COL-334 Computer Networks
Assignment-1

Chinmay Mittal (2020CS10336)

Networking Tools

1. **IP Address:**
    a. The IP address of the machine can change as we change the internet service provider. An IP address is assigned by the ISP from the set of IP addresses it owns. When we shift to a new ISP, the IP address assigned to us changes.
    b. Command used to check the IP address on Windows machine *ipconfig /all* (command for private IP address)

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : iitd.ac.in
   Description . . . . . . . . . . . : Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC
   Physical Address. . . . . . . . . : 90-0F-0C-31-2D-75
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::9dc6:8168:7034:28f1%22(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.194.18.78(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.224.0
   Lease Obtained. . . . . . . . . . : 24 August 2022 17:33:58
   Lease Expires . . . . . . . . . . : 24 August 2022 18:03:46
   Default Gateway . . . . . . . . . : 10.194.0.1
   DHCP Server . . . . . . . . . . . : 10.7.10.2
   DHCPv6 IAID . . . . . . . . . . . : 546311948
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-2B-E3-AF-90-0F-0C-31-2D-75
   DNS Servers . . . . . . . . . . . : 10.10.2.2
                                       10.10.1.2
   Primary WINS Server . . . . . . . : 10.8.2.3
   Secondary WINS Server . . . . . . : 10.8.2.8
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       iitd.ac.in
                                       cc.iitd.ac.in
```

IP address of my machine when connected to IITD WiFi

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC
   Physical Address. . . . . . . . . : 90-0F-0C-31-2D-75
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2405:204:3485:d439:9dc6:8168:7034:28f1(Preferred)
   Temporary IPv6 Address. . . . . . : 2405:204:3485:d439:7853:bb6a:e1c9:24b5(Preferred)
   Link-local IPv6 Address . . . . . : fe80::9dc6:8168:7034:28f1%22(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.43.161(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 24 August 2022 18:00:00
   Lease Expires . . . . . . . . . . : 24 August 2022 19:00:00
   Default Gateway . . . . . . . . . : fe80::30ab:6aff:fe7b:1f65%22
                                       192.168.43.1
   DHCP Server . . . . . . . . . . . : 192.168.43.1
   DHCPv6 IAID . . . . . . . . . . . : 546311948
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-2B-E3-AF-90-0F-0C-31-2D-75
   DNS Servers . . . . . . . . . . . : 192.168.43.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

IP address of my machine on Jio network

To check the public IP address I used Google chrome. The results are as follows:

2. **<u>Nslookup</u>**
    a. Changing the DNS server can change the returned IP address of the host since different DNS servers can have different cached versions of the IP address of a given host.

```
dress:  10.10.2.2

ww.google.com
rver:  dns1.cc.iitd.ac.in
dress:  10.10.2.2

n-authoritative answer:
me:    www.google.com
dresses:  2404:6800:4002:807::2004
          142.251.42.4

ww.facebook.com
rver:  dns1.cc.iitd.ac.in
dress:  10.10.2.2

n-authoritative answer:
me:    star-mini.c10r.facebook.com
dresses:  2a03:2880:f12f:83:face:b00c:0:25de
          157.240.16.35
iases:  www.facebook.com
```

```
Address:  8.8.8.8

> www.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4002:810::2004
          172.217.167.196

> www.facebook.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    star-mini.c10r.facebook.com
Addresses:  2a03:2880:f144:181:face:b00c:0:25de
          157.240.239.35
Aliases:  www.facebook.com
```

```
Server:  UnKnown
Address:  192.168.43.1

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4002:82c::2004
          142.250.206.132

> www.facebook.com
Server:  UnKnown
Address:  192.168.43.1

Non-authoritative answer:
Name:    star-mini.c10r.facebook.com
Addresses:  2a03:2880:f144:82:face:b00
          157.240.198.35
Aliases:  www.facebook.com
```

Nslookup program run with different DNS servers

3. **PING**
    a. To alter the TTL values (Time To Live), we use the -i command. Reducing the TTL causes the TTL to expire in transit. For every router the packet visits the router reduces the TTL by one and when it becomes zero the packet is dropped and TTL expired in transit (without reaching destination) message is sent back
    b. To alter the size of the packets sent, we use the -l command (this increases the time taken by the packet)

