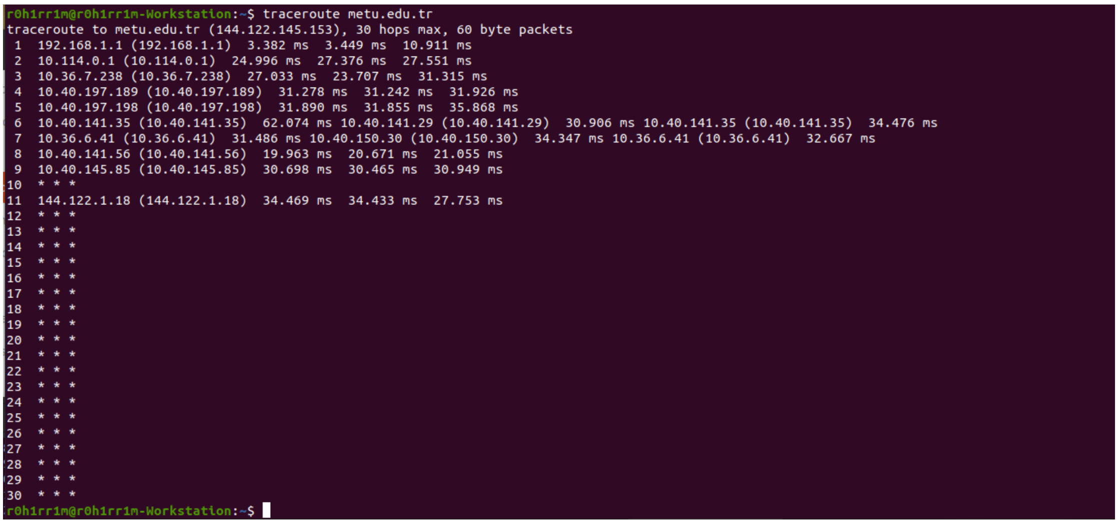
Student Information

Full Name: Furkan Göksel
ID Number: 2237436

1 Question 1

My traceroute output is as follows:



```
r0h1rr1m@r0h1rr1m-Workstation:~$ traceroute metu.edu.tr
traceroute to metu.edu.tr (144.122.145.153), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  3.382 ms  3.449 ms  10.911 ms
 2  10.114.0.1 (10.114.0.1)  24.996 ms  27.376 ms  27.551 ms
 3  10.36.7.238 (10.36.7.238)  27.033 ms  23.707 ms  31.315 ms
 4  10.40.197.189 (10.40.197.189)  31.278 ms  31.242 ms  31.926 ms
 5  10.40.197.198 (10.40.197.198)  31.890 ms  31.855 ms  35.868 ms
 6  10.40.141.35 (10.40.141.35)  62.074 ms 10.40.141.29 (10.40.141.29)  30.906 ms 10.40.141.35 (10.40.141.35)  34.476 ms
 7  10.36.6.41 (10.36.6.41)  31.486 ms 10.40.150.30 (10.40.150.30)  34.347 ms 10.36.6.41 (10.36.6.41)  32.667 ms
 8  10.40.141.56 (10.40.141.56)  19.963 ms  20.671 ms  21.055 ms
 9  10.40.145.85 (10.40.145.85)  30.698 ms  30.465 ms  30.949 ms
10  * * *
11  144.122.1.18 (144.122.1.18)  34.469 ms  34.433 ms  27.753 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
r0h1rr1m@r0h1rr1m-Workstation:~$
```

Figure 1: traceroute default output

As we can see from the output the IP address of metu.edu.tr is 144.122.145.153. However, in the traceroute output we couldn't see this address (in other words, the given output path doesn't contain the destination). Also in line 10, there are three asterisks that indicate no response. Therefore I couldn't see the whole path. The reason is the probes that are sent by traceroute may be ignored by these routers (line 10 may show this behavior) or blocked by firewall (after line 11 the reason may be this behavior, and maybe the responses can be blocked by firewall either).

2 Question 2

From the traceroute manual:

default
The traditional, ancient method of tracerouting. Used by default. Probe packets are udp datagrams with so-called "unlikely" destination ports. The "unlikely" port of the first probe is 33434, then for each next probe it is incremented by one. Since the ports are expected to be unused, the destination host normally returns "icmp unreach port" as a final response. (Nobody knows what happens when some application listens for such ports, though).

