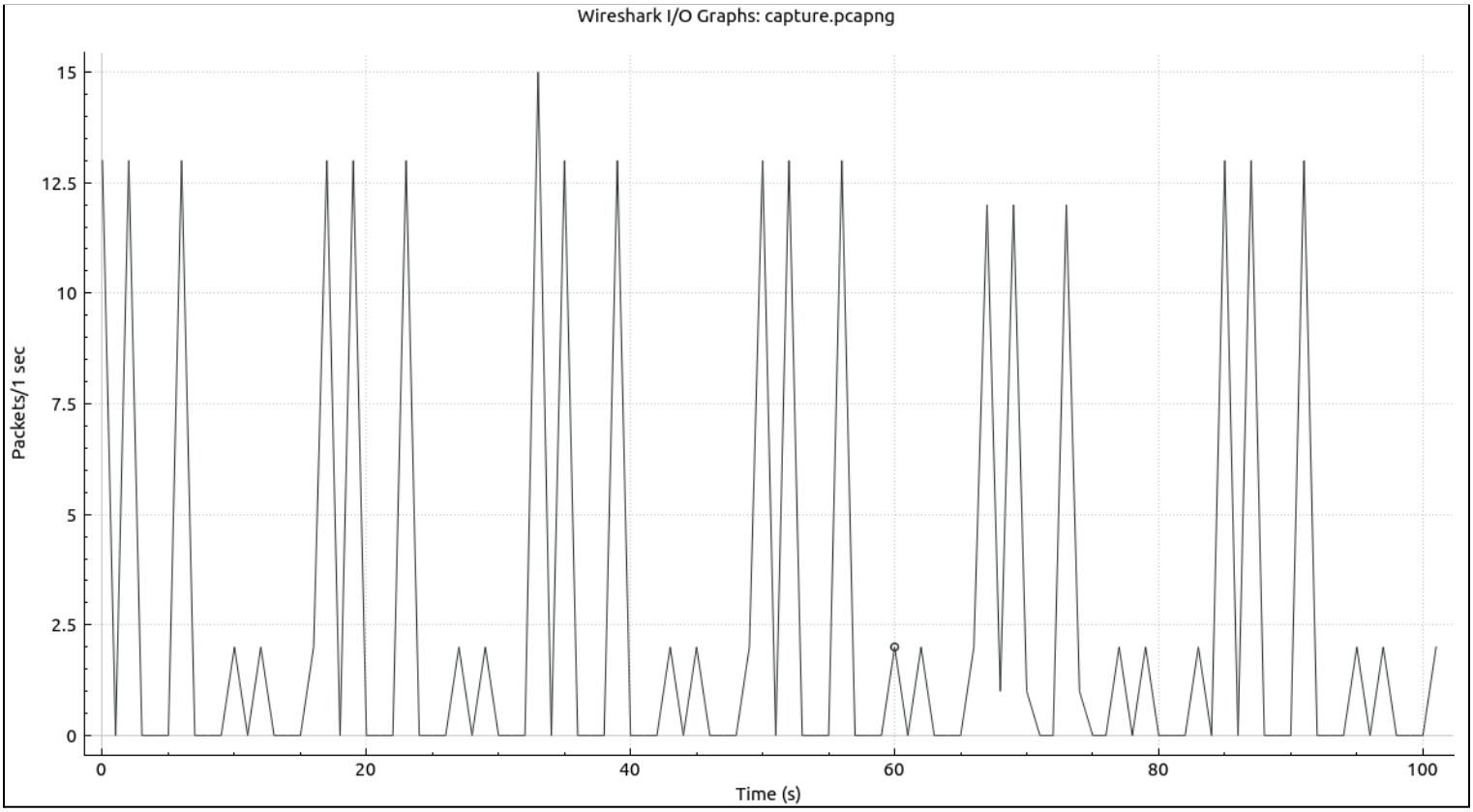
CSE232: Assignment 2

Mihir Chaturvedi (2019061)

