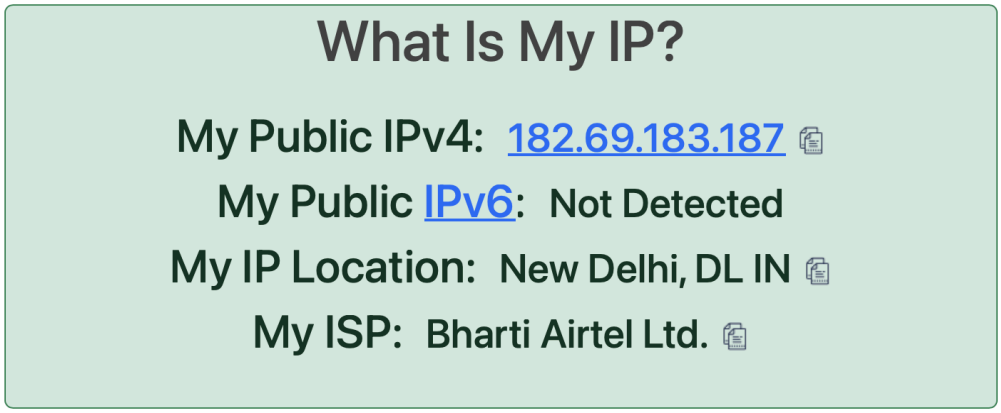
PA01: Using command-line utilities for network debugging

Paarth Goyal (2022343)

Github Link : https://github.com/pluto-tofu/Computer-Networks-CSE232-

Q1)

```
(base) paarthgoyal@192 ~ % ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether d0:88:0c:95:46:39
        inet6 fe80::189e:8a4f:febc:61d3%en0 prefixlen 64 secured scopeid 0xb
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
(base) paarthgoyal@192 ~ %
```



The primary network card on my device was en0, upon inspecting the information as shown by the ifconfig command the IP address was 192.168.1.3 and when checked on the “What Is My IP” website the IP address was 182.69.183.187 which is different from what was shown by ifconfig.

The IP addresses differ from each other as ifconfig shows the local or private IP address which is typically assigned to the device by the router while on the “What Is My IP” website we can see the public IP address which is used when visiting websites or when availing any kind of internet services. Please note that two devices on different networks can have the same private IP addresses.

