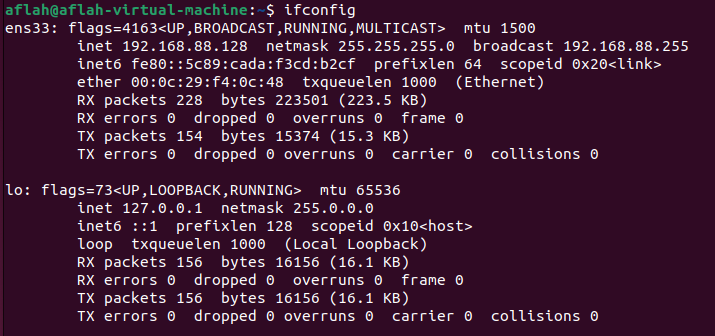
**CN Assignment**

**Mohammad Aflah Khan, 2020082**

**A1)**

**a)**

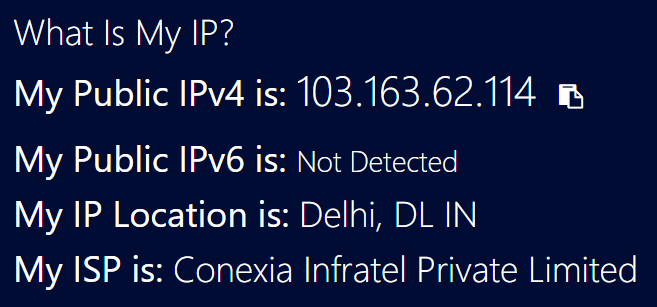


On Running ifconfig I see 2 outputs, here ens33 is the relevant output as ens33 as ens33 is the default network interface in Ubuntu. lo is just the loopback device which can be used to access networks locally.

Hence my IP Address according to ifconfig is 192.168.88.128

**b)**

If I open <https://www.whatismyip.com/> I see the following –



So according to the website my IP Address is 103.163.62.114

Both these values are different as ifconfig displays the local IP address, however when someone tries to access the internet they need to pass over switches and routers in the network. The website returns the IP address which the rest of the world will see (i.e. the global IP Address) when I send requests as this is the IP address provided to me by my ISP.

**A2)**

**a)**

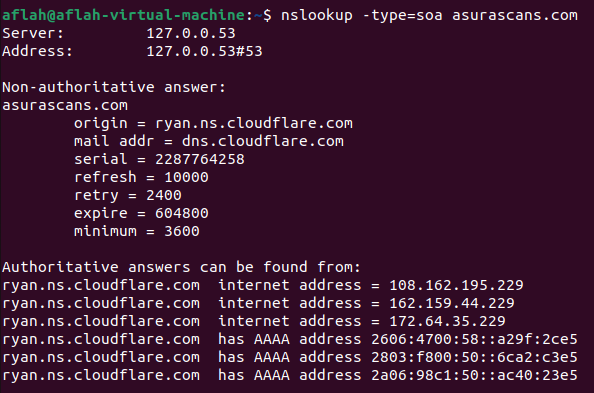
Finding the authoritative URL can be done is 2 steps:

1. First find the origin URL using a SOA (Search of Authority) query
2. Use the Origin URL for a lookup

This is because to get the authoritative answer we need to provide the authoritative name server as a part of the request. To procure the authoritative name server we can use a -type=soa flag which returns the same.

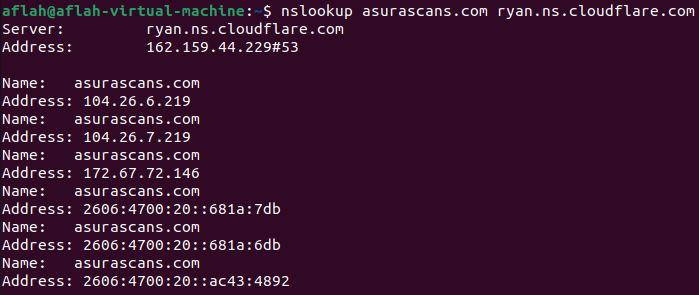
Let’s say we are interested to find details for asurascans.com which is a comic translation website.