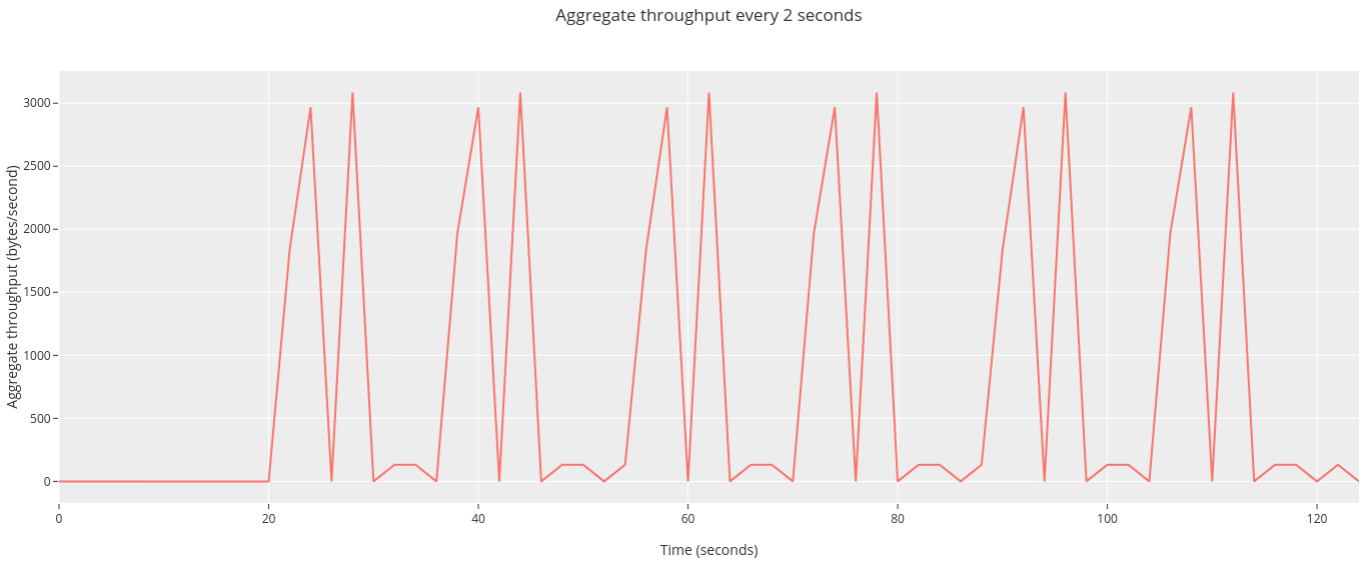
Question 1.

- I used Wireshark to analyze the packets on the `lo` interface.
- After starting Wireshark, I started the server program.
- The client program is run multiple times through the bash script that I had made.
- After 2 minutes, I stop the capture and filter the list using `tcp.port == 5000`.

Via Wireshark:

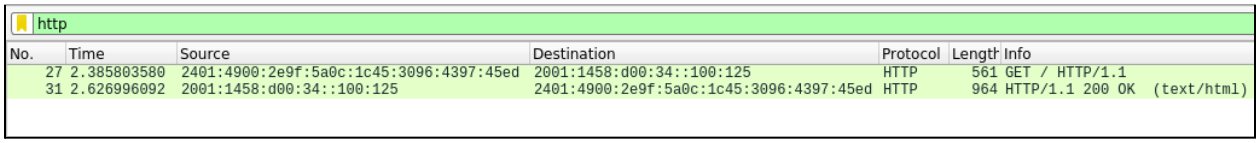


- After exporting the filtered packets to CSV format, cleaning it up, and calculating the aggregate throughput every two seconds, the plot:



Question 2.

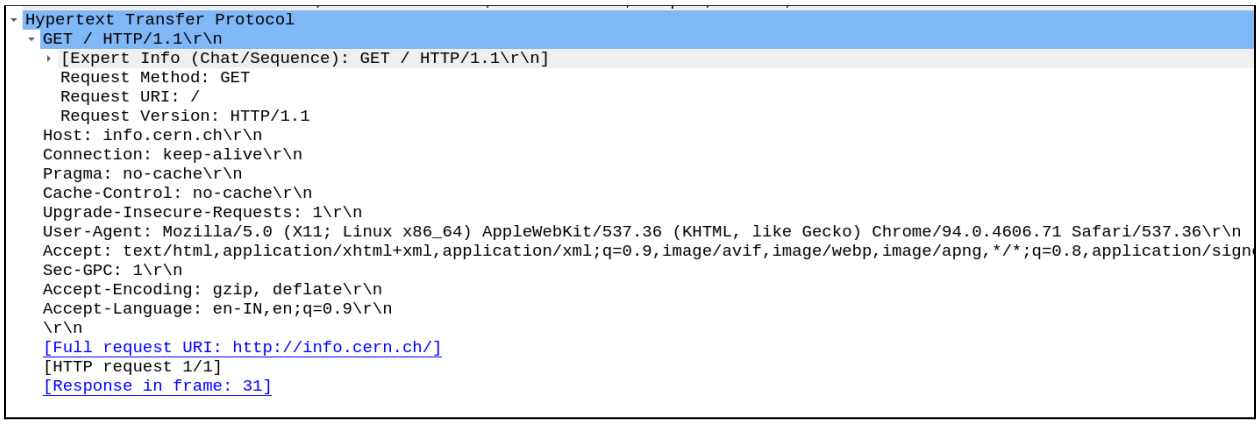
Following are the request and response packets exchanged between my network, (`2401:4900:2e9f:5a0c:1c45:3096:4396:45ed`) and info.cern.ch's network (`2001:1458:d00:34::100:125`):