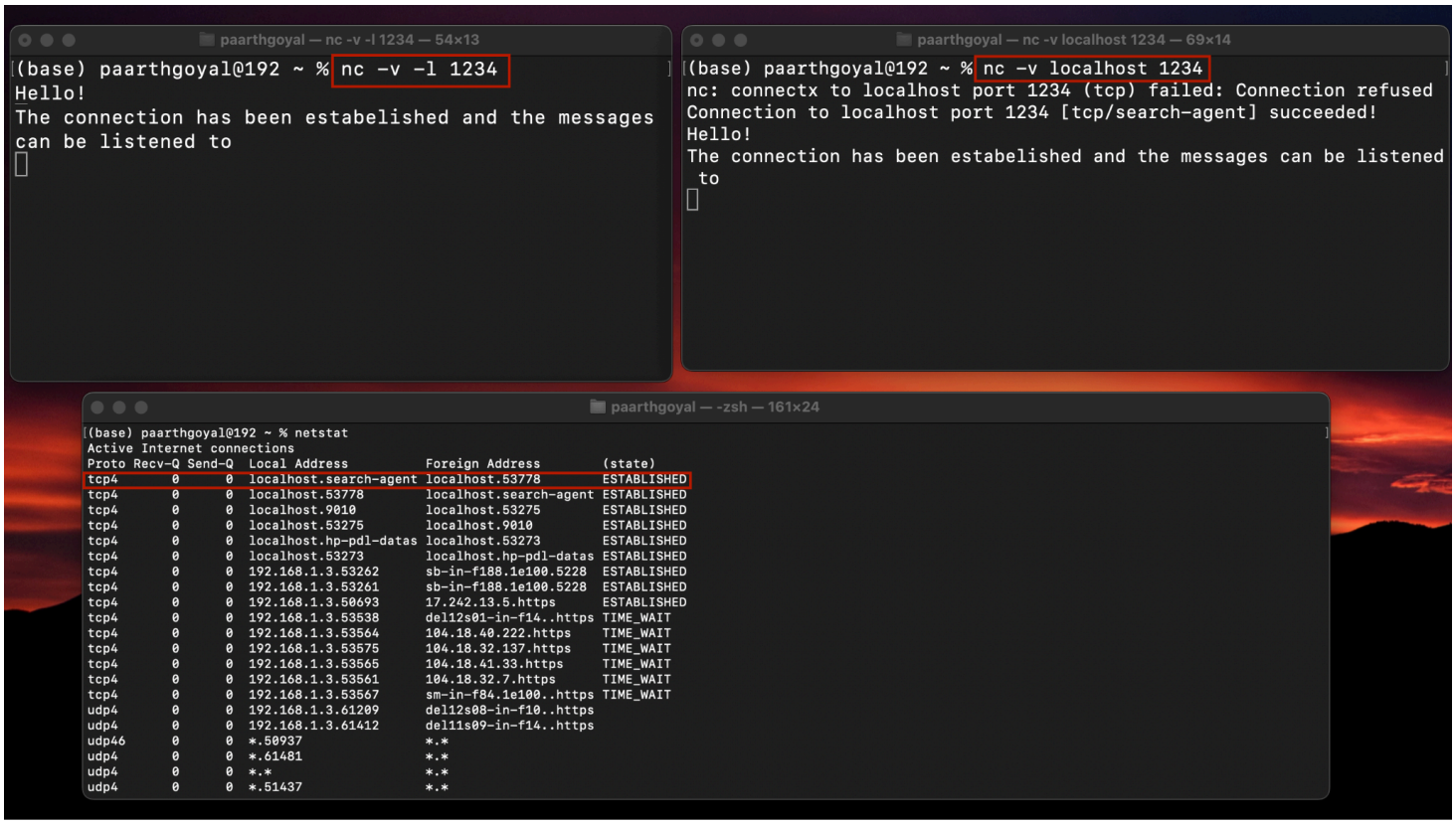
Q2)

```
(base) paarthgoyal@192 ~ % ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether d0:88:0c:95:46:39
        inet6 fe80::10ef:8ce2:90:ce64%en0 prefixlen 64 secured scopeid 0xb
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
(base) paarthgoyal@192 ~ % sudo ifconfig en0 inet 172.16.1.10
Password:
(base) paarthgoyal@192 ~ % ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether d0:88:0c:95:46:39
        inet6 fe80::10ef:8ce2:90:ce64%en0 prefixlen 64 secured scopeid 0xb
        inet 172.16.1.10 netmask 0xffff0000 broadcast 172.16.255.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
(base) paarthgoyal@192 ~ % sudo ifconfig en0 inet 192.168.1.3
(base) paarthgoyal@192 ~ % ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether d0:88:0c:95:46:39
        inet6 fe80::10ef:8ce2:90:ce64%en0 prefixlen 64 secured scopeid 0xb
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
(base) paarthgoyal@192 ~ %
```

The IP address of the device can be changed by using the command “sudo ifconfig en0 inet <new IP address>”, the new assigned IP address was selected randomly and the change can be seen in the screenshot. The IP address was reverted back to the original using the same command.

The IP address can also be reverted to the original one which was assigned by the dhcp upon restarting the device.

Q3)



On the terminal window to the left, the command “nc -v -l 1234” was used to set-up a server on the port number 1234 which listens for incoming connections (-l flag), The verbose (-v) option was passed in just for better understanding.

On the terminal window to the right an attempt was made to connect to the server using the command “nc -v localhost 1234”, the connection was successful and the text messages typed over here could be simultaneously seen appearing on the server side.

When using the netstat command, the state of the connection can also be observed which is shown as “ESTABLISHED” (Please check the boxed portion in red). The local address for this connection is shown as localhost.search-agent by default.