Also from our pcap (Figure 2), we can see that we use UDP datagrams as probes in the default method, and as it said in the question document, it sends three probes

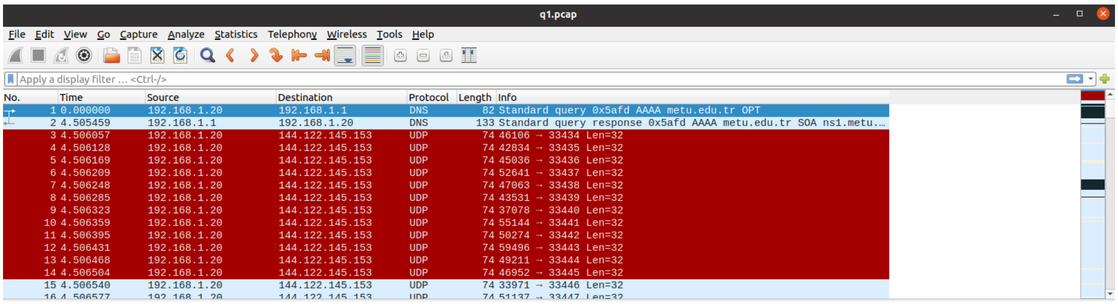


Figure 2: Default traceroute pcap

at each ttl. Moreover, we can see that for each packet it increases the destination port number (and it is started from 33434 as expected), and what I observe is the same as the explanation given in both manual and assignment text.

3 Question 3

Again from the manual,

-I: Use ICMP ECHO for probes

Also again from the manual,

In the modern network environment the traditional traceroute methods can not be always applicable, because of widespread use of firewalls. Such firewalls filter the "unlikely" UDP ports, or even ICMP echoes. To solve this, some additional tracerouting methods are implemented (including tcp), see LIST OF AVAILABLE METHODS below. Such methods try to use particular protocol and source/destination port, in order to bypass firewalls (to be seen by firewalls just as a start of allowed type of a network session).

3.1 Question 3.1

What have we changed using the -I flag?

Traceroute sends probes to detect route, and with flags we can change probe packet types. As we can see from above and from Figure 3, if we set this flag, traceroute starts sending ICMP packets instead of UDP datagrams as probes.

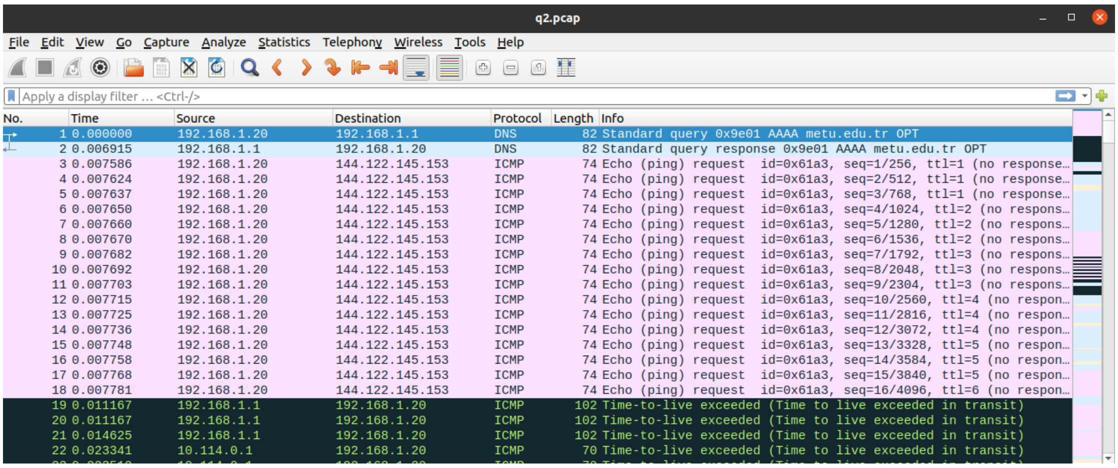


Figure 3: -I flag traceroute pcap

3.2 Question 3.2

Why would you get a different route trace/Wireshark capture using this flag?