Finding the Origin URL –

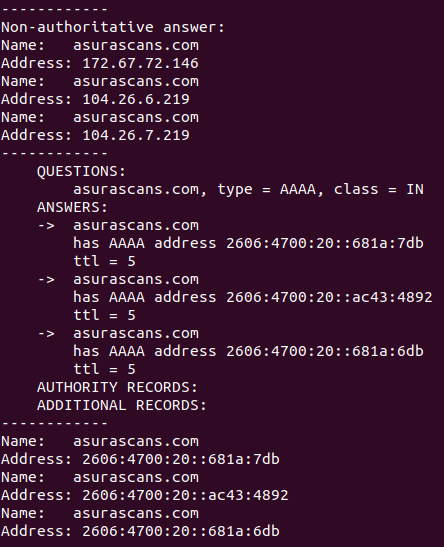
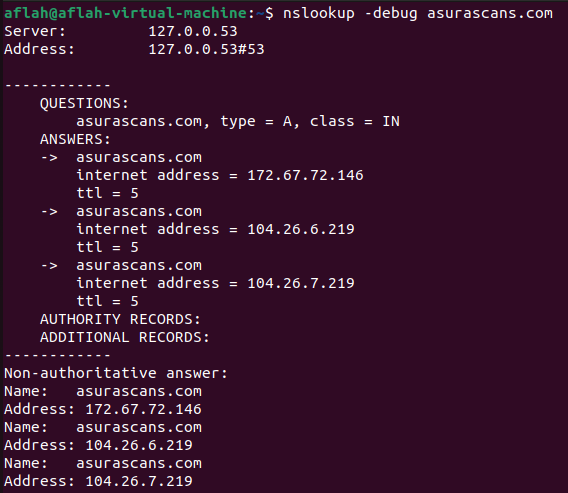


Hence our origin URL is ryan.ns.cloudflare.com

Doing lookup for this URL –



**b)**



Time to Live (TTL)–

Command used: nslookup -debug asurascans.com

IPv4 addresses are type A

IPv6 addresses are type AAAA

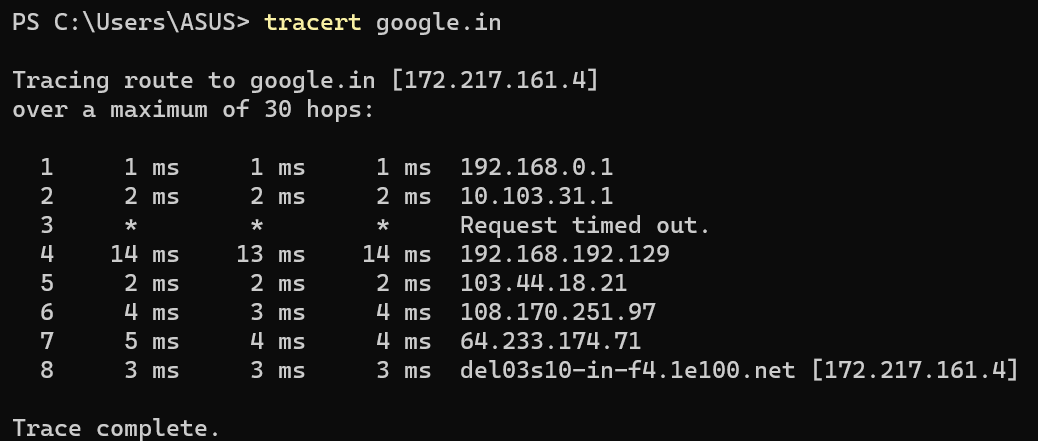
TTL for IPv4 is 5 seconds

TTL for IPv6 is 5 seconds

Hence the entries stay in the cache for 5 seconds after which they need to be refetched