Q4) - a

```
Last login: Fri Aug 30 09:52:00 on ttys000
(base) paarthgoyal@192 ~ % nslookup google.in
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   google.in
Address: 142.250.196.164

(base) paarthgoyal@192 ~ % nslookup -type=soa google.in
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
google.in
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 668858537
        refresh = 900
        retry = 900
        expire = 1800
        minimum = 60

Authoritative answers can be found from:
google.in       nameserver = ns2.google.com.
google.in       nameserver = ns1.google.com.
google.in       nameserver = ns3.google.com.
google.in       nameserver = ns4.google.com.

(base) paarthgoyal@192 ~ % nslookup google.in ns1.google.com
Server:         ns1.google.com
Address:        216.239.32.10#53

Name:   google.in
Address: 142.250.193.36
```

When “nslookup google.in” was executed initially, the result was a non-authoritative response as the IP address for the domain was fetched from the local DNS cache. (Or some other intermediate cache which was not the authoritative server)

The next time when “nslookup -type=soa google.in” was executed, we were able to figure out the authoritative domain from where the responses for domain to IP conversion were originating, which was “ns1.google.com”.

After executing “nslookup google.in ns1.google.com” we were finally able to receive an authoritative response for google.in from “ns1.google.com”.

Q4) - b

```
(base) paarthgoyal@192 ~ % dig google.in

; <<>> DiG 9.10.6 <<>> google.in
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39620
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.in.                     IN      A

;; ANSWER SECTION:
google.in.              300     IN      A       142.250.196.164

;; Query time: 393 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Fri Aug 30 10:47:37 IST 2024
;; MSG SIZE  rcvd: 54
```

Considering the example of google.in, when the command “dig google.in” was executed we received a bunch of information. Out of this when we take a closer look at the answer section we see an “A” record which maps domain name to IPv4 address.

The time to live (TTL) is the time for which this entry will be stored in the local DNS cache, after the TTL has elapsed the entry will be removed from the cache.

In this case we can observe that the TTL is 300 seconds (default unit for TTL is seconds) which means that the entry will expire from the local DNS cache after 5 minutes.

Q5) - a

```
(base) paarthgoyal@192 ~ % traceroute google.in
traceroute to google.in (142.250.196.164), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  5.362 ms  3.912 ms  4.097 ms
 2  abts-north-dynamic-255.187.69.182.airtelbroadband.in (182.69.187.255)  13.106 ms  11.093 ms  9.106 ms
 3  125.18.240.149 (125.18.240.149)  10.946 ms
    125.18.240.153 (125.18.240.153)  12.939 ms  9.173 ms
 4  116.119.109.0 (116.119.109.0)  9.618 ms  9.744 ms  7.772 ms
 5  142.250.161.56 (142.250.161.56)  11.344 ms  12.930 ms  12.291 ms
 6  * * *
 7  142.251.52.202 (142.251.52.202)  15.123 ms
    142.251.54.98 (142.251.54.98)  11.329 ms
    142.251.54.74 (142.251.54.74)  10.001 ms
 8  216.239.54.92 (216.239.54.92)  13.275 ms
    192.178.82.234 (192.178.82.234)  11.630 ms
    216.239.54.92 (216.239.54.92)  10.027 ms
 9  209.85.250.56 (209.85.250.56)  11.706 ms  12.632 ms
    142.250.63.117 (142.250.63.117)  12.161 ms
10  142.250.239.229 (142.250.239.229)  50.473 ms
    172.253.72.137 (172.253.72.137)  46.592 ms
    192.178.248.45 (192.178.248.45)  55.724 ms
11  108.170.231.131 (108.170.231.131)  41.609 ms
    142.250.208.231 (142.250.208.231)  50.351 ms
    142.250.239.57 (142.250.239.57)  47.116 ms
12  maa03s47-in-f4.1e100.net (142.250.196.164)  43.435 ms
    108.170.231.131 (108.170.231.131)  42.005 ms
    maa03s47-in-f4.1e100.net (142.250.196.164)  41.981 ms
```