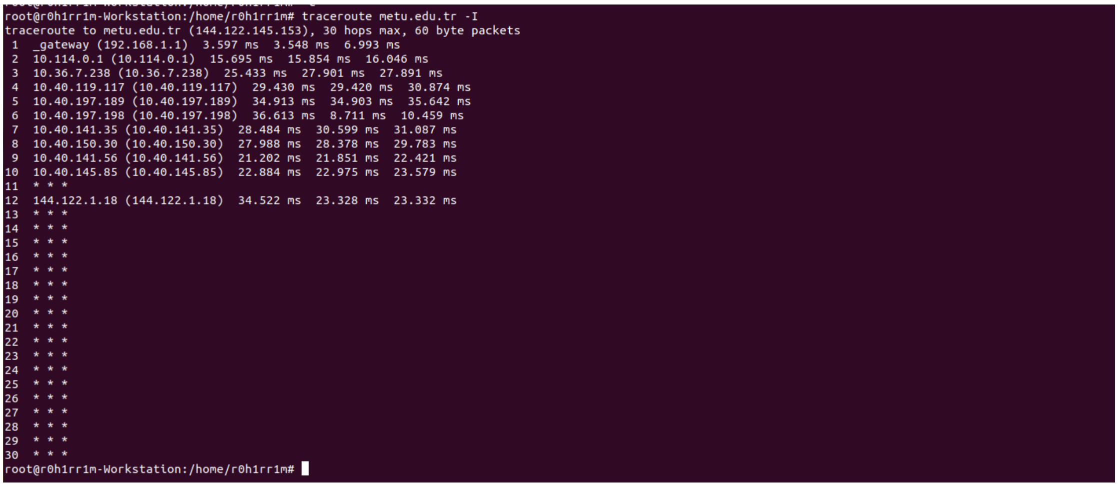


Figure 4: -I flag traceroute output

As stated in the manual, although we changed probe type, we still couldn't see behind 144.122.1.18. This means they don't allow ICMP traffic either (either ignore or block). However, there are different IP addresses in the path. The reason may be multiple paths of existence between routers. When we refer to the document Solarwinds The Shortfalls of Traceroute in Modern Multi-Path Networks whitepaper (`https://www.solarwindsmsp.com/sites/solarwindsmsp/files/resources/SW_MSP_Netpath_Traceroute_WhitePaper.pdf`), and from the manual (*If the probe answers come from different gateways, the address of each responding system will be printed.*), we can say that in the modern world, there exist different paths, and each packet is individual and because of that they can follow different routing paths (this may be the reason of different routes in these outputs). This is the intermediate router choice, and we can't control this.

4 Question 4

The university from Argentina that I chose is the National University of the South, its website is `uns.edu.ar` .

The university from Malaysia that I chose is Tunku Abdul Rahman University College, its website is `tarc.edu.my` .

4.1 Question 4.1

Write the websites/universities you have chosen alongside their IP addresses that you can reach using the traceroute command.
I checked their IP addresses that are found by traceroute with dig command, and outputs are as follow:

```
root@r0h1rr1m-Workstation:/home/r0h1rr1m# dig uns.edu.ar

; <<>> DiG 9.16.1-Ubuntu <<>> uns.edu.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 634
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;uns.edu.ar.                    IN      A

;; ANSWER SECTION:
uns.edu.ar.             7170    IN      A       200.49.224.150

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Cum Ara 25 23:31:50 +03 2020
;; MSG SIZE  rcvd: 55

root@r0h1rr1m-Workstation:/home/r0h1rr1m# traceroute uns.edu.ar
traceroute to uns.edu.ar (200.49.224.150), 30 hops max, 60 byte packets
```

Figure 5: Output for National University of the South

```
root@r0h1rr1m-Workstation:/home/r0h1rr1m# dig tarc.edu.my

; <<>> DiG 9.16.1-Ubuntu <<>> tarc.edu.my
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57020
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;tarc.edu.my.                   IN      A

;; ANSWER SECTION:
tarc.edu.my.            2162    IN      A       103.52.192.135

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Cum Ara 25 23:41:35 +03 2020
;; MSG SIZE  rcvd: 56

root@r0h1rr1m-Workstation:/home/r0h1rr1m# traceroute tarc.edu.my
traceroute to tarc.edu.my (103.52.192.135), 30 hops max, 60 byte packets
```

Figure 6: Output for Tunku Abdul Rahman University College

Therefore, uns.edu.ar is **200.49.224.150**, and tarc.edu.my is **103.52.192.135**

4.2 Question 4.2

If you have found a university website that you cannot reach using the traceroute commands given above, what's the "closest" you can get to the actual server using different traceroute options?

