

# Computer Networks COL334

Zuhaib Ul Zamann

18th August 2021

## 1 Question 1

### 1.1 Part a

1. The IP address of my machine when connected to reliance jio fiber is 192.168.29.116
2. The IP address of my machine when connected to mobile hotspot is 192.168.43.116
3. The IP addresses change as we change the ISPs.
4. Disconnecting and then reconnecting to the interface also changes the IP address.

### 1.2 Part b

1. The IP address associated with [www.google.com](http://www.google.com) is 142.251.42.68. Changing the name server to ns1.google.com gives IP address as 172.217.167.4
2. The IP address associated with [www.facebook.com](http://www.facebook.com) is 157.240.198.35 which comes as non-authoritative answer Changing the nameserver **d.ns.facebook.com** yeilds the IP address of the site as 157.240.239.35

### 1.3 Part c

The maximum packet size that is successful when i ping is 1472 on [www.iitd.ac.in](http://www.iitd.ac.in) and 1452 on [www.facebook.com](http://www.facebook.com), [google.com](http://google.com), though occasionally they accept 1472 byte packets too

On my system, I am able to send ping packets of maxsize 1472 to most of the sites.

However, I had the opportunity to check on other system also, where I was able to send packets of maximum size 1472 to [iitd.ac.in](http://iitd.ac.in) and rest as 1452.

```
C:\Users\khaleef naik>ping -l 1472 www.iitd.ac.in

Pinging www.iitd.ac.in [103.27.9.24] with 1472 bytes of data:
Reply from 103.27.9.24: bytes=1472 time=433ms TTL=46
Reply from 103.27.9.24: bytes=1472 time=471ms TTL=46
Reply from 103.27.9.24: bytes=1472 time=256ms TTL=46
Reply from 103.27.9.24: bytes=1472 time=49ms TTL=46

Ping statistics for 103.27.9.24:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 49ms, Maximum = 471ms, Average = 302ms
```

Figure 1: Packet size 1472- Successful ping

```
C:\Users\khaleef naik>ping -l 1473 www.iitd.ac.in

Pinging www.iitd.ac.in [103.27.9.24] with 1473 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 103.27.9.24:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Figure 2: Packet size 1473- Failed ping

```
C:\Users\khaleef naik>ping -l 1452 youtube.com

Pinging youtube.com [2404:6800:4002:81e::200e] with 1452 bytes of data:
Reply from 2404:6800:4002:81e::200e: time=40ms
Reply from 2404:6800:4002:81e::200e: time=30ms
Reply from 2404:6800:4002:81e::200e: time=53ms
Reply from 2404:6800:4002:81e::200e: time=29ms

Ping statistics for 2404:6800:4002:81e::200e:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 29ms, Maximum = 53ms, Average = 38ms

C:\Users\khaleef naik>ping -l 1453 youtube.com

Pinging youtube.com [2404:6800:4002:81e::200e] with 1453 bytes of data:
Request timed out.
Request timed out.
Request timed out.
```

Figure 3: Youtube Pings

## 1.4 Part d

1. Tracert to iid.ac.in gives the IP addresses of the intermediate routers in IPv4, i.e. it defaults to IPv4.[link](#).
2. Using tracert on google.com, facebook.com etc usually defaults to IPv6.[link](#)
3. We can force the tracert to use IPv4 by using the option -4, e.g.tracert -4 www.google.com.[link](#)

## 2 Question 2

### 2.1 Part a

The DNS request response time for <http://apache.org> is 0.07386ms

### 2.2 Part b

The approximate number of http requests that were generated on visiting <http://apache.org> is **74**. The requests can be found in the given [link](#). This tells us that webpage content comes from many subrequests sent to different webpages. Browsers show complex webpages in the order of the responses they receive.

## 2.3 Part c

The total time taken to load the website is  $(14.292597 - 7.671774) = 6.620823\text{ms}$   
The whole **apache packet trace** can be found in the link [link](#)

## 2.4 Part d

<http://www.cse.iitd.ac.in/> has less number(two packets) of http requests compared to <http://apache.org>, because all the subdomains follow https protocol which is not captured by wireshark compared to what apache does.

## 3 Question 3

Code submitted:

```
1 import subprocess
2 import re
3 import time
4 import matplotlib
5 import matplotlib.pyplot as plt
6 class Traceroute:
7     def __init__(self, destination, count):
8         self.destination = destination
9         self.version = "-4"
10        self.data = []
11        self.RTT = []
12        self.count = count
13    def run(self):
14        for ttl in range(1, 31):
15            bashCmd = ["ping", self.version, "-i", str(ttl), self.
destination]
16            process = subprocess.Popen(bashCmd, stdout = subprocess
.PIPE)
17            output, error = process.communicate()
18            process.kill()
19            output = output.split(b"\n")
20            output = [i.decode('utf-8') for i in output]
21            print("TTL = ", ttl)
22            s = None
23            for i in range(2, 5):
24                if (output[i].find('expired') == -1 and output[i].
find('timed out') == -1):
25                    print(output[i])
26                    print(output[i].split()[4].split("=")[1][:2])
27                    self.RTT.append(int(output[i].split()[4].split(
"=")[1][:2]))
28                    self.data.append(ttl)
29                    plt.plot(self.data, self.RTT)
30                    plt.xlabel("Number of Hops")
31                    plt.ylabel("Round Trip Time")
32                    plt.savefig("TTL vs RTT.png")
33                    plt.show()
34                    return
35                elif (output[i].find('timed out') == -1):
```