```
C:\Users\Chinmay Mittal>ping 142.251.42.4

Pinging 142.251.42.4 with 32 bytes of data:
Reply from 142.251.42.4: bytes=32 time=45ms TTL=116
Reply from 142.251.42.4: bytes=32 time=27ms TTL=116
Reply from 142.251.42.4: bytes=32 time=40ms TTL=116
Reply from 142.251.42.4: bytes=32 time=35ms TTL=116

Ping statistics for 142.251.42.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 27ms, Maximum = 45ms, Average = 36ms
```

```
C:\Users\Chinmay Mittal>ping 142.251.42.4 -l 64

Pinging 142.251.42.4 with 64 bytes of data:
Reply from 142.251.42.4: bytes=64 time=703ms TTL=116
Reply from 142.251.42.4: bytes=64 time=168ms TTL=116
Reply from 142.251.42.4: bytes=64 time=207ms TTL=116
Reply from 142.251.42.4: bytes=64 time=548ms TTL=116

Ping statistics for 142.251.42.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds:
    Minimum = 168ms, Maximum = 703ms, Average = 406ms
```

```
C:\Users\Chinmay Mittal>ping google.com -i 1

Pinging google.com [142.250.193.238] with 32 bytes of data:
Reply from 10.194.0.14: TTL expired in transit.
Reply from 10.194.0.14: TTL expired in transit.
Reply from 10.194.0.14: TTL expired in transit.
Reply from 10.194.0.14: TTL expired in transit.

Ping statistics for 142.250.193.238:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

4. **Traceroute**
    a. To force traceroute to use IPv4, we can use the -4 flag
    b. To force missing routers to respond, we can increase the timeout using the -w command and wait longer for their response.
    c. IP addresses in the range (10.0.0.0 - 10.255.255.255, 192.168.0.0 - 192.168.255.255 etc. are private IP addresses), and some of these appear in the traceroute

```
C:\Users\Chinmay Mittal>tracert google.com

Tracing route to google.com [142.250.193.206]
over a maximum of 30 hops:

  1    2 ms     2 ms     2 ms  10.184.0.14
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     5 ms     6 ms     5 ms  del11s17-in-f14.1e100.net [142.250.193.206]

Trace complete.
```

Traceroute over IITD WiFi to google.com

```
C:\Users\Chinmay Mittal>tracert www.google.com

Tracing route to www.google.com [2404:6800:4002:82b::2004]
over a maximum of 30 hops:

  1    72 ms    10 ms     8 ms  2405:204:3485:d439::d1
  2     *        *        *     Request timed out.
  3    56 ms    68 ms    57 ms  2405:200:331:1503::ff02
  4    52 ms    39 ms    38 ms  2405:200:801:300::dcc
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7   223 ms    26 ms    60 ms  2001:4860:1:1::1ea2
  8    63 ms    31 ms    46 ms  2404:6800:8121::1
  9    47 ms    38 ms    53 ms  2001:4860:0:1::54fa
 10    67 ms    39 ms    40 ms  2001:4860:0:1::5e5b
 11    51 ms    39 ms    37 ms  del11s20-in-x04.1e100.net [2404:6800:4002:82b::2004]

Trace complete.
```

```
C:\Users\Chinmay Mittal>tracert -4 www.google.com

Tracing route to www.google.com [142.250.192.196]
over a maximum of 30 hops:

  1     6 ms     5 ms     5 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3    50 ms   317 ms    57 ms  10.71.83.50
  4     *      177 ms    22 ms  172.26.100.118
  5    32 ms    49 ms    64 ms  172.26.100.102
  6   144 ms     *       28 ms  192.168.44.24
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10    53 ms    37 ms     *     72.14.195.22
 11    65 ms    54 ms     *     209.85.244.73
 12   283 ms   101 ms   215 ms  142.250.236.55
 13    47 ms    56 ms    40 ms  del11s12-in-f4.1e100.net [142.250.192.196]

Trace complete.
```