Actually I can reach National University of the South IP address using ICMP probes (one of the above commands), as it can be seen from Figure 7.

```
root@r0h1rr1m-Workstation:/home/r0h1rr1m# traceroute uns.edu.ar -I
traceroute to uns.edu.ar (200.49.224.150), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  10.757 ms  14.102 ms  14.090 ms
 2  10.114.0.1 (10.114.0.1)  22.860 ms  22.531 ms  22.474 ms
 3  10.36.7.238 (10.36.7.238)  22.277 ms  23.160 ms  23.537 ms
 4  10.40.195.201 (10.40.195.201)  23.138 ms  23.129 ms  23.120 ms
 5  10.40.197.189 (10.40.197.189)  23.109 ms  23.100 ms  23.092 ms
 6  10.40.197.198 (10.40.197.198)  23.829 ms  17.714 ms  9.576 ms
 7  10.40.141.29 (10.40.141.29)  27.522 ms  27.931 ms  28.537 ms
 8  10.36.6.41 (10.36.6.41)  26.588 ms  27.427 ms  27.414 ms
 9  10.38.219.1 (10.38.219.1)  18.627 ms  19.932 ms  20.404 ms
10  ix-ae-6-0.tcore1.it6-ankara.as6453.net (5.23.8.21)  65.755 ms  65.740 ms  65.726 ms
11  if-ae-37-3.tcore1.fr0-frankfurt.as6453.net (195.219.50.165)  64.992 ms  65.702 ms  65.647 ms
12  be3051.agr41.fra03.atlas.cogentco.com (130.117.15.85)  57.853 ms  57.657 ms  57.437 ms
13  be3187.ccr42.fra03.atlas.cogentco.com (130.117.1.118)  60.338 ms  59.771 ms  61.286 ms
14  be2800.ccr42.par01.atlas.cogentco.com (154.54.58.238)  69.600 ms  69.931 ms  69.129 ms
15  be2318.ccr32.bio02.atlas.cogentco.com (154.54.61.117)  91.686 ms  92.050 ms  92.040 ms
16  be2332.ccr42.dca01.atlas.cogentco.com (154.54.85.245)  153.687 ms  154.378 ms  154.370 ms
17  be2113.ccr42.atl01.atlas.cogentco.com (154.54.24.222)  166.328 ms  166.319 ms  166.310 ms
18  be3483.ccr22.mia01.atlas.cogentco.com (154.54.28.50)  183.302 ms  183.767 ms  183.758 ms
19  be3401.ccr21.mia03.atlas.cogentco.com (154.54.47.30)  179.782 ms  183.796 ms  183.784 ms
20  38.104.95.2 (38.104.95.2)  329.596 ms  330.255 ms  330.250 ms
21  186.0.210.10 (186.0.210.10)  330.245 ms  330.241 ms  330.276 ms
22  186.0.210.9 (186.0.210.9)  330.232 ms  330.228 ms  330.224 ms
23  200.49.239.130 (200.49.239.130)  330.219 ms  461.253 ms  464.674 ms
24  uns.edu.ar (200.49.224.150)  464.657 ms  464.648 ms  464.639 ms
```

Figure 7: Output of traceroute uns.edu.ar -I command

However, neither -I nor default option works on Tunku Abdul Rahman University College IP address. as it can be seen from Figure 8 and Figure 9

```
root@r0h1rr1m-Workstation:/home/r0h1rr1m# traceroute tarc.edu.my
traceroute to tarc.edu.my (103.52.192.135), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.062 ms  2.473 ms  3.179 ms
 2  10.114.0.1 (10.114.0.1)  13.783 ms  14.089 ms  14.066 ms
 3  10.36.7.238 (10.36.7.238)  14.589 ms  15.226 ms  15.204 ms
 4  10.40.197.189 (10.40.197.189)  15.180 ms  15.156 ms  15.133 ms
 5  10.40.197.194 (10.40.197.194)  15.124 ms  15.100 ms  15.077 ms
 6  10.40.141.33 (10.40.141.33)  33.989 ms  37.948 ms  47.032 ms
 7  10.36.95.166 (10.36.95.166)  38.262 ms  38.570 ms  40.383 ms
 8  10.36.6.38 (10.36.6.38)  41.266 ms  41.230 ms  41.194 ms
 9  100ge8-1.core1.mrs1.he.net (184.104.195.189)  95.365 ms  95.739 ms  95.703 ms
10  100ge14-2.core1.sin1.he.net (184.105.65.13)  255.712 ms  255.677 ms  255.639 ms
11  telekom-malaysia-inc.100gigabitethernet10-2.core1.sin1.he.net (74.82.46.50)  255.604 ms telekom-malaysia-inc.100gigabitethernet6-2.core1.sin1.he.net (65.49.109.190)  256.010 ms  255.422 ms
12  * * *
13  1.9.247.190 (1.9.247.190)  214.881 ms  217.176 ms  214.772 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Figure 8: Output of traceroute tarc.edu.my command

But when we run it with -T flag of traceroute, traceroute could reach the end server, and output is shown in Figure 10. In order to explain this option, I want to mention about -T parameter explanation in the manual:

Well-known modern method, intended to bypass firewalls.
Uses the constant destination port (default is 80, http). If some filters are present in the network path, then most probably any "unlikely" udp ports (as for default method) or even icmp echoes (as for icmp) are filtered, and whole tracerouting

