COL334 Assignment 1

Aniruddha Deb
2020CS10869

August 2022

Networking Tools

IP address of My Machine

Default IP address (on the IITD Network) (10.184.12.111)

```
[~/NextSem/COL334] % ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether f4:5c:89:a9:03:83
        inet6 fe80::1494:634:ec2a:9abc%en0 prefixlen 64 secured scopeid 0x4
IP Address inet 10.184.12.111 netmask 0xfffe000 broadcast 10.184.31.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

IP address on connecting to Mobile Hotspot (172.20.10.3)

```
[~/NextSem/COL334] % ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether f4:5c:89:a9:03:83
        inet6 fe80::1494:634:ec2a:9abc%en0 prefixlen 64 secured scopeid 0x4
        inet 172.20.10.3 netmask 0xfffffff0 broadcast 172.20.10.15
        inet6 2402:3a80:c81:db74:81a:1f3:2058:71c2 prefixlen 64 optimistic autoconf secured
        inet6 2402:3a80:c81:db74:7dd4:d725:19f:b9b8 prefixlen 64 optimistic autoconf temporary
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

IP address of Google and Facebook

IP of google (142.250.196.78)

```
[~/NextSem/COL334] % nslookup google.com
Server:         172.20.10.1
Address:        172.20.10.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.196.78
```

IP of facebook (157.240.16.35)

```
[~/NextSem/COL334] % nslookup facebook.com
Server:         172.20.10.1
Address:        172.20.10.1#53

Non-authoritative answer:
Name:   facebook.com
Address: 157.240.16.35
```

IP of google using Cloudflare DNS (172.217.27.174)

```
[~/NextSem/COL334] % nslookup google.com 1.1.1.1
Server:         1.1.1.1
Address:        1.1.1.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.27.174
```

Pinging google

Pinging google with default settings

```
[~/NextSem/COL334] % ping -c 4 google.com
PING google.com (142.250.196.78): 56 data bytes
64 bytes from 142.250.196.78: icmp_seq=0 ttl=53 time=107.031 ms
64 bytes from 142.250.196.78: icmp_seq=1 ttl=53 time=93.801 ms
64 bytes from 142.250.196.78: icmp_seq=2 ttl=53 time=82.800 ms
64 bytes from 142.250.196.78: icmp_seq=3 ttl=53 time=96.283 ms

--- google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 82.800/94.979/107.031/8.612 ms
```

Pinging google with a larger packet size

```
[~/NextSem/COL334] % ping -c 4 google.com -s 1000
PING google.com (142.250.196.78): 1000 data bytes
76 bytes from 142.250.196.78: icmp_seq=0 ttl=53 time=102.207 ms
wrong total length 96 instead of 1028
76 bytes from 142.250.196.78: icmp_seq=1 ttl=53 time=90.813 ms
wrong total length 96 instead of 1028
76 bytes from 142.250.196.78: icmp_seq=2 ttl=53 time=94.324 ms
wrong total length 96 instead of 1028

--- google.com ping statistics ---
4 packets transmitted, 3 packets received, 25.0% packet loss
round-trip min/avg/max/stddev = 90.813/95.781/102.207/4.764 ms
```

Pinging google with varying TTL's

```
[~/NextSem/COL334] % ping -c 1 google.com -m 5
PING google.com (142.250.196.78): 56 data bytes
36 bytes from 10.174.166.2: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  88 5400 6e52   0 0000  01  01 416f 172.20.10.3  142.250.196.78


--- google.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
[~/NextSem/COL334] % ping -c 1 google.com -m 10
PING google.com (142.250.196.78): 56 data bytes
36 bytes from 182.19.106.105: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  08 5400 d90c   0 0000  01  01 d734 172.20.10.3  142.250.196.78


--- google.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
[~/NextSem/COL334] % ping -c 1 google.com -m 15
PING google.com (142.250.196.78): 56 data bytes
92 bytes from 108.170.253.113: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  60 5400 1913   0 0000  01  01 96d6 172.20.10.3  142.250.196.78


--- google.com ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
[~/NextSem/COL334] % ping -c 1 google.com -m 30
PING google.com (142.250.196.78): 56 data bytes
64 bytes from 142.250.196.78: icmp_seq=0 ttl=53 time=90.315 ms

--- google.com ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 90.315/90.315/90.315/0.000 ms
```