```

36         s = output[i]
37         if(s==None):
38             self.RTT.append(0.0)
39             self.data.append(ttl)
40             print("* * * RTT = 0 due to timeout")
41
42         else:
43             s = s.split()
44             bashCmd = ["ping", self.version, s[2][: -1]]
45             print(s[2][: -1])
46             process = subprocess.Popen(bashCmd, stdout =
subprocess.PIPE)
47             output, error = process.communicate()
48             process.kill()
49             output = output.split(b"\n")
50             output = [i.decode('utf-8') for i in output]
51             cnt = 0
52             time = 0
53             for i in range(2, 5):
54                 if(output[i].find('expired')== -1 and output[i].
find('timed out') == -1):
55                     st = output[i].split()[4].split("=")
[1][: -2]
56                     print(st, "ms",end = ' ')
57                     time+=int(st)
58                     cnt+=1
59                 else:
60                     print("*",end = ' ')
61             if(cnt == 0):
62                 self.data.append(ttl)
63                 self.RTT.append(0)
64                 print("RTT = 0 due to ping failure")
65             else:
66                 self.data.append(ttl)
67                 self.RTT.append(time/cnt)
68                 print("RTT = ", time/cnt)
69 print("Enter the destination to Traceroute: ")
70 url = input().split()[0]
71 print("Tracing route to ", url)
72 tracker = Traceroute(url, 0)
73 tracker.run()

```

The way the above code works is by sending ICMP requests and listening to ICMP failed responses to get the IP address of the router at which TTL expired, then pinging the same IP address to get the corresponding RTT. The code assumes that the path followed by the ping requests remains the same during the entire run of the traceroute.