```
root@r0h1rr1m-Workstation:/home/r0h1rr1m# traceroute tarc.edu.my -I
traceroute to tarc.edu.my (103.52.192.135), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.218 ms  1.427 ms  1.876 ms
 2  10.114.0.1 (10.114.0.1)  15.741 ms  15.733 ms  15.974 ms
 3  10.36.7.238 (10.36.7.238)  9.663 ms  11.084 ms  11.077 ms
 4  10.40.197.141 (10.40.197.141)  11.065 ms  11.058 ms  11.050 ms
 5  10.40.197.189 (10.40.197.189)  11.040 ms  11.033 ms  11.026 ms
 6  10.40.197.194 (10.40.197.194)  12.989 ms  8.209 ms *
 7  10.40.141.33 (10.40.141.33)  63.288 ms  63.278 ms  63.269 ms
 8  10.36.95.166 (10.36.95.166)  59.677 ms  59.668 ms  59.661 ms
 9  10.36.6.38 (10.36.6.38)  59.648 ms  59.640 ms  59.631 ms
10  100ge8-1.core1.mrs1.he.net (184.104.195.189)  84.070 ms  114.475 ms  114.467 ms
11  100ge14-2.core1.sin1.he.net (184.105.65.13)  216.721 ms  238.376 ms  238.335 ms
12  telekom-malaysia-inc.100gigabitethernet6-2.core1.sin1.he.net (65.49.109.190)  226.112 ms  220.504 ms  220.459 ms
13  * * *
14  1.9.247.190 (1.9.247.190)  251.784 ms  309.072 ms  309.018 ms
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Figure 9: Output of traceroute tarc.edu.my -I command

```
root@r0h1rr1m-Workstation:/home/r0h1rr1m# traceroute tarc.edu.my -T
traceroute to tarc.edu.my (103.52.192.135), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.937 ms  3.098 ms  3.334 ms
 2  10.114.0.1 (10.114.0.1)  12.179 ms  13.786 ms  13.977 ms
 3  10.36.7.238 (10.36.7.238)  30.606 ms  30.599 ms  30.592 ms
 4  10.40.197.189 (10.40.197.189)  30.584 ms  30.601 ms  30.594 ms
 5  10.40.197.194 (10.40.197.194)  30.585 ms  47.451 ms  30.572 ms
 6  10.40.141.33 (10.40.141.33)  58.132 ms  66.511 ms *
 7  10.36.95.166 (10.36.95.166)  178.167 ms  184.128 ms  184.098 ms
 8  10.36.6.38 (10.36.6.38)  186.431 ms  186.399 ms  187.332 ms
 9  100ge8-1.core1.mrs1.he.net (184.104.195.189)  225.577 ms  225.651 ms  225.621 ms
10  100ge14-2.core1.sin1.he.net (184.105.65.13)  358.123 ms  358.090 ms  358.752 ms
11  telekom-malaysia-inc.100gigabitethernet6-2.core1.sin1.he.net (65.49.109.190)  356.405 ms telekom-malaysia-inc.100gigabitethernet10-2.core1.sin1.he.net (74.82.46.50)  358.688 ms  358.573 ms
12  * * *
13  1.9.247.190 (1.9.247.190)  625.658 ms  628.483 ms  625.742 ms
14  * * *
15  * * *
16  * * *
17  * * *
18  103.52.192.135 (103.52.192.135)  532.593 ms  533.282 ms  532.106 ms
```

Figure 10: Output of traceroute tarc.edu.my -T command

will just stop at such a firewall. To bypass a network filter, we have to use only allowed protocol/port combinations. If we trace for some, say, mailserver, then more likely -T -p 25 can reach it, even when -I can not.

So what is this explanation? First of all, I saw that both our default UDP and ICMP probes methods don't work, and the traceroute couldn't reach the web server. However, I knew that I could reach the website using my browser, and my browser had to use the same internal routers (routers in the tarc.edu.my's subnet) to reach tarc.edu.my webserver. Therefore I assumed that both my ICMP and UDP probes are dropped by a firewall in the Tunku Abdul Rahman University College network, but the same firewall knows that there is a web server in the network, so it sees TCP communication to port 80 is legitimate traffic, and it knows that it shouldn't drop these packets.

In order to support this assumption, I checked some vulnerabilities that may be the reason for such blocking. Because if they are blocked by a firewall, there should be a reason, otherwise my assumption would be wrong.

So, firstly why our UDP probes are blocked? (Refer to `https://www.netscout.com/what-is-ddos/udp-flood`) In UDP Flood DDoS Attack, an attacker sends UDP packets to random ports, and if the destination port is closed, the host issues a "Destination Unreachable" packet back to the sender. If an attacker bombards UDP packets like that, the system would be unresponsive to legitimate traffic. In order to avoid this kind of attack, one can set his/her firewall such that it blocks all traffic that goes to closed ports. Since our default UDP mode of traceroute says "Probe packets are udp datagrams with so-called "unlikely" destination ports. The "unlikely" port of the first probe is 33434, then for each next probe it is incremented by one.", our probes will be blocked also because they use the same approach in this attack type even if the intention is not attacking the webserver.