**A3)**

Command used - tracert google.com (on my Windows Machine)

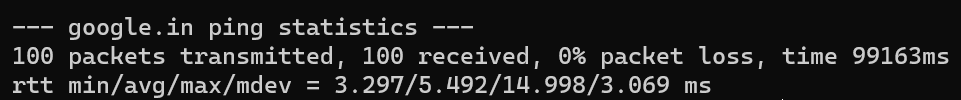


**a)**

As there are 8 Intermediate hosts and 1 Destination host, the average latency to each can be computed by first averaging the 3 values to get the average Round Trip Time and then then dividing by 2 gives us the average latency. Latency is approximately half the Round Trip Time assuming it’s uniform and same both ways.

|  |  |  |
| --- | --- | --- |
| IP Address | Average Latency Computation | Average Latency |
| 192.168.0.1 | ((1+1+1)/3)/2 | 0.5 ms |
| 10.103.31.1 | ((2+2+2)/3)/2 | 1 ms |
| 192.168.192.129 | ((14+13+14)/3)/2 | 6.833 ms |
| 103.44.18.21 | ((2+2+2)/3)/2 | 1 ms |
| 108.170.251.97 | ((4+3+4)/3)/2 | 1.833 ms |
| 64.233.174.71 | ((5+4+4)/3)/2 | 2.166 ms |
| 172.217.161.4 (Destination) | ((3+3+3)/3)/2 | 1.5 ms |

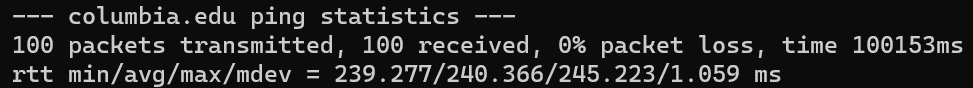
**b)**



Command Used - ping -c 100 google.in

Hence the Average Latency = Average RTT/2 = 5.492/2 = 2.746 ms

**c)**



Command Used - ping -c 100 columbia.edu

Hence the Average Latency = Average RTT/2 = 240.366/2 = 120.184 ms

**d)**

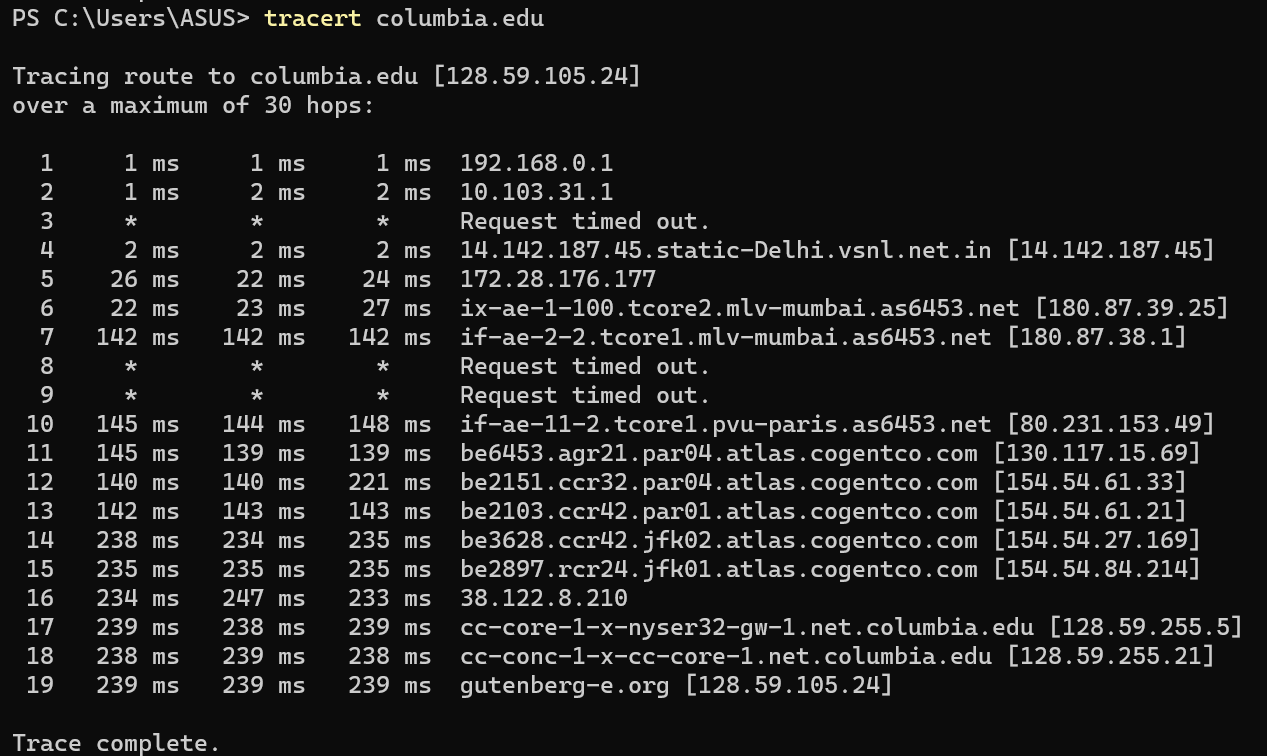
Sum of Average Latencies is 14.832 ms however actual average latency is only 2.746 ms. These are very different with the ping latency being significantly lower.