Traceroute iitd.ac.in

Traceroute via internal network

```
[~] % traceroute www.iitd.ac.in
traceroute to www.iitd.ac.in (10.10.211.212), 64 hops max, 52 byte packets
 1  10.184.0.14 (10.184.0.14)  4.548 ms  6.552 ms  2.532 ms
 2  10.254.236.10 (10.254.236.10)  7.915 ms  2.429 ms
    10.254.236.18 (10.254.236.18)  3.051 ms
 3  www.iitd.ac.in (10.10.211.212)  2.810 ms  6.430 ms  7.676 ms
```

Accessed via Intranet: All hops are on Private network routers

traceroute via mobile network (google.com was used as my mobile network couldn't ping/traceroute to iitd.ac.in)

```
[~] % traceroute google.com
traceroute to google.com (142.250.196.78), 64 hops max, 52 byte packets
 1  172.20.10.1 (172.20.10.1)  10.645 ms  3.461 ms  3.806 ms
 2  10.174.42.246 (10.174.42.246)  201.707 ms  168.259 ms  498.706 ms
 3  * * *
 4  10.174.165.81 (10.174.165.81)  57.702 ms  64.691 ms  73.359 ms
 5  10.174.166.2 (10.174.166.2)  94.993 ms  65.420 ms  65.718 ms
 6  * * *
 7  * * *
 8  10.174.41.205 (10.174.41.205)  88.938 ms  77.292 ms  76.161 ms
 9  118.185.22.90 (118.185.22.90)  55.000 ms  73.058 ms  55.021 ms
10  182.19.106.105 (182.19.106.105)  58.194 ms  462.465 ms  72.134 ms
11  72.14.205.216 (72.14.205.216)  59.699 ms  71.108 ms  59.807 ms
12  * * *
13  108.170.248.177 (108.170.248.177)  87.588 ms  56.729 ms  73.875 ms
14  108.170.248.163 (108.170.248.163)  75.011 ms
    108.170.248.179 (108.170.248.179)  56.685 ms
    108.170.248.171 (108.170.248.171)  54.353 ms
15  142.250.212.7 (142.250.212.7)  89.164 ms
    142.250.212.1 (142.250.212.1)  89.733 ms
    209.85.251.15 (209.85.251.15)  112.550 ms
16  108.170.253.97 (108.170.253.97)  124.753 ms
    108.170.253.113 (108.170.253.113)  88.710 ms
    108.170.253.97 (108.170.253.97)  88.926 ms
17  108.170.253.97 (108.170.253.97)  90.890 ms  106.072 ms
    142.251.55.121 (142.251.55.121)  84.397 ms
18  maa03s46-in-f14.1e100.net (142.250.196.78)  92.792 ms
    142.251.55.121 (142.251.55.121)  85.876 ms
    142.250.236.157 (142.250.236.157)  90.103 ms
```