When it comes to ICMP packets, (Refer to `https://www.netscout.com/what-is-ddos/icmp-flood`) in ICMP Flood DDoS Attack similar to UDP Flood DDoS Attack, attacker attempts to overwhelm a targeted device with ICMP echo-requests (pings). Actually, these ICMP packets are used to check the connectivity or health of network devices, and they are legitimate. When a host receives an ICMP packet, it responds to this packet to say "I'm okay, I can receive and reply to your messages". But again, by flooding the target with request packets, the host device is forced to respond with an equal number of reply packets. As a result of this, the target becomes inaccessible to other traffic. In order to eliminate this chance, one can drop ICMP packets in the firewall. Since our -I flag (ICMP method) says "Most usual method for now, which uses icmp echo packets for probes.", the traceroute's packets will be also dropped.

After learning these two vulnerabilities, as I said before, I thought that traceroute should send an allowed packet type as probes to reach the final destination, and since I can reach the website over my browser, and my browser uses HTTP protocol which relies on TCP packets whose destination is port 80, if traceroute somehow is able to use TCP packets to port 80 for its execution logic, its packets can also pass from the firewall. Thankfully, traceroute has a built-in method that uses TCP (manual explanation is written above). I used this mod and traceroute was able to reach the final destination.

5 Question 5

According to iana.org (refer to `https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml`), in the Internet Protocol version 4 (IPv4), field called "Protocol" identifies the next level protocol. As it can be seen from the Figure 11, it is 1, and it means ICMP.

6 Question 6

First of all, according to manual of traceroute, when we say traceroute uns.edu.ar -I 92, we give a packet length parameter to traceroute (which is 92), and this value is used to set total size of the probing packet. Also I am referring to RFC 791 (`https://tools.ietf.org/html/rfc791`) while answering the question.

Total Length is the length of the datagram, measured in octets, including internet header and data. So it is 92 as expected, and it can be seen in Figure 11. The header length field indicates the length of the IP header and it is 20 (again from

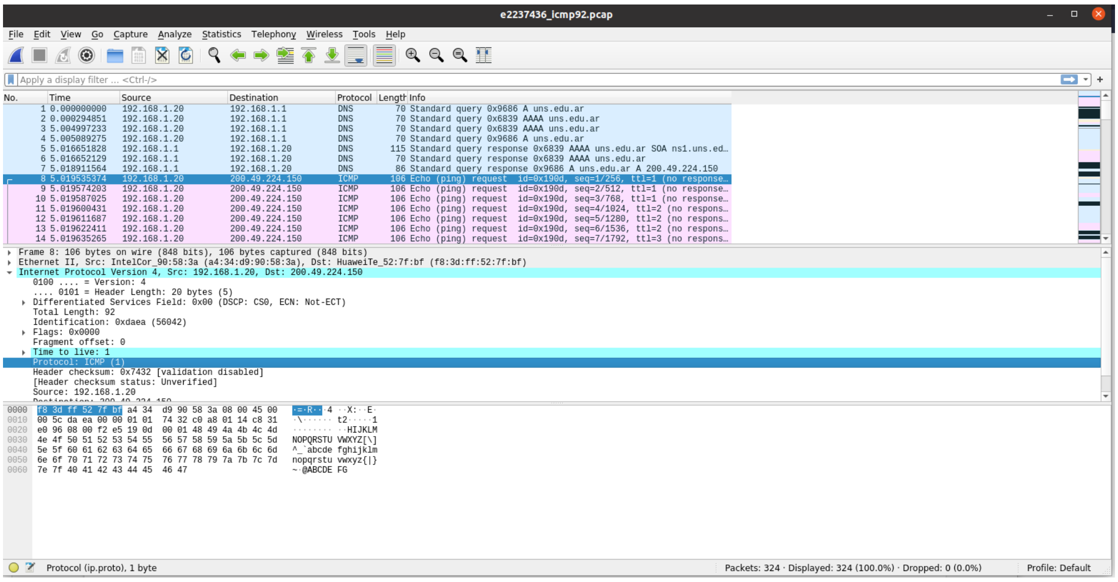


Figure 11: Screenshot of protocol field

Figure 11). 20 is expected and it is normal because RFC says The maximal internet header is 60 octets, and a typical internet header is 20 octets. So 92-20 will give the payload length (according to RFC) and it is 72.