Traceroute to google over jio network with IPv6 and IPv4

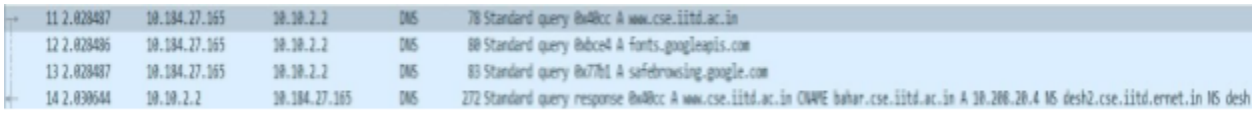
```
C:\Users\Chinmay Mittal>tracert -w 10000 www.iitd.ac.in

Tracing route to www.iitd.ac.in [10.10.211.212]
over a maximum of 30 hops:

  1  3680 ms  4833 ms  4668 ms  10.184.0.14
  2  4876 ms  4993 ms  3843 ms  10.254.236.18
  3  4499 ms  3787 ms  4109 ms  www.iitd.ac.in [10.10.211.212]

Trace complete.
```

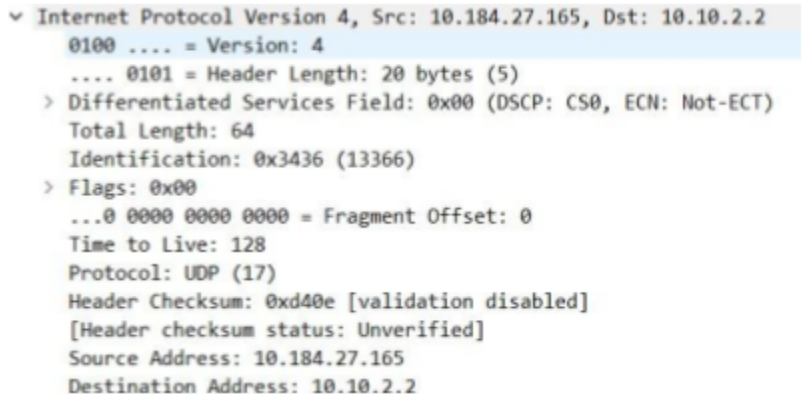
Forcing missing routers to respond using the -w timeout command and increasing the timeout to wait for the response

```
C:\Users\Chinmay Mittal>tracert www.iitd.ac.in

Tracing route to www.iitd.ac.in [10.10.211.212]
over a maximum of 30 hops:

  1     5 ms     3 ms     5 ms  10.184.0.14
  2     *        *        *     Request timed out.
  3  3877 ms     *        *     www.iitd.ac.in [10.10.211.212]
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *     3542 ms  www.iitd.ac.in [10.10.211.212]

Trace complete.
```

DNS Task

1. DNS query and response are sent over UDP.





DNS response and query and Protocol used for this message is UDP

2. One query is sent from my browser to the DNS server.
3. One DNS server is involved, which responds with the IP address of the hostname.
4. DNS server 10.10.2.2 i.e dns1.cc.iitd.ernet.in is involved, which returns the response, i.e the IP address of the hostname www.cse.iitd.ac.in.
5. The DNS server queried is the one that responds since it is authoritative for the hostname it doesn't query any other DNS server.
6. Two resource records are involved in the answer for resolving the IP of the host. One RR was involved in getting the canonical host for the alias. The second RR was involved in getting the IP address of the canonical host.

Name	Value	Type	TTL
www.cse.iitd.ac.in	bahar.cse.iitd.ac.in	CNAME	3600 (1 hr)
bahar.cse.iitd.ac.in	10.208.20.4	A	3600 (1 hr)