Green IP represents our Mobile Hotspot

Private IP Addresses are highlighted in Red

Routers marked in yellow did not respond

Multiple IP addresses indicate multiple paths to destination

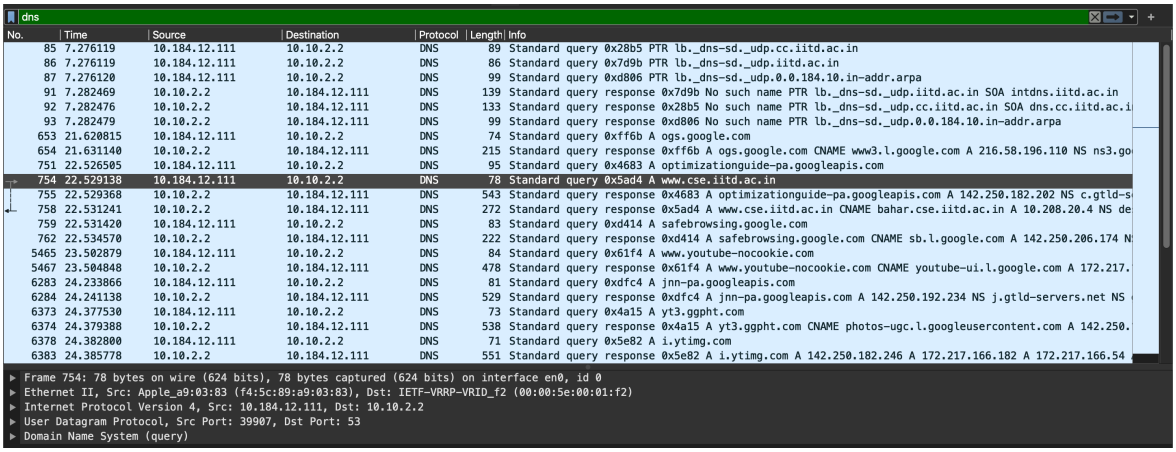
No paths used IPv6 queries. This is because MacOS ships with the BSD utilities `traceroute`, which uses IPv4 and `traceroute6`, which uses IPv6. The linux version of traceroute supports both protocols and takes a `-4` flag to force IPv4, or `-6` flag to force IPv6.

To get the missing routers to reply, we can:

1. Change the wait time for packets via `-w` to a longer value
2. Use the ICMP protocol via `-I`
3. Use a reserved port (one that's not blocked by any network firewalls configured by routers, eg any port whose number is less than 1024) via `-p` <port no>.

Packet Analysis

DNS Task



1. *Locate the DNS query and response messages. Are then sent over UDP or TCP?*

   The Query and Response are highlighted in the Screenshot. The DNS packets are sent over UDP, as is visible in the bottom of the screenshot.

2. *How many DNS queries are sent from your browser (host machine) to DNS Server(s)?*

   In the span of 25 seconds, 11 DNS queries are sent to the DNS server `10.10.2.2`. All are responded to within 25 seconds.

3. *How many DNS servers are involved?*

   There is only one server involved, `10.10.2.2` (this is the IITD internal DNS server, `dns1.cc.iitd.ac.in`)

4. *Which DNS Server replies with actual IP Address(es).*

   The server queried (`dns1.cc.iitd.ac.in`) replies with actual IP addresses

5. *Do all DNS servers respond?*

   Queries were sent out to only one DNS server, and all queries were responded to

6. *Clearly list the resource records involved in resolving the IP address of the site, mentioning, Name, value, type, TTL appropriately in the complete resolving process of this DNS conversation including query/queries and response/answer(s).*

Request Query:

Name	Type	Class
`www.cse.iitd.ac.in`	`A`	`IN`

Response Answers:

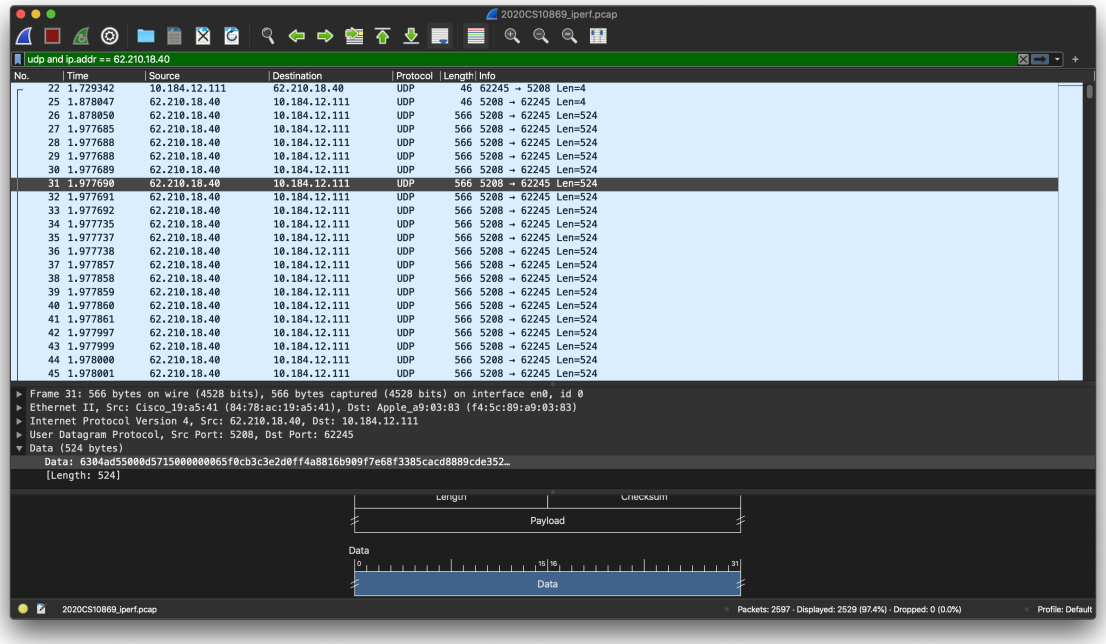
Name	Type	Class	TTL	len	CNAME	A
`www.cse.iitd.ac.in`	`CNAME`	`IN`	3600	8	`bahar.cse.iitd.ac.in`	
`bahar.cse.iitd.ac.in`	`A`	`IN`	3600	4		`10.208.20.4`

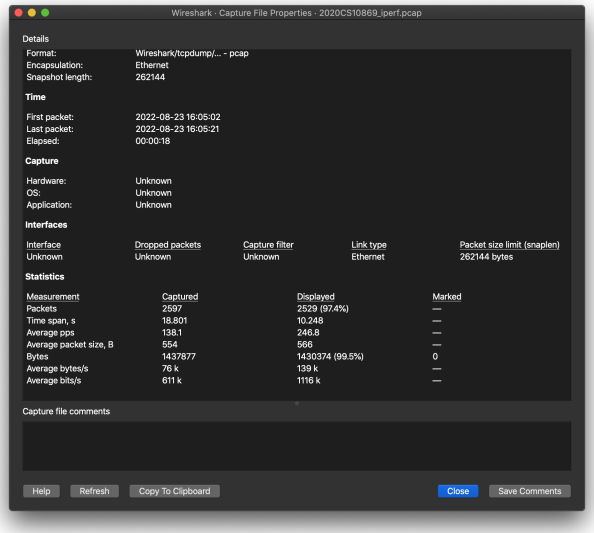
Authoritative nameserver(s) and additional data (Address records for Authoritative nameserver(s)) were also sent in the response, but there were no further requests to these nameservers for resolving the IP address of cse.iitd.ac.in.

Iperf Task

```
[~/NextSem/COL334/assignment1] % iperf3 -u -t 10 -c ping.online.net -p 5208 -R
Connecting to host ping.online.net, port 5208
Reverse mode, remote host ping.online.net is sending
[  7] local 10.184.12.111 port 62245 connected to 62.210.18.40 port 5208
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-1.00   sec   128 KBytes  1.05 Mbits/sec  0.043 ms  0/251 (0%)
[  7]   1.00-2.00   sec   128 KBytes  1.05 Mbits/sec  0.024 ms  0/250 (0%)
[  7]   2.00-3.00   sec   128 KBytes  1.05 Mbits/sec  0.071 ms  0/250 (0%)
[  7]   3.00-4.00   sec   128 KBytes  1.05 Mbits/sec  0.031 ms  0/250 (0%)
[  7]   4.00-5.00   sec   128 KBytes  1.05 Mbits/sec  0.029 ms  0/250 (0%)
[  7]   5.00-6.00   sec   128 KBytes  1.05 Mbits/sec  0.084 ms  0/250 (0%)
[  7]   6.00-7.00   sec   116 KBytes   952 Kbits/sec  0.140 ms  0/227 (0%)
[  7]   7.00-8.00   sec   140 KBytes  1.15 Mbits/sec  0.040 ms  0/274 (0%)
[  7]   8.00-9.00   sec   128 KBytes  1.05 Mbits/sec  0.078 ms  0/250 (0%)
[  7]   9.00-10.00  sec   128 KBytes  1.05 Mbits/sec  0.035 ms  0/250 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-10.00  sec  1.26 MBytes  1.06 Mbits/sec  0.000 ms  0/2502 (0%)  sender
[  7]   0.00-10.00  sec  1.25 MBytes  1.05 Mbits/sec  0.035 ms  0/2502 (0%)  receiver

iperf Done.
```