7 Question 7

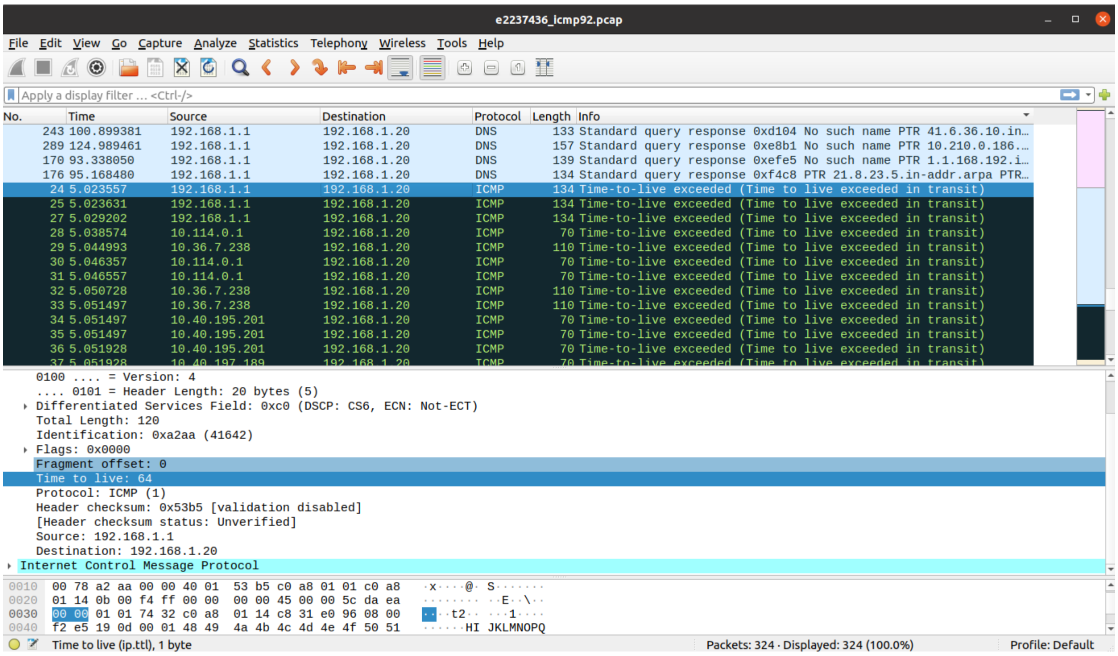


Figure 12: Screenshot of ICMP exceeded packet

I sorted the packets using the info column in order to clusterize packets. In Figure 12, it can be seen that the packet no of the chosen packet is 24. Its TTL value is

64, and its identification value is 41642. Identification value is used in the fragmentation, and RFC says "The identification field is used to distinguish the fragments of one datagram from those of another.". Therefore, it is unique and changes for each ICMP exceeded packets. When it comes to TTL, TTL is unchanged for the same source (eg. If packets come from 192.168.1.1, for all three packets, TTL is the same). But, it differs when the source address is changed. It is set by different routers to different values. Since its purpose is just determining the lifetime of a datagram, it can be either the same or different for other packets because its aim is not identification.