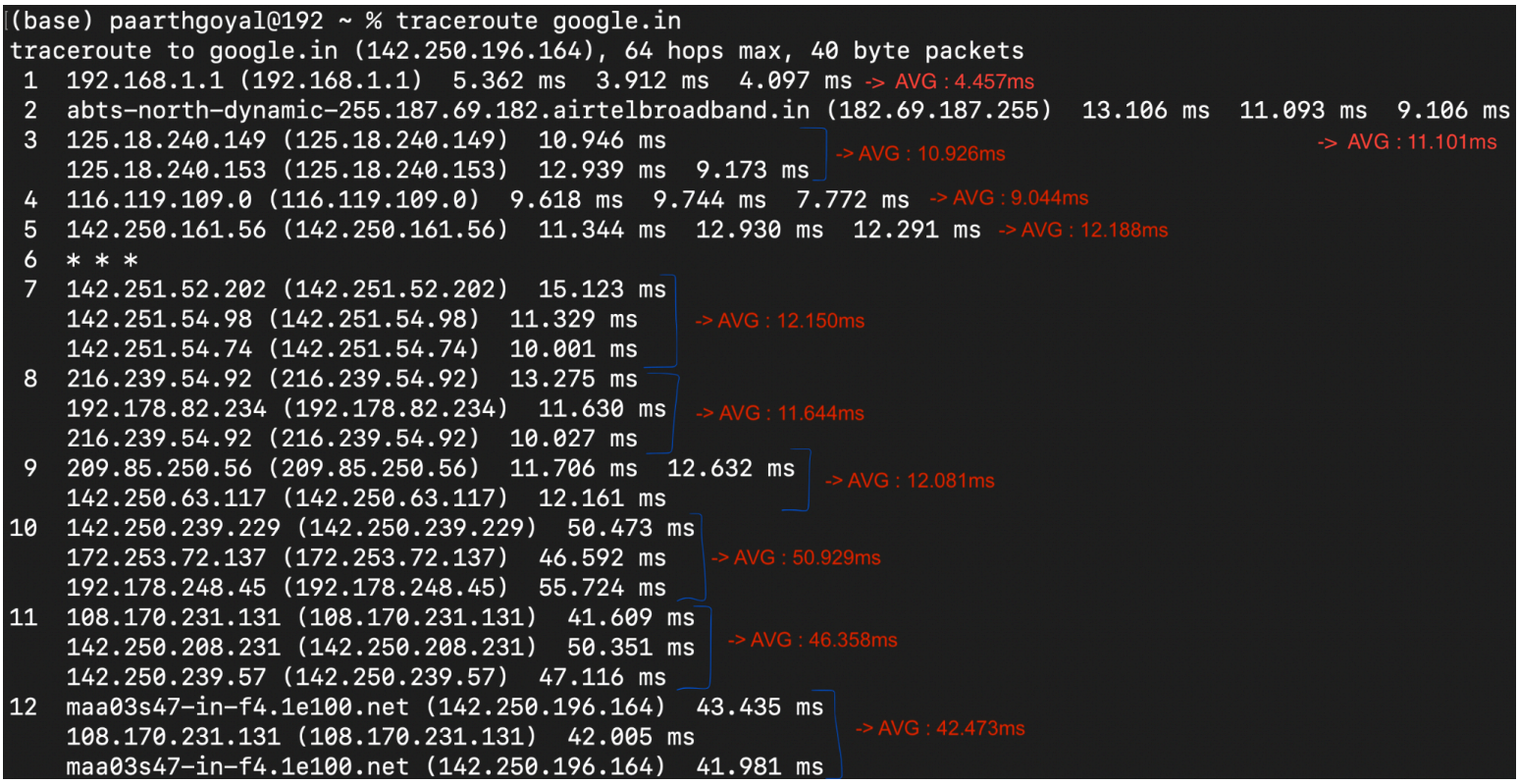
Upon executing the “traceroute google.in” command we can see all the intermediate hosts and their IP addresses (please check screenshot) through which the packets are transmitted.

We can observe that in front of some serial numbers there are three separate IP addresses, the reason for this is a technique known as load balancing. In this example there are three packets that are being sent to the destination, at some points those three packets are split up and sent forward using alternate paths, alternatively there could have been a single host which forwarded all the three packets but sending them through multiple different pathways to optimize performance.

Please note that the last host (12) is the destination.

If load balancing had not been implemented and instead of redirecting packets to multiple routes there would have been a single host then the total number of hosts would have been 11 (*** is still an intermediate host, it’s just that it’s information has not been disclosed or it was overloaded with requests).

Q5) - a (continued)



The average latency for each intermediate host and the destination has been calculated and shown in the above attached screenshot. For the hosts where there was load balancing, I have taken the weighted mean of the RTT values as the packets were sent along different routes.

Please check the screenshot for all the values.