One query was sent to the DNS server

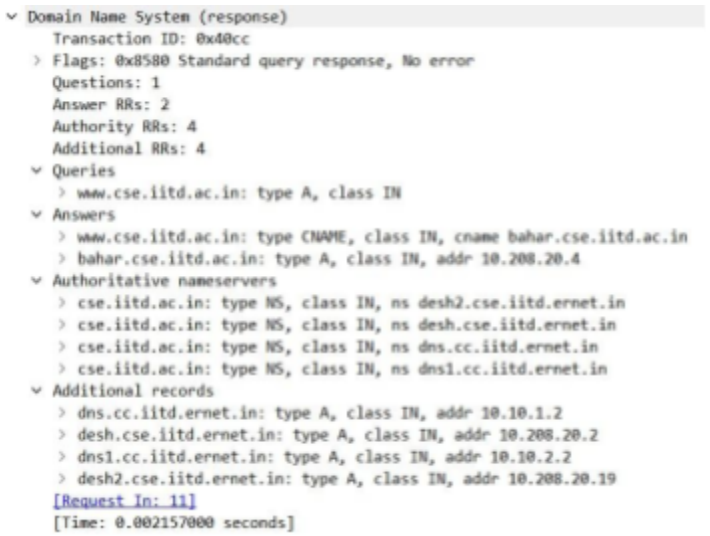
Name	Type
www.cse.iitd.ac.in	A

The DNS server also replies with some additional authoritative name server in the DNS response though they are not directly used in resolving the host's IP address.

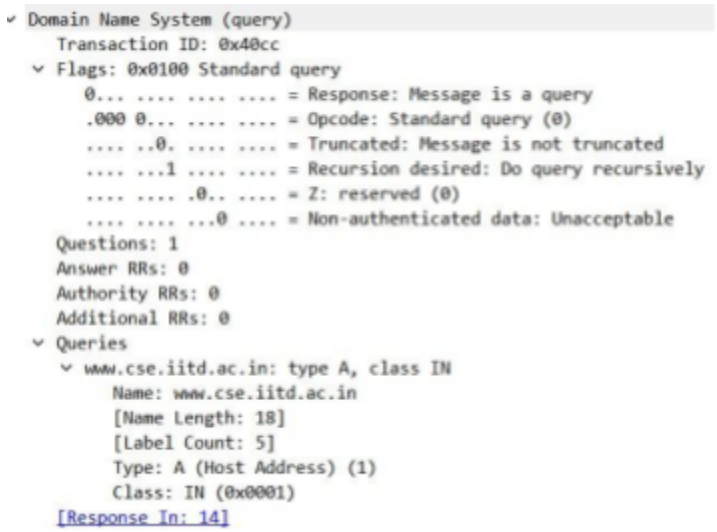
Name	Value	Type	TTL
cse.iitd.ac.in	desh2.cse.iitd.ernet.in	NS	3600 (1hr)
cse.iitd.ac.in	desh.cse.iitd.ernent.in	NS	3600 (1hr)
cse.iitd.ac.in	dns.cc.iitd.ernet.in	NS	3600 (1hr)
cse.iitd.ac.in	dns1.cc.iitd.ernent.in	NS	3600 (1hr)

The DNS server also replies with the IP addresses of these authoritative name servers in the Additional records section of the DNS response. These were again not used in resolving the IP address of the host.

Name	Value	Type	TTL
desh2.cse.iitd.ernet.in	10.208.20.19	A	3600 (1hr)
desh.cse.iitd.ernent.in	10.208.	A	3600 (1hr)
dns.cc.iitd.ernet.in	10.10.1.2	A	3600 (1hr)
dns1.cc.iitd.ernent.in	10.10.2.2	A	3600 (1hr)