1. *How many UDP packets are exchanged in this communication between iperf3 client and remote server?*

   2529 packets are exchanged: The client sends 1 UDP packet to the server (to initiate the test), and the server sends 2528 UDP packets to the client

2. *Who is sending bulk data to whom? What is the average size of the packet sent?*

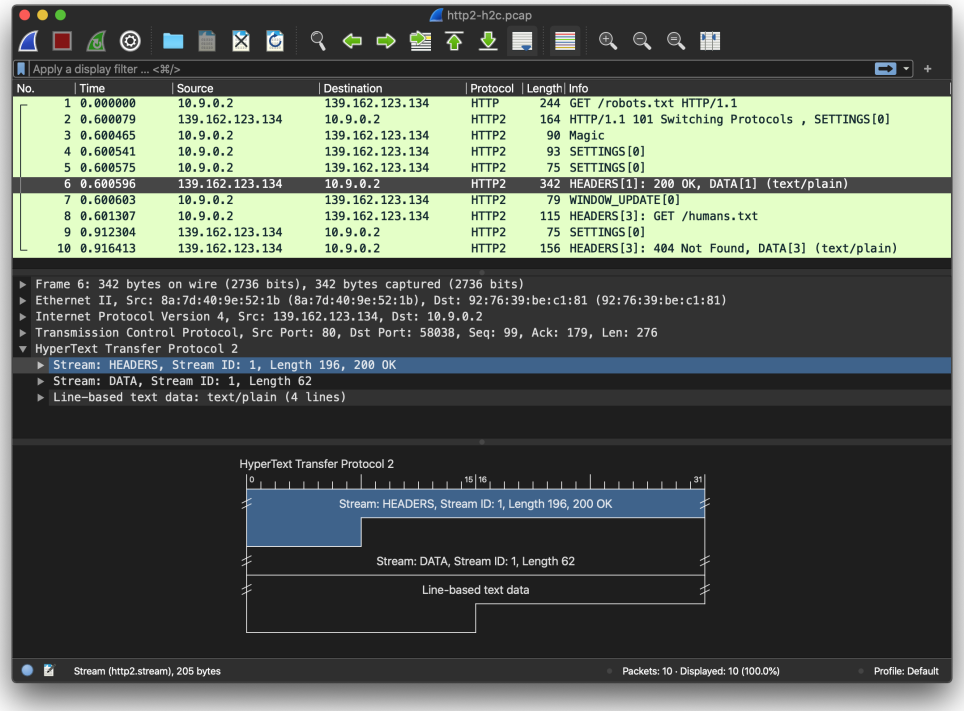
   The Server (`62.210.18.40`) sends bulk data to the client (our machine). The average packet size sent is **566** bytes

3. *Calculate the throughput (bytes transferred per unit time) for this UDP conversation using UDP's length field. Explain how you calculated this value using Wireshark capture in this experiment along with relevant screenshots. Verify your calculation with the one done by Wireshark using "Capture File properties" as well with the one displayed by iperf3 terminal. If you observe the major difference in your calculation and with the other two listed here, comment why and how?*

   The throughput is $2528 \times 566 \times 8/10 \approx 1.14$ Mbits/s. `iperf3` computed the bitrate as 1.06 Mbits/s, and Wireshark computed it as 1.11 Mbits/s (which is close to what we calculated it as).