Q5) - b

```
Last login: Fri Aug 30 11:04:37 on ttys000
(base) paarthgoyal@192 ~ % ping google.in -c 50
PING google.in (142.250.183.4): 56 data bytes
64 bytes from 142.250.183.4: icmp_seq=0 ttl=119 time=39.042 ms
64 bytes from 142.250.183.4: icmp_seq=1 ttl=119 time=48.908 ms
64 bytes from 142.250.183.4: icmp_seq=2 ttl=119 time=47.662 ms
64 bytes from 142.250.183.4: icmp_seq=3 ttl=119 time=46.143 ms
64 bytes from 142.250.183.4: icmp_seq=4 ttl=119 time=49.692 ms
64 bytes from 142.250.183.4: icmp_seq=5 ttl=119 time=47.428 ms
64 bytes from 142.250.183.4: icmp_seq=6 ttl=119 time=46.224 ms
64 bytes from 142.250.183.4: icmp_seq=7 ttl=119 time=45.783 ms
64 bytes from 142.250.183.4: icmp_seq=8 ttl=119 time=38.085 ms
64 bytes from 142.250.183.4: icmp_seq=9 ttl=119 time=48.033 ms
64 bytes from 142.250.183.4: icmp_seq=10 ttl=119 time=46.142 ms
64 bytes from 142.250.183.4: icmp_seq=11 ttl=119 time=41.470 ms
64 bytes from 142.250.183.4: icmp_seq=12 ttl=119 time=48.694 ms
64 bytes from 142.250.183.4: icmp_seq=13 ttl=119 time=46.486 ms
64 bytes from 142.250.183.4: icmp_seq=14 ttl=119 time=37.506 ms
64 bytes from 142.250.183.4: icmp_seq=15 ttl=119 time=47.305 ms
64 bytes from 142.250.183.4: icmp_seq=16 ttl=119 time=37.396 ms
64 bytes from 142.250.183.4: icmp_seq=17 ttl=119 time=48.316 ms
64 bytes from 142.250.183.4: icmp_seq=18 ttl=119 time=48.312 ms
64 bytes from 142.250.183.4: icmp_seq=19 ttl=119 time=37.833 ms
64 bytes from 142.250.183.4: icmp_seq=20 ttl=119 time=40.089 ms
64 bytes from 142.250.183.4: icmp_seq=21 ttl=119 time=47.200 ms
64 bytes from 142.250.183.4: icmp_seq=22 ttl=119 time=37.734 ms
64 bytes from 142.250.183.4: icmp_seq=23 ttl=119 time=48.077 ms
64 bytes from 142.250.183.4: icmp_seq=24 ttl=119 time=47.160 ms
64 bytes from 142.250.183.4: icmp_seq=25 ttl=119 time=45.784 ms
64 bytes from 142.250.183.4: icmp_seq=26 ttl=119 time=46.751 ms
64 bytes from 142.250.183.4: icmp_seq=27 ttl=119 time=41.962 ms
64 bytes from 142.250.183.4: icmp_seq=28 ttl=119 time=46.928 ms
64 bytes from 142.250.183.4: icmp_seq=29 ttl=119 time=46.176 ms
64 bytes from 142.250.183.4: icmp_seq=30 ttl=119 time=48.753 ms
64 bytes from 142.250.183.4: icmp_seq=31 ttl=119 time=39.850 ms
64 bytes from 142.250.183.4: icmp_seq=32 ttl=119 time=42.325 ms
64 bytes from 142.250.183.4: icmp_seq=33 ttl=119 time=45.719 ms
64 bytes from 142.250.183.4: icmp_seq=34 ttl=119 time=48.749 ms
64 bytes from 142.250.183.4: icmp_seq=35 ttl=119 time=48.599 ms
64 bytes from 142.250.183.4: icmp_seq=36 ttl=119 time=48.113 ms
64 bytes from 142.250.183.4: icmp_seq=37 ttl=119 time=43.264 ms
64 bytes from 142.250.183.4: icmp_seq=38 ttl=119 time=49.318 ms
64 bytes from 142.250.183.4: icmp_seq=39 ttl=119 time=48.262 ms
64 bytes from 142.250.183.4: icmp_seq=40 ttl=119 time=46.803 ms
64 bytes from 142.250.183.4: icmp_seq=41 ttl=119 time=49.729 ms
64 bytes from 142.250.183.4: icmp_seq=42 ttl=119 time=47.513 ms
64 bytes from 142.250.183.4: icmp_seq=43 ttl=119 time=46.421 ms
64 bytes from 142.250.183.4: icmp_seq=44 ttl=119 time=46.088 ms
64 bytes from 142.250.183.4: icmp_seq=45 ttl=119 time=48.471 ms
64 bytes from 142.250.183.4: icmp_seq=46 ttl=119 time=47.522 ms
64 bytes from 142.250.183.4: icmp_seq=47 ttl=119 time=46.549 ms
64 bytes from 142.250.183.4: icmp_seq=48 ttl=119 time=44.905 ms
64 bytes from 142.250.183.4: icmp_seq=49 ttl=119 time=47.410 ms

--- google.in ping statistics ---
50 packets transmitted, 50 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 37.396/45.574/49.729/3.591 ms
```