DNS response message

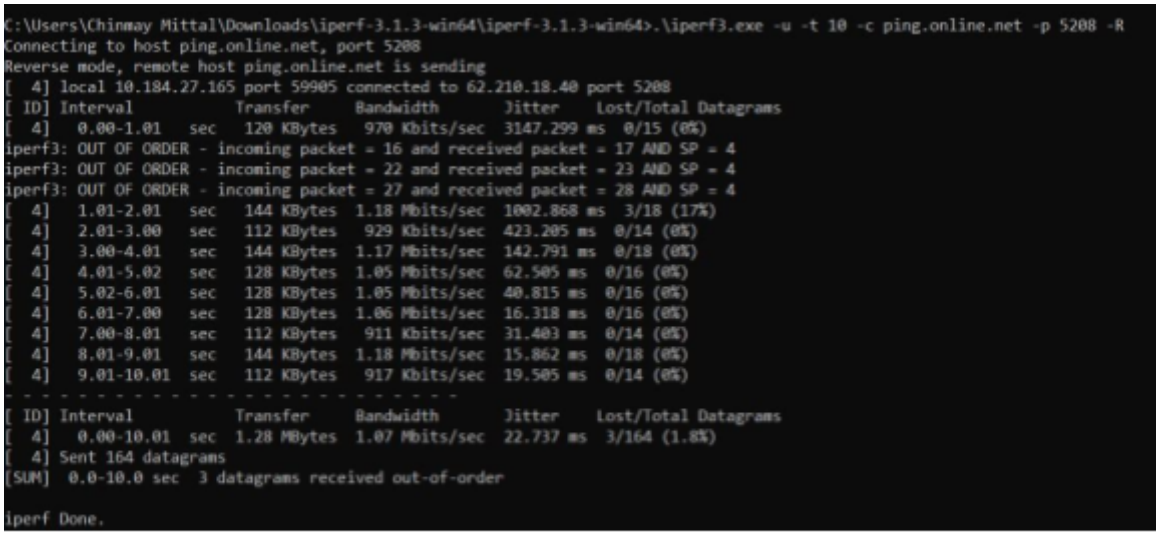


DNS query message

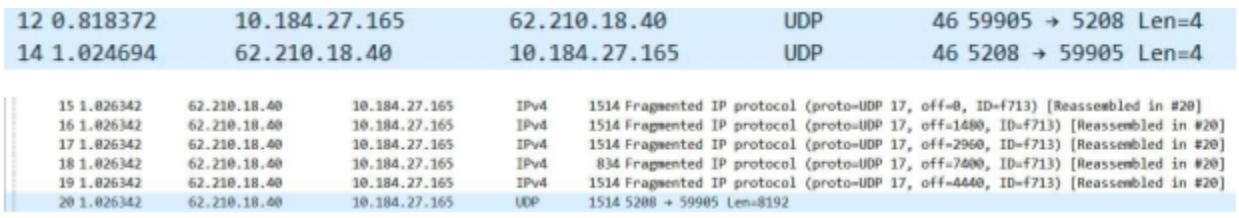
iPerf Task

1.
   - 166 packets are sent over UDP in the exchange (found by filter UDP).
   - There are 2 initial UDP packets which are not fragmented (small size) and 164 packets of large size which undergo IPv4 fragmentation and each of these 164 UDP packets are fragmented into 6 IPv4 fragments. Thus the total IP packets used to send these UDP message are 6*164 + 2 = 986.
2.
   - Since the iperf command is run in reverse mode, the server is sending the client data in bulk.
   - Most UDP packets have 8192 bytes of data and 8 bytes of headers, except 2 packets which have 4 bytes of data and 8 bytes of headers. Hence the average size of packets sent is close to 8200 bytes (including UDP headers and not considering fragments separately).
   - If each IPv4 fragment is considered separately as a packet then the average packet size is 1398 (this considers the 14 byte ethernet header and 20 bytes IPv4 header included in each IP packet), this value is found from the wireshark file properties window.
3.
   - Length in UDP's length field is 8200. There are 164 packets of this kind (ignoring the 2 initial UDP packets with length 12).
   - This means that the total data sent during the session is 8200*164, which is 1344800 bytes or 1.2825 Mbytes (this is close to the iperf terminal).
   - To find the time of transfer we subtract the time to receive the final packet from the time to receive the initial UDP packet (11.166 - 1.026 = 10.14 s).

- The throughput is thus 0.126 MBytes/s (1.2825 Mbytes / 10.14 s) or 1.01 Mbits/sec, which is close to the value reported by iperf.
- The rate displayed by the capture filter window is different in Wireshark because of IP fragmentation. Because of the large size of the UDP packets, the Internet Protocol fragments the UDP packets into smaller chunks. Only one of these fragments are captured by the UDP filter applied in Wireshark (and their size includes the IPv4 and ethernet headers). Because of this, the throughput displayed by Wireshark (using the udp filter and file properties window) is wrong.
- If we include all the IP fragments (I used the not tcp filter) the file properties window in wireshark returns a throughput of 1.06 Mbits / sec (this is slightly greater since it includes the data of the IPv4 and ethernet headers also). Also the average packet size is different since it is actually the average fragment size over which the UDP payload is divided and also includes other headers.