The reason for this is that tracert must wait at every intermediate host to get the response while ping simply sends packets which do not need to wait anywhere!

**e)**

The maximum ping latency is 6.833 ms while average latency is 2.746 ms. These values are more comparable as we are now not considering waiting at every intermediate host rather we only consider waiting at one host. So this acts similar to how ping does as it also just waits for response from destination to send back activity status and journey statistics.

**f)**



Number of hops for columbia.edu = 19

Number of hops for google.in = 8

Average Latency for columbia.edu = 120.184 ms

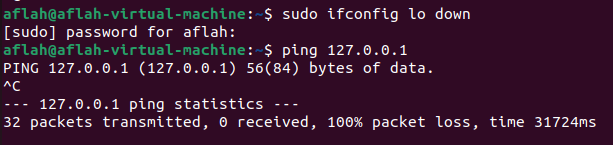
Average Latency for google.in = 2.746 ms

The latency and hops are much lower for google.in as it is a company which makes money off servicing users. If it had slower speeds people wouldn’t use it and switch to competitors. To ensure that it has high speeds it also has many more data centres across the world as compared to columbia.edu which is an educational website and does not have to focus on these high requirements.

**A4)**

To get 100% packet loss on our local server, we can simply have the loopback interface driver shut down. The command for the same is – sudo ifconfig lo down