Upon execution of the command “ping google.in -c 50” the result obtained can be seen in the attached screenshot.

For a round trip ie. packet is sent to destination and then a response packet is sent back from destination to source, some statistics are :

1) MIN time -> 37.396 ms
2) AVG RTT / AVG latency -> 45.574 ms
3) MAX time -> 49.729 ms
4) STD Deviation -> 3.591 ms

Q5) - c

When we add up all the latencies for the intermediate hosts along the route we get :

4.457 + 11.101 + 10.926 + 9.044 + 12.188 + 12.150 + 11.644 + 12.081 + 50.929 + 46.358 + 42.473 = 223.351ms

Now, 223.351ms is greater than average latency in part (b) i.e. 45.574 ms when we directly ping google.in. And this is obvious as during traceroute we essentially ping each intermediate host, suppose for example consider the following route :

Source -> HOST A -> HOST - B -> HOST - C -> Destination

If we are currently at Source and we call “traceroute Destination” then the times that are outputted for each intermediate host are the RTT times from the source to intermediate host at that step. Ie RTT from source to A, RTT from source to B, RTT from source to C and finally RTT from source to Destination. This is why we say that we essentially ping each intermediate host.

As the sum will consider the RTT from source to intermediate host the result will be different from average latency in part (b), if we had the RTT from between pairs of successive hosts only then would the both of them matched.

The last entry ie. 42.473ms is the RTT from source to destination which is theoretically the same as average latency 45.574ms (this is not exactly same due to many variances when sending a packet) but when we consider the sum of all RTT’s from source to intermediate hosts that will not match.

Q5) - d

```
(base) paarthgoyal@192 ~ % traceroute google.in
traceroute to google.in (142.250.196.164), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  5.362 ms  3.912 ms  4.097 ms
 2  abts-north-dynamic-255.187.69.182.airtelbroadband.in (182.69.187.255)  13.106 ms  11.093 ms  9.106 ms
 3  125.18.240.149 (125.18.240.149)  10.946 ms
    125.18.240.153 (125.18.240.153)  12.939 ms  9.173 ms
 4  116.119.109.0 (116.119.109.0)  9.618 ms  9.744 ms  7.772 ms
 5  142.250.161.56 (142.250.161.56)  11.344 ms  12.930 ms  12.291 ms
 6  * * *
 7  142.251.52.202 (142.251.52.202)  15.123 ms
    142.251.54.98 (142.251.54.98)  11.329 ms
    142.251.54.74 (142.251.54.74)  10.001 ms
 8  216.239.54.92 (216.239.54.92)  13.275 ms
    192.178.82.234 (192.178.82.234)  11.630 ms
    216.239.54.92 (216.239.54.92)  10.027 ms
 9  209.85.250.56 (209.85.250.56)  11.706 ms  12.632 ms
    142.250.63.117 (142.250.63.117)  12.161 ms
10  142.250.239.229 (142.250.239.229)  50.473 ms
    172.253.72.137 (172.253.72.137)  46.592 ms
    192.178.248.45 (192.178.248.45)  55.724 ms
11  108.170.231.131 (108.170.231.131)  41.609 ms
    142.250.208.231 (142.250.208.231)  50.351 ms
    142.250.239.57 (142.250.239.57)  47.116 ms
12  maa03s47-in-f4.1e100.net (142.250.196.164)  43.435 ms
    108.170.231.131 (108.170.231.131)  42.005 ms
    maa03s47-in-f4.1e100.net (142.250.196.164)  41.981 ms
```

The maximum latency among the intermediate nodes is 55.724ms which is also shown (highlighted in red) in the above screenshot.