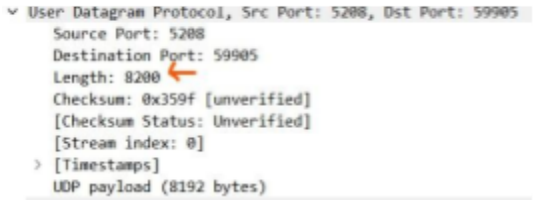
Iperf task statistics displayed by the iperf3 program on the terminal



Initial not fragmented UDP packets (top) and fragmented large UDP packets (bottom)

Displayed
986 (97.1%)
10.349
95.3
1398
1378348 (99.8%)
133 k
1065 k

Wireshark statistics of all UDP packets (including all IP fragments), 2 non fragmented packets + 984 fragments (6 fragments for each of the 164 packets)



Length field in UDP packet used for throughput calculations

Displayed
166 (16.4%)
10.349
16.0
1496
248388 (18.0%)
24 k
192 k

Filter statistics for the UDP filter in wireshark for each fragmented UDP packet only one fragment is displayed and hence the calculations don't match

HTTP Task

1. There is one HTTP request packet and one HTTP response packet with status code 101 for switching protocols. The second packet is both an HTTP 2 and an HTTP 1.1 packet. After these two packets, 8 HTTP2 packets are exchanged. So total 2 packet are HTTP 1.1 and 9 packets are 9
2. The response to the original GET request in the HTTP packet is obtained in the 6th HTTP2 packet, and before that, 4 HTTP2 packets are exchanged.
3.
    a. HTTP1 headers have information about the request type, such as GET / PUT / POST etc. It also contains information about the host/server and caching directives, the resource to be fetched, and the server responding to the request.
    b. HTTP2 allows response and request multiplexing. HTTP2 does not change the semantics of HTTP1 but modifies how the data is formatted and converted into frames which are sent over streams. HTTP2 headers also contain information about the stream and the frame.

PING Task

Note: I changed the remote server to ping.online.net since ping-ams1.online.net was not working.

1. The total number of IP packets exchanged was 10 (5 requests and 5 replies). Although each packet was fragmented into 3 packets before sending, counting the total IP fragmented packets this way gives 30.
2. Each ping packet had a payload of 3500 bytes and 8 bytes of ICMP header. This data of 3508 was fragmented into 3 packets. Two packets of size 1514 (1480 payload, 20 bytes IPv4 header, 14 bytes ethernet header) and one packet of size 582 (548 payload, 20 bytes IPv4 header, 14 bytes ethernet header). The total payload for all these packets matches 3508, i.e (3500 bytes data + 8 bytes header) of the ping packet.
3.

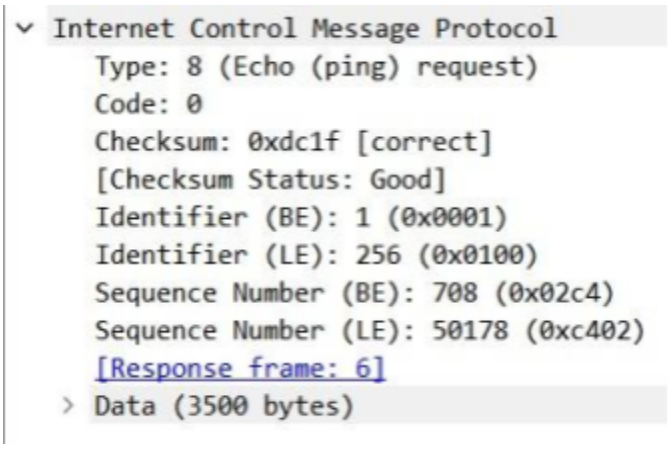
```
C:\Users\Chinmay Mittal>ping -n 5 -l 3500 ping.online.net

Pinging ping.online.net [62.210.18.40] with 3500 bytes of data:
Reply from 62.210.18.40: bytes=3500 time=157ms TTL=47
Reply from 62.210.18.40: bytes=3500 time=147ms TTL=47
Reply from 62.210.18.40: bytes=3500 time=162ms TTL=47
Reply from 62.210.18.40: bytes=3500 time=147ms TTL=47
Reply from 62.210.18.40: bytes=3500 time=158ms TTL=47

Ping statistics for 62.210.18.40:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 147ms, Maximum = 162ms, Average = 154ms
```