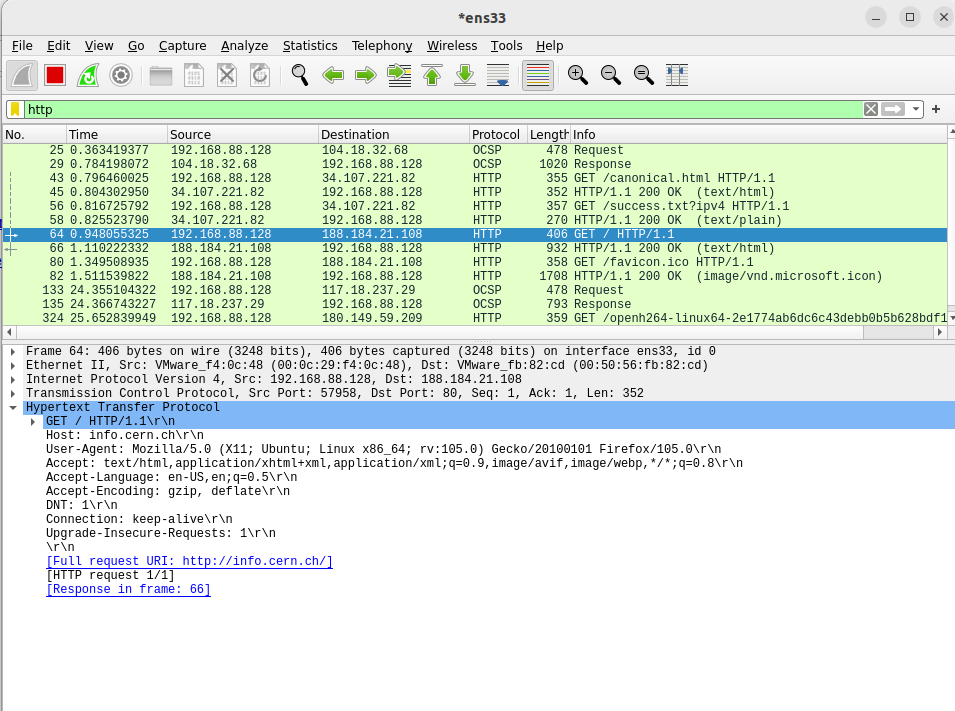
Now if we ping 127.0.0.1 we get 100% packet loss as there is no response since we shut down the loopback interface driver which is responsible for sending responses addressed to 127.0.0.1



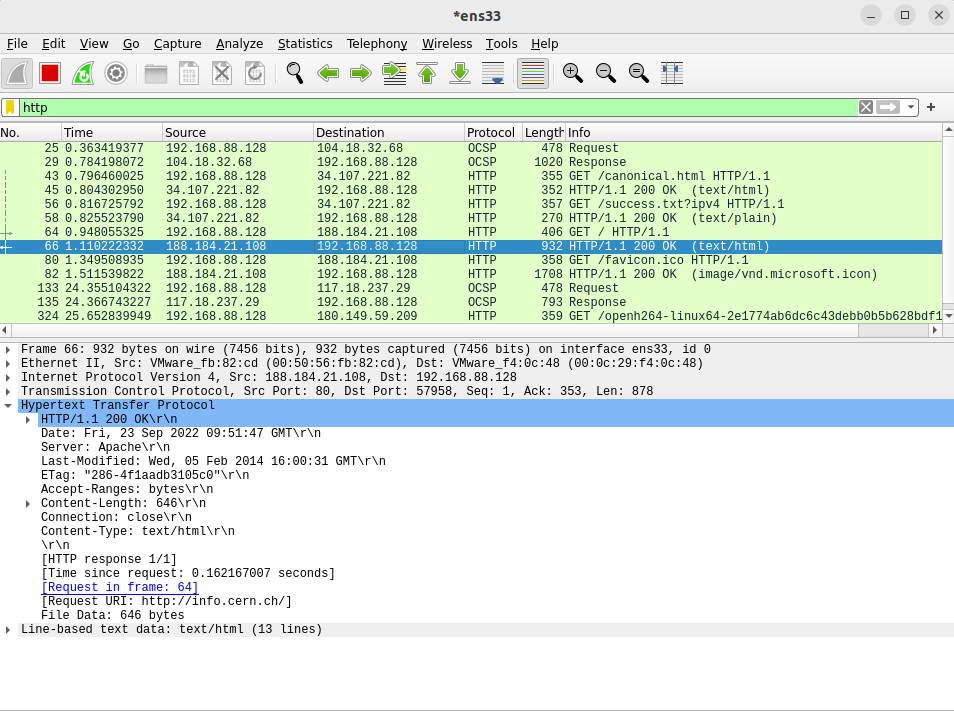
**A5)**

**The capture logs are also present in the zip**

* For HTTP Request Packages
  + HTTP Request Type: GET / HTTP/1.1\r\n
  + User Agent Type: User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:105.0) Gecko/20100101 Firefox/105.0\r\n
  + HTTP Request Packet URL: [Full request URI: <http://info.cern.ch/>]