This maximum latency is not the same as the average latency obtained in part (b) which is 45.574ms.

The reasons for this difference ie. the intermediate host having a higher latency than the destination could be :

1) Relatively high load on the intermediate host.
2) Packets taking alternate routes to reach the destination due to load balancing.
3) Physical path differences or differences in network architecture and performance.

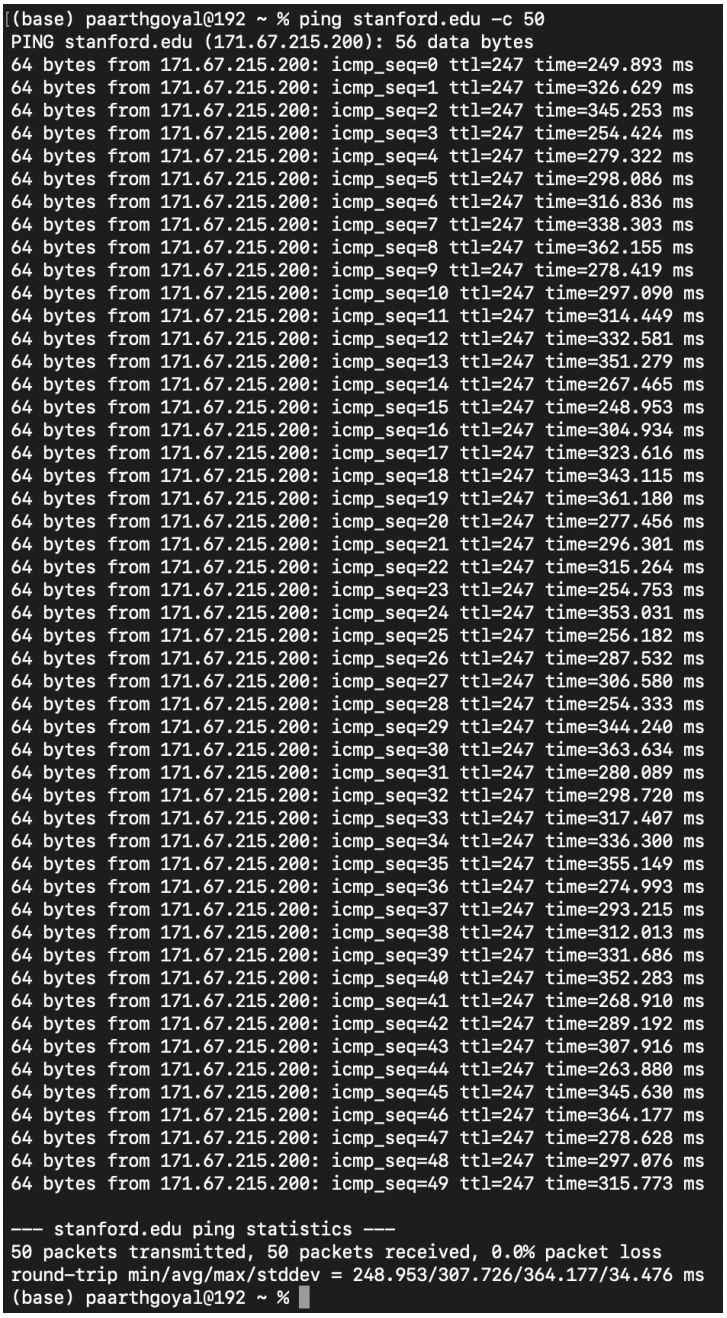
Q5) - e

We can observe multiple entries with different IP addresses on a single hop due to a technique known as load balancing (as discussed in answers to previous questions).

Incoming packets are split up and sent via different routes, for example 3 packets are forwarded by three different routers to three different routes.

This technique is used to optimize performance and ensure that one single router is not overloaded by too many packets hence splitting its load amongst multiple routers and reducing overall latency.

Q5) - f



Upon execution of the command “ping stanford.edu -c 50” the result obtained can be seen in the attached screenshot.