Ping Task: sending and receiving 5 ping packets to ping.online.net i.e 62.210.18.40

Packet No.	Size of Data	Time taken to receive response	TTL	fragmented
1	3500 bytes + 8 bytes header	157 ms	447	true
2	3500 bytes + 8 bytes header	147 ms	447	true
3	3500 bytes + 8 bytes header	162 ms	447	true
4	3500 bytes + 8 bytes header	147 ms	447	true
5	3500 bytes + 8 bytes header	158 ms	447	true

All the request packets were fragmented, and this was observed in the Flags header of the IPv4 protocol over which these packets were sent.

IPv4 header indicating that the ping requests are fragmented



ICMP header in ping packet indicating it is a request

Each of the ping requests and responses are fragmented into 3 IP packets the data in them is as follows. The total payload for each ping packet is 3508 (3500 bytes data + 8 bytes header)

	Fragment size	Ethernet Header	IPv4 header	Payload
Fragment No. 1	1514 bytes	14 bytes	20 bytes	1480 bytes
Fragment No. 2	1514 bytes	14 bytes	20 bytes	1480 bytes
Fragment No. 3	582 bytes	14 bytes	20 bytes	548 bytes
			Total Payload =	3508 bytes

The details for the 3 fragments for each of the 5 ping requests are as follows

Details for fragments of request 1

Time of sending relative to start of capture	Source	Destination	Length
2.709193	10.184.27.165	62.210.18.40	1514 bytes
2.709193	10.184.27.165	62.210.18.40	1514 bytes
2.709193	10.184.27.165	62.210.18.40	582 bytes

.

Details for fragments of request 2

Time of sending relative to start of capture	Source	Destination	Length
3.726977	10.184.27.165	62.210.18.40	1514 bytes
3.726977	10.184.27.165	62.210.18.40	1514 bytes
3.726977	10.184.27.165	62.210.18.40	582 bytes

Details for fragments of request 3

Time of sending relative to start of capture	Source	Destination	Length
4.756741	10.184.27.165	62.210.18.40	1514 bytes
4.756741	10.184.27.165	62.210.18.40	1514 bytes
4.756741	10.184.27.165	62.210.18.40	582 bytes

Details for fragments of request 4

Time of sending relative to start of capture	Source	Destination	Length
5.767294	10.184.27.165	62.210.18.40	1514 bytes
5.767294	10.184.27.165	62.210.18.40	1514 bytes
5.767294	10.184.27.165	62.210.18.40	582 bytes

Details for fragments of request 5

Time of sending relative to start of capture	Source	Destination	Length
6.774096	10.184.27.165	62.210.18.40	1514 bytes
6.774096	10.184.27.165	62.210.18.40	1514 bytes
6.774096	10.184.27.165	62.210.18.40	582 bytes

All the 5 ping responses are fragmented and the details of the fragments of each of the 5 responses are as follows

Details for fragments of response 1

Receiving time wrt start of capture	Source	Destination	Length
2.863704	62.210.18.40	10.184.27.165	1514 bytes
2.863704	62.210.18.40	10.184.27.165	582 bytes
2.866380	62.210.18.40	10.184.27.165	1514 bytes

Details for fragments of response 2

Receiving time wrt start of capture	Source	Destination	Length
3.873034	62.210.18.40	10.184.27.165	1514 bytes
3.873034	62.210.18.40	10.184.27.165	582 bytes
3.873994	62.210.18.40	10.184.27.165	1514 bytes

Details for fragments of response 3

Receiving time wrt start of capture	Source	Destination	Length
4.900686	62.210.18.40	10.184.27.165	1514 bytes
4.912149	62.210.18.40	10.184.27.165	582 bytes
4.919018	62.210.18.40	10.184.27.165	1514 bytes