   `iperf3` does not take into account header bytes while computing the throughput: it computes throughput as payload bits by time, and hence the value for the throughput there does not agree with the wireshark/manually computed values.

HTTP Task



1. *How many HTTP/2 and HTTP/1.1 packets are present?*

   There are **9** HTTP/2 and **1** HTTP/1.1 packets present: These are counted by applying http and http2 filters on wireshark and analysing the packets themselves: packet 2 (switching protocols) has both HTTP/1.1 and HTTP/2 data in it.

2. *How many HTTP/2 packets are exchanged between client and server here before the first object is fetched?*

   The first object is fetched at packet 6, so there are **4** HTTP/2 packets exchanged before the first object is fetched

3. *What main difference do you observe in headers of HTTP/2 packets displayed here, compared to the headers of HTTP/1.1 packets ?*

   HTTP/2 headers are sent in Streams inside the packet, separate from the data: these streams are encoded in Binary, and this supports multiplexing multiple streams in the same packet allowing for faster data transfer (shown in the screenshot). HTTP/1.1 headers are sent in plaintext and are actual 'headers', i.e. they come before the body in the packet.

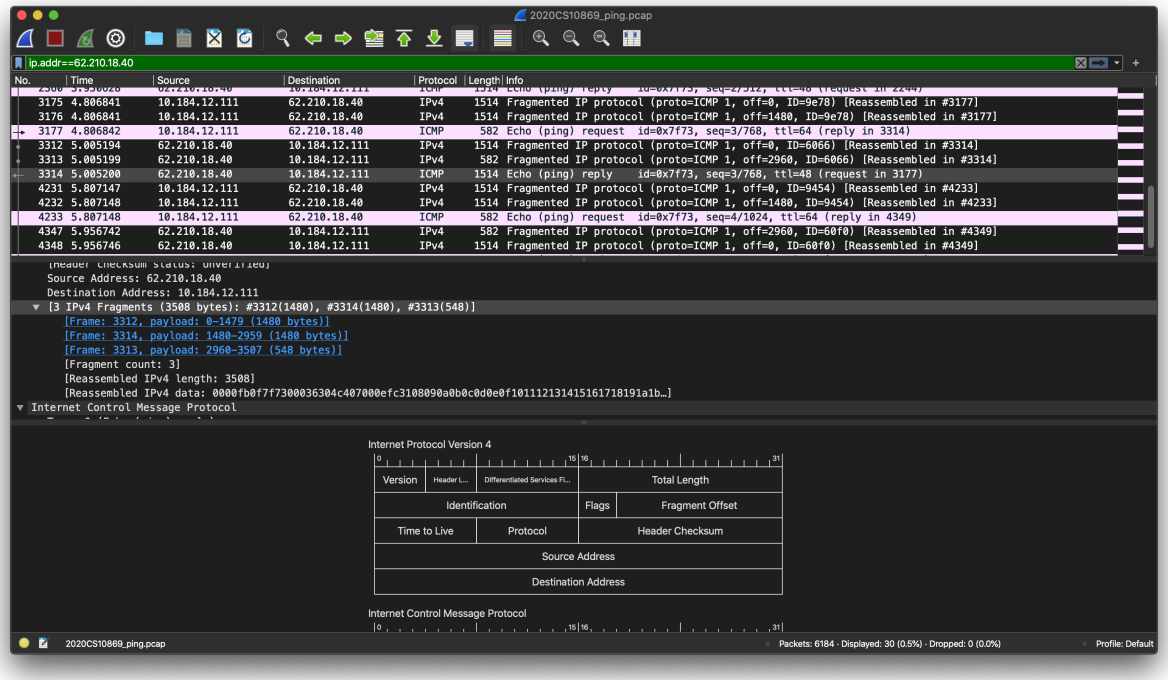
PING Task

NOTE

`ping-ams1.online.net` was inaccessible from ping. As a replacement, `ping.online.net` was used

```
[~/NextSem/COL334/assignment1] % ping -s 3500 ping.online.net -c 5
PING ping.online.net (62.210.18.40): 3500 data bytes
3508 bytes from 62.210.18.40: icmp_seq=0 ttl=48 time=310.585 ms
3508 bytes from 62.210.18.40: icmp_seq=1 ttl=48 time=141.894 ms
3508 bytes from 62.210.18.40: icmp_seq=2 ttl=48 time=144.734 ms
3508 bytes from 62.210.18.40: icmp_seq=3 ttl=48 time=198.483 ms
3508 bytes from 62.210.18.40: icmp_seq=4 ttl=48 time=149.699 ms

--- ping.online.net ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 141.894/189.079/310.585/64.181 ms
```