For a round trip ie. packet is sent to destination and then a response packet is sent back from destination to source, some statistics are :

1) MIN time -> 248.953 ms
2) AVG RTT / AVG latency -> 307.726 ms
3) MAX time -> 364.177 ms
4) STD Deviation -> 34.476ms

Q5) - g

```
(base) paarthgoyal@192 ~ % traceroute stanford.edu
traceroute to stanford.edu (171.67.215.200), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  4.022 ms  2.064 ms  3.361 ms
 2  abts-north-dynamic-255.187.69.182.airtelbroadband.in (182.69.187.255)  41.181 ms  22.000 ms  17.707 ms
 3  125.18.240.149 (125.18.240.149)  14.177 ms  14.627 ms
    125.18.240.153 (125.18.240.153)  16.399 ms
 4  116.119.57.43 (116.119.57.43)  249.905 ms
    182.79.146.238 (182.79.146.238)  306.548 ms
    116.119.44.132 (116.119.44.132)  306.083 ms
 5  * * *
 6  * * *
 7  * port-channel12.core3.sjc2.he.net (184.104.195.50)  262.977 ms *
 8  * * *
 9  stanford-university.e0-62.core2.pao1.he.net (184.105.177.238)  356.318 ms  306.375 ms  306.120 ms
10  campus-ial-nets-b-vl1118.sunet (171.66.255.228)  410.826 ms  407.816 ms
    campus-ial-nets-a-vl1018.sunet (171.64.255.228)  309.946 ms
11  * * *
12  web.stanford.edu (171.67.215.200)  390.195 ms  305.648 ms  307.211 ms
(base) paarthgoyal@192 ~ %
```

Upon executing the command “traceroute stanford.edu” we can observe that the number of hops is 12 (starting from IP address of personal device to IP address of router and so on).

Inspecting the output of the command “traceroute google.in” we can observe that the number of hops is also 12 (starting from IP address of personal device to IP address of router and so on).

Please note that the number of hops being the same in both the cases is coincidental and does not imply any special information about the routing paths.

Q5) - h

The average latency for stanford.edu was 307.726 ms while the average latency for google.in was 45.574 ms. The possible reasons for the difference between latencies could be :

1) Physical distance between personal device and the server hosting the website, google.in may have servers that are closer to our device when compared to stanford.edu.
2) Difference in routing path for both google.in and stanford.edu.
3) Difference in performance of intermediate hosts that forward the packets to destination.
4) Overall traffic on the network which can lead to higher latency while forwarding packets and receiving a response.
5) Firewalls and other security measures like rate limitation can also lead to higher latency.

Q-6)

```
(base) paarthgoyal@192 ~ % sudo ifconfig lo0 down
Password:
(base) paarthgoyal@192 ~ % ping 127.0.0.1 -c 5
PING 127.0.0.1 (127.0.0.1): 56 data bytes
ping: sendto: Can't assign requested address
ping: sendto: Can't assign requested address
Request timeout for icmp_seq 0
ping: sendto: Can't assign requested address
Request timeout for icmp_seq 1
ping: sendto: Can't assign requested address
Request timeout for icmp_seq 2
ping: sendto: Can't assign requested address
Request timeout for icmp_seq 3
^C
--- 127.0.0.1 ping statistics ---
5 packets transmitted, 0 packets received, 100.0% packet loss
(base) paarthgoyal@192 ~ % sudo ifconfig lo0 up
(base) paarthgoyal@192 ~ % ping 127.0.0.1 -c 5
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.110 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.156 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.162 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.134 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.142 ms

--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.110/0.141/0.162/0.018 ms
(base) paarthgoyal@192 ~ %
```

The IP address 127.0.0.1 (also known as loopback address) is special as the computer uses it to refer to itself. A server when running on the same device will be accessible under the name localhost or the IP address 127.0.0.1 which is the loopback address.

The ifconfig command provides a method to disable the loopback interface via the command “ifconfig lo0 down”. After this command is executed the device won’t be able to access this IP address.

While the loopback interface has been disabled we send 5 ping messages to it using “ping 127.0.0.1 -c 5” but they cannot reach the intended destination, eventually the command comes to a halt and we have to exit it by pressing ctrl + c. Upon exiting we can see that 5 packets were transmitted and 0 packets were received, resulting in failure of the ping command and 100 percent packet loss.

In the next step we use “ifconfig lo0 up” to enable the loopback interface again and ping the same IP address 5 times. This time ping works as intended with 0 percent packet loss. (Please check the attached screenshot)