The link to the image of self implemeted traceroute on iitd.ac.in is [IITD TRACER-ROUTE](#)

### 3.1 Observations

1. There seem to exist some **taboo values** for TTL in the sense that most of them give time out no matter the site I ping. These values are **7, 8 and 9**, with ISP as airtel.

2. With ISP as reliance no such taboo values exist. However, most of the intermediate routers returned by tracert or ping are private IP addresses in this case.
3. Pinging using IPv6 as IP address always replies with IPv6 address of destination and does not reveal the intermediate IP addresses.
4. IITD sites mostly choose default as IPv4 when pinging.
5. There is a general increase of RTT values with number of hops (excluding time outs and private IPs). This is more clear from the Youtube traceroute.

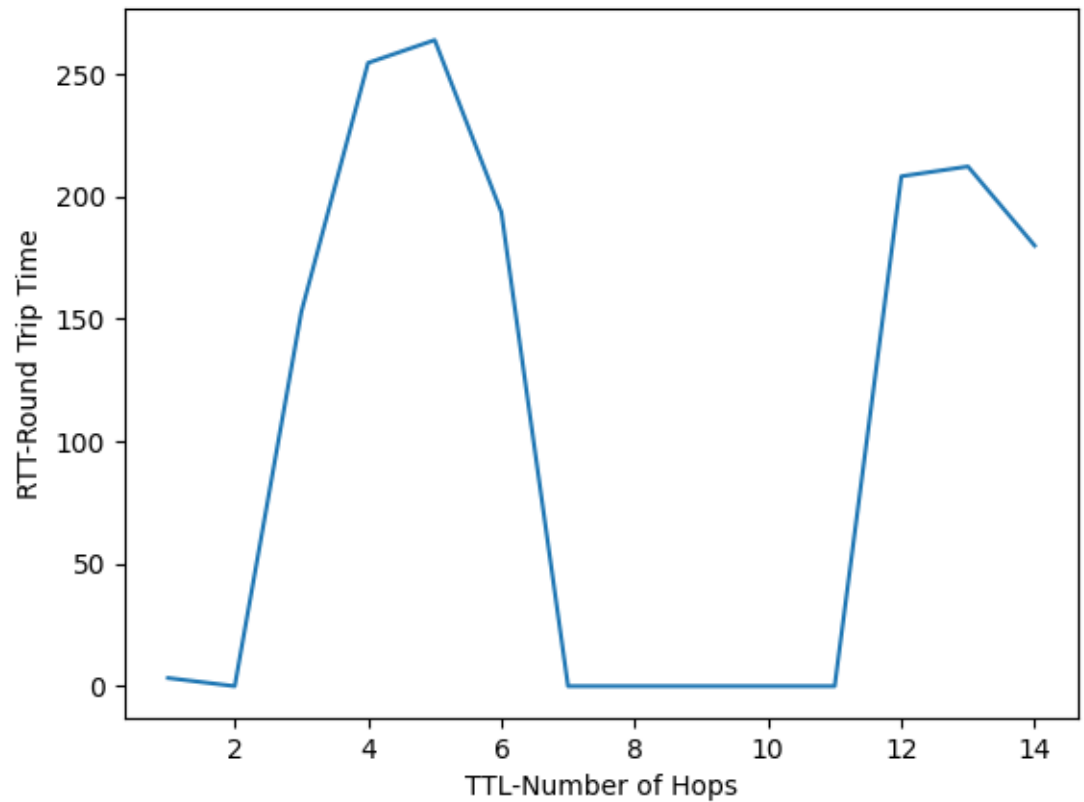


Figure 4: Plot of Number of hops vs RTT for iitd

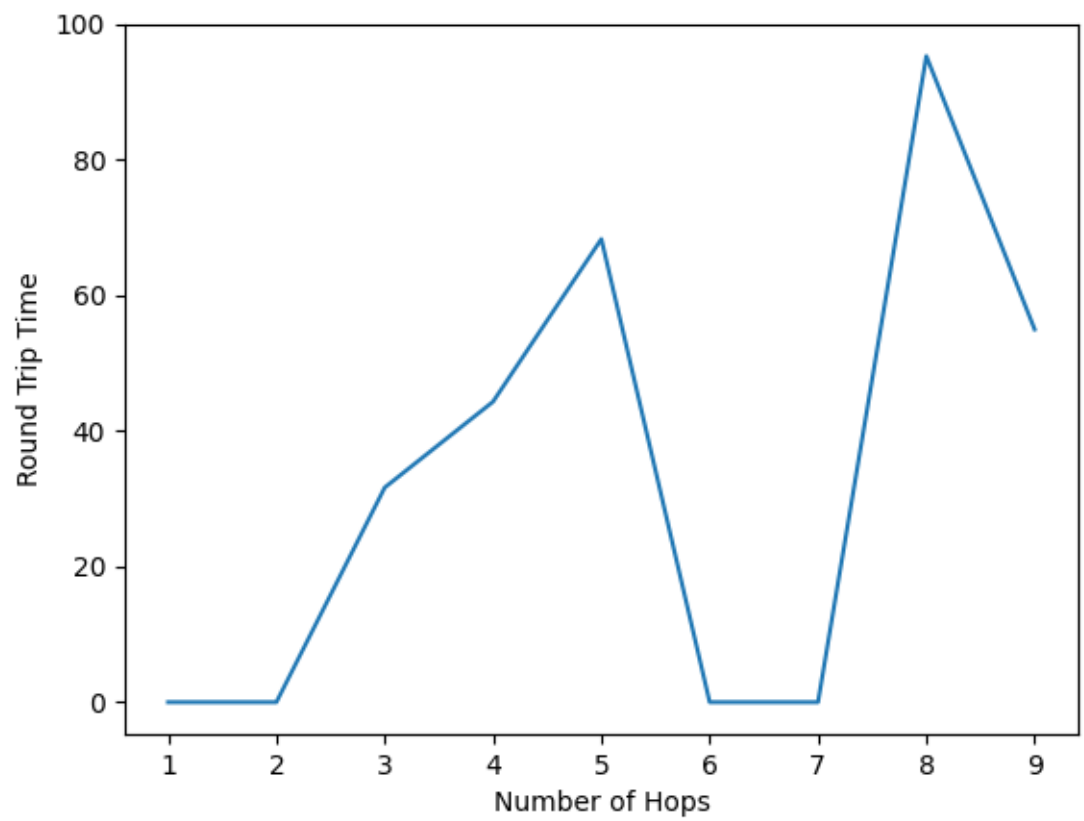


Figure 5: Plot of Number of hops vs RTT for Youtube

```

G:\WindowsData\SEM VII\COL334-Computer Networks\Assignments>python basic.py
Enter the destination to Traceroute:
youtube.com
Tracing route to youtube.com
TTL = 1
192.168.43.1
* * * RTT = 0 due to ping failure
TTL = 2
192.168.59.1
* * * RTT = 0 due to ping failure
TTL = 3
122.185.40.254
26 ms 34 ms 35 ms RTT = 31.666666666666668
TTL = 4
122.185.40.253
40 ms 35 ms 58 ms RTT = 44.333333333333336
TTL = 5
116.119.42.201
69 ms 86 ms 50 ms RTT = 68.33333333333333
TTL = 6
* * * RTTT = 0 due to timeout
TTL = 7
* * * RTTT = 0 due to timeout
TTL = 8
142.251.52.201
47 ms 121 ms 118 ms RTT = 95.33333333333333
TTL = 9
Reply from 142.250.194.14: bytes=32 time=55ms TTL=57
55

```

Figure 6: Youtube Traceroute