1. *How many total IP packets are exchanged in the communication between your host and the remote server representing* `ping.online.net` ?

   30 IP(v4) packets are exchanged between the host and the remote server: 15 from host to server and 15 from server to host

   Note that although only 5 ICMP packets are sent, the MTU (Maximum Transmission Unit) of the network is 1500 Bytes, hence each ICMP packet is fragmented into three IPv4 packets each, giving us 15 packets in one direction.

2. *What is the size of each ping request sent from your host to remote server?*

   Each ping request has a payload of 3500 bytes; with ICMP headers, this packet becomes 3508 bytes in total

3. *Make a table for each ping request packet sent from your host to remote, the respective field indicating it, if the request packet is fragmented or not. If packet is fragmented (add details on number of IP fragments and on each fragment), Time of sending each individual fragment/packet, length of the individual fragment/packet), time of receiving ping response, the respective field indicating if response packet is fragmented or not, if response packet is fragmented, include the number of IP fragments, total actual length of data carried by the respective fragment in respective ping request and response.*

no	req_frag	t_send	len	resp_frag	t_recv	len
1		1.79 s	3508		2.10 s	3508
	839	1.79 s	1480	916	2.10 s	548
	840	1.79 s	1480	917	2.10 s	1480
	841	1.79 s	548	918	2.10 s	1480
2		2.80 s	3508		2.94 s	3508
	1415	2.80 s	1480	1559	2.94 s	1480
	1416	2.80 s	1480	1560	2.94 s	548
	1417	2.80 s	548	1561	2.94 s	1480
3		3.80 s	3508		3.95 s	3508
	2242	3.80 s	1480	2358	3.95 s	1480
	2243	3.80 s	1480	2359	3.95 s	548
	2244	3.80 s	548	2360	3.95 s	1480
4		4.80 s	3508		5.00 s	3508
	3175	4.80 s	1480	3312	5.00 s	1480
	3176	4.80 s	1480	3313	5.00 s	548
	3177	4.80 s	548	3314	5.00 s	1480
5		5.80 s	3508		5.95 s	3508
	4231	5.80 s	1480	4347	5.95 s	548
	4232	5.80 s	1480	4348	5.95 s	1480
	4233	5.80 s	548	4349	5.95 s	1480