Details for fragments of response 4

Receiving time wrt start of capture	Source	Destination	Length
5.914485	62.210.18.40	10.184.27.165	1514 bytes
5.914485	62.210.18.40	10.184.27.165	582 bytes
5.914485	62.210.18.40	10.184.27.165	1514 bytes

Details for fragments of response 5

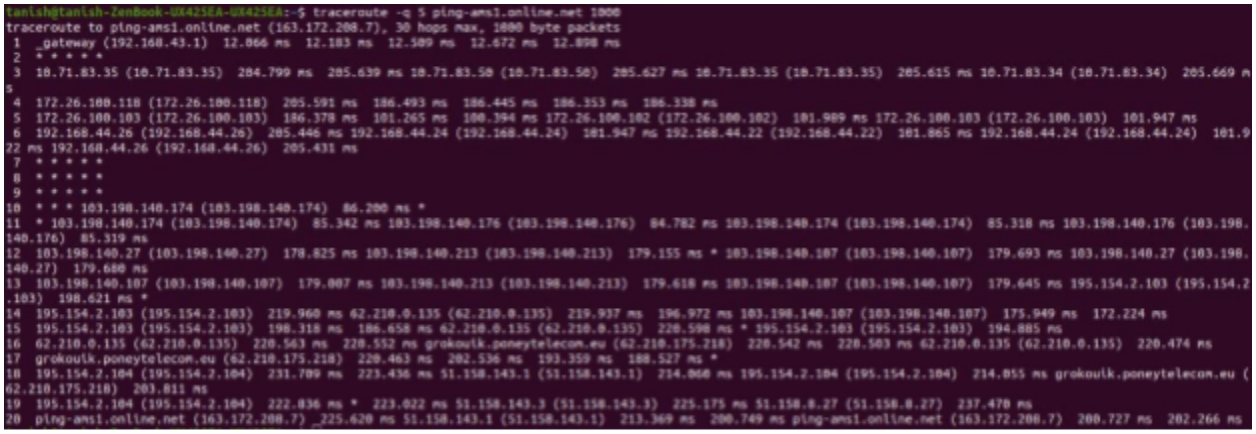
Receiving time wrt start of capture	Source	Destination	Length
6.919972	62.210.18.40	10.184.27.165	1514 bytes
6.929547	62.210.18.40	10.184.27.165	582 bytes
6.932741	62.210.18.40	10.184.27.165	1514 bytes

Respective fields indicating that the ping response is fragmented and eventually re-assembled

Traceroute task

Note: I changed the traceroute command to traceroute -q 5 ping-ams1.online.net 1000 because the number of hops was very large when the packet size was 3500 bytes.

1. The number of hops with packet size 1000 bytes was 20 as shown in the image below.
2.
    - Since there were 20 hops and each hop receives and returns 5 packets of data ideally the number of packets exchanged should be 20*5*2 = 200. Hence the client ideally sends 100 packets of data and each intermediate router (20 of them) return 5 packets of data each.
    - To find the packets that are sent from our machine we set the filter ip.dst == 163.172.208.7 (ping-ams1.online.net) since all outgoing traceroute packers are have destination as the final server and to filter the incoming traceroute packets from the intermediate routers the filter is icmp since whenever the TTL value of a traceroute packet becomes zero the intermediate routers sends an ICMP response back.
    - The number of packets displayed by wireshark with these filters is 180. The number of response packets by the ICMP filter is 73 and the number of request packets are 107 .
    - This doesn't match because some packets might not be returned from the intermediate routers if the request is timed-out. Also sometimes the client sends outgoing packets with TTL greater than the number of hops to the server before the response packet from the server is received which increases the outgoing packets.
3.
    - For each packet sent from our host the source address and the destination address remain the same but the TTL (time to live) field keeps changing, increasing by one incrementally.
    - Each packet starts at the host and it is intended to be sent to the server and thus their address does not change. For every router traceroute increases TTL value by 1 so that when TTL becomes zero the corresponding router returns packets to the host.
    - The identification field and the header checksum in the IPv4 header is also different as is expected to be for each packet sent from the host.



Trace route to ping-ams1.online.net