8 Question 8

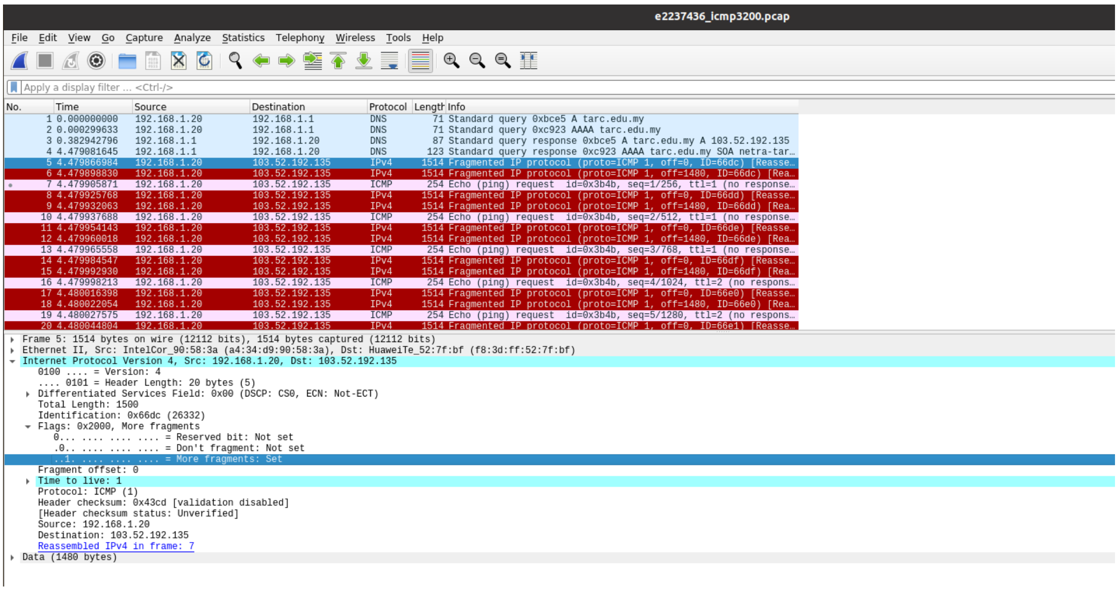


Figure 13: Screenshot of first fragment of first ICMP Echo Request

```
Flags:  3 bits

  Various Control Flags.

    Bit 0: reserved, must be zero
    Bit 1: (DF) 0 = May Fragment,  1 = Don't Fragment.
    Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.

        0   1   2
      +---+---+---+
      |   | D | M |
      | 0 | F | F |
      +---+---+---+
```

Figure 14: Screenshot from RFC

Screenshot of the first fragment (we know it is the first fragment because fragment offset value is 0, offset value and the identification value are used to reassemble the packet) can be seen in Figure 13. In the fragmentation section of RFC, it is written that The More Fragments flag bit (MF) is set if the datagram is not the last fragment. Also, in Figure 14 explains the flags field. Based on this information, more fragments bit is set, and this indicates packet is fragmented.

9 Question 9

From RFC it is written that:

> The receiver of the fragments uses the identification field to ensure that fragments of different datagrams are not mixed. The fragment offset field tells the receiver the position of a fragment in the original datagram. The fragment offset and length determine the portion of the original datagram covered by this fragment. The more-fragments flag indicates (by being reset) the last fragment. These fields provide sufficient information to reassemble datagrams.

So, based on this explanation, there is no indication of the number of fragments in any packets. While reassembling the packets according to fragment offset and identification values, the stop condition is 0 value in more fragments flag. The approach is like a loop that runs until 0 value in the flag. Therefore, we cannot know how many fragments have been created by the fragmentation from the first datagram.

In order to determine how many fragments have been created, we will look at the identification value (26332) and more-fragments flag. At packet no 7 (which is shown in Figure 15, we see that the identification value is the same, and its more fragments flag is 0. This will tell us that packet no 7 is the last packet in this fragmentation. Therefore, we can say that there are three fragments in this fragmentation.

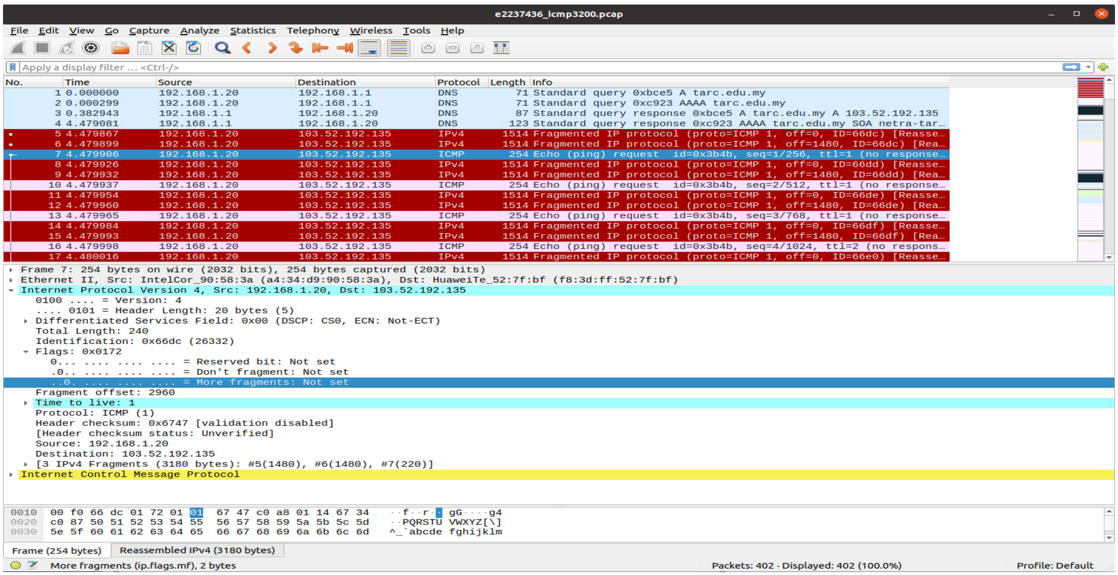


Figure 15: The last fragmented packet of fragmentations

10 Question 10

Since the packets are fragmented, first of all their **fragment offset** values will be different for reassembling process. Also, **total length** and **flags** will be different (eg. last fragment' more fragments flag). Lastly, **checksum** will be different.