Traceroute Task

```
[~/NextSem/COL334/assignment1] % traceroute -q 5 ping.online.net 3500
traceroute to ping.online.net (62.210.18.40), 64 hops max, 3500 byte packets
 1  172.20.10.1 (172.20.10.1)  29.160 ms  4.579 ms  6.898 ms  5.808 ms  6.642 ms
 2  * * * * *
 3  * * * * *
 4  10.174.165.81 (10.174.165.81)  79.691 ms  77.779 ms  94.938 ms  77.254 ms  82.288 ms
 5  10.174.166.2 (10.174.166.2)  74.207 ms  71.357 ms  78.823 ms  81.095 ms  79.860 ms
 6  10.174.166.17 (10.174.166.17)  81.920 ms  79.436 ms  80.139 ms  81.184 ms  79.773 ms
 7  10.174.41.202 (10.174.41.202)  80.791 ms  84.830 ms  82.953 ms  78.747 ms  79.026 ms
 8  10.174.41.205 (10.174.41.205)  85.672 ms  85.860 ms  92.959 ms  87.974 ms  92.569 ms
 9  118.185.22.90 (118.185.22.90)  76.315 ms  75.855 ms  81.378 ms  80.728 ms  113.410 ms
10  182.19.108.204 (182.19.108.204)  114.514 ms  120.619 ms  100.805 ms  110.084 ms  154.290 ms
11  ae4-100-xcr1.sng.cw.net (212.165.2.1)  134.770 ms  135.473 ms  147.059 ms  132.702 ms  149.128 ms
12  ae34-xcr1.mrx.cw.net (195.2.2.57)  1902.724 ms  921.800 ms * *  364.811 ms
13  4.68.111.209 (4.68.111.209)  692.557 ms  206.660 ms  281.558 ms  244.661 ms  643.006 ms
14  ae1.3111.edge7.paris1.level3.net (4.69.133.234)  463.680 ms  656.566 ms  238.834 ms  349.276 ms  598.003 ms
15  212.3.235.202 (212.3.235.202)  250.526 ms  697.951 ms  822.473 ms  239.739 ms  241.253 ms
16  * * * * *
17  45x-s44-2-a9k2.dc3.poneytelecom.eu (195.154.1.107)  390.896 ms  921.783 ms  923.379 ms  920.092 ms  920.610 ms
18  ping.online.net (62.210.18.40)  921.263 ms  248.684 ms  236.330 ms  438.239 ms  792.755 ms
```

NOTE

The IITD network blocks traceroute, hence this was performed on a mobile network. Similar to the previous, `ping.online.net` was used as the target.

1. *How many hops are involved in finding the route to* `ping.online.net`

   18 hops are involved

2. *How many total IP packets are exchanged in the communication to get the final traceroute output of* `ping.online.net`*? How many of them are sent from client to remote machine (server/router) ? How many of them are sent from the remote machine (hop/server/router) to the local client ? Tabulate this with an entry for a router/server and the client too.*

   343 IP packets are exchanged to get the final traceroute:

   - **73** of these are ICMP responses from remote machines (hops/servers/routers) to our client.
     - 5 of these are ICMP Port unreachable (indicating the target was reached)
     - The remaining 68 are TTL exceeded. This is not a multiple of 5, as host #12 (`195.2.2.57`) dropped two UDP packets
   - **270** are packets sent out by our machine to various remote machines:
     - There are 18 attempted hops
     - Each attempt sends out 5 UDP packets
     - Because of the MTU of 1500 on the network, each UDP packet (of size 3500) is split into three packets

   Hence the total number is $18 \times 3 \times 5 = 270$.

Hops	IP	UDP packets sent	ICMP packets received
1	172.20.10.1	5	5
2	-	5	0
3	-	5	0
4	10.174.165.81	5	5
5	10.174.166.2	5	5
6	10.174.166.17	5	5
7	10.174.41.202	5	5
8	10.174.41.205	5	5
9	118.185.22.90	5	5
10	182.19.108.204	5	5
11	212.165.2.1	5	5
12	195.2.2.57	5	3
13	4.68.111.209	5	5
14	4.69.133.224	5	5
15	212.3.235.202	5	5
16	-	5	0
17	195.154.1.107	5	5
18	62.210.18.40	5	5

3. *Which fields in the IP datagram always change from one datagram to the next within this series of IP packets sent by your host/client ? Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?*

   Fields staying Constant:

   - Source Address: `172.20.10.3`. Must remain constant, as we are sending out packets.
   - Destination Address: `62.210.18.40`. Must remain constant, as this is the target address we're trying to trace the route to
   - Protocol: UDP. Must stay constant, as we don't change protocols in the middle of the process.

   Fields changing:

   - Time to Live (TTL): To isolate a single hop, the client must send UDP packets with incremental TTL values (varying from 1 to the maximum, here 64). Hence, this is incremented after every 5 UDP packets.
   - Flags: The fragmentation flag is set on some IP datagrams, to indicate that more fragments are present
   - Fragment Offset: This indicates the offset of that particular IPv4 packet, so that the recipient can reconstruct the complete UDP packet from the sequence of IPv4 packets it receives
   - Header Checksum: This would change, as the content of the header varies from packet to packet