* For HTTP Response Packets
  + HTTP Response Code: 200 OK
  + HTTP response description: A 200 OK Code implies the request was successful
  + Name and version of the web server: Apache



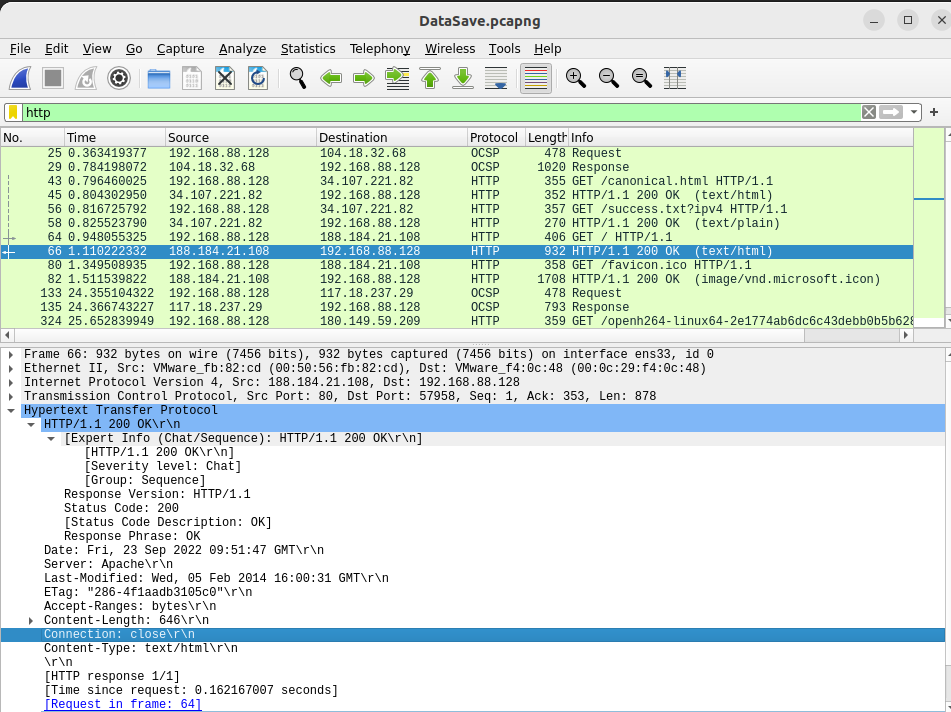
* 2 Web Objects are downloaded. These 2 Web Objects are procured via 2 GET Requests as visible in the Screenshot above.

The 2 Web Objects that are downloaded are a HTML Page and a Favicon

They are downloaded over different TCP Connections

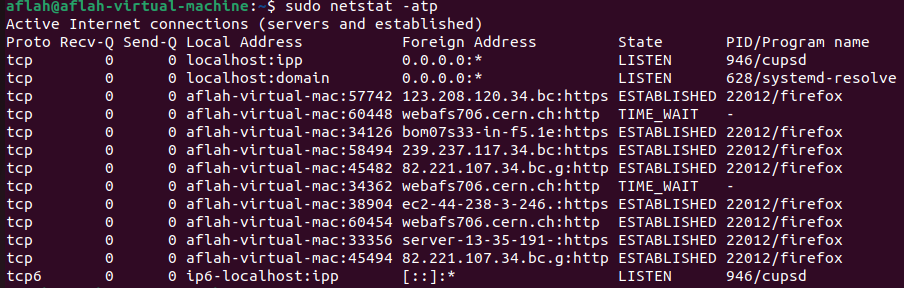
We observe 2 different Source Ports used for requesting the 2 different Web Objects. For the first request we see that source port 57598 is used while for the second request source port 57962 is used.

* The HTTP Connection is Non-Persistent as we use 2 different ports to request for 2 different objects and the Connection attribute in the Response Packet is ‘Close’



**A6)**

1. Command Used: netstat -atp

 This command lists all the active TCP Connections

1. The connection status is ESTABILISHED as seen in the terminal screenshot