http

No.	Time	Source	Destination	Protocol	Length	Info
27	2.385803580	2401:4900:2e9f:5a0c:1c45:3096:4397:45ed	2001:1458:d00:34::100:125	HTTP	561	GET / HTTP/1.1
31	2.626996092	2001:1458:d00:34::100:125	2401:4900:2e9f:5a0c:1c45:3096:4397:45ed	HTTP	964	HTTP/1.1 200 OK (text/html)

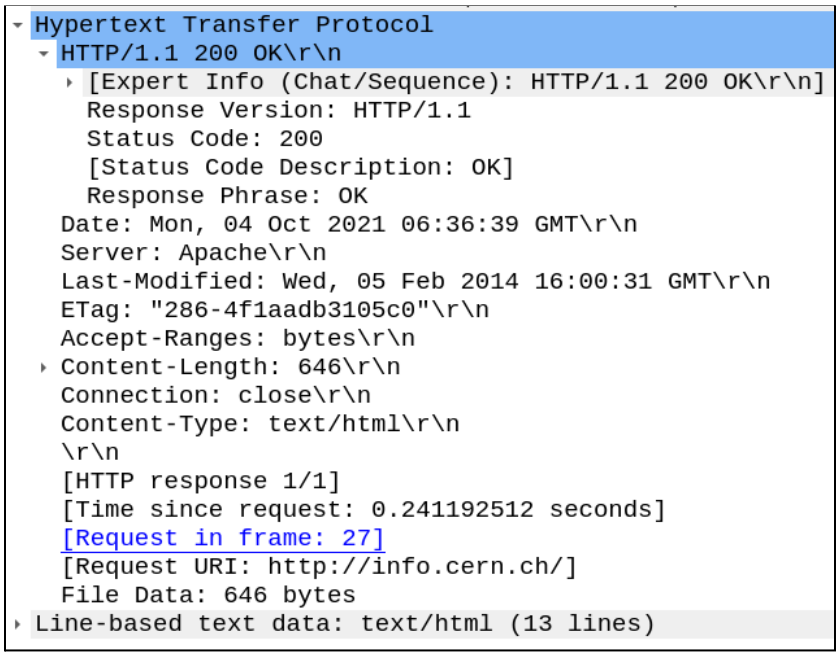
Request packet:



```
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
  Host: info.cern.ch\r\n
  Connection: keep-alive\r\n
  Pragma: no-cache\r\n
  Cache-Control: no-cache\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Safari/537.36\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sign
  Sec-GPC: 1\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-IN,en;q=0.9\r\n
  \r\n
  [Full request URI: http://info.cern.ch/]
  [HTTP request 1/1]
  [Response in frame: 31]
```

- Packet protocol type: HTTP (Hypertext Transfer Protocol)
- Request type: HTTP **GET** Request
- User-agent type: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Safari/537.36
- HTTP request packet's URL: /
- Name and version of the web server: Hostname of web server: info.cern.ch

Response packet:



- Packet protocol type: HTTP (Hypertext Transfer Protocol)
- HTTP response code: 200
- HTTP response description: OK

Question 3.

a)

`wlp1s0` is the name of my WiFi interface. Using the command `ifconfig wlp1s0`:

```
❯ ifconfig wlp1s0
wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.128.86  netmask 255.255.255.0  broadcast 192.168.128.255
        inet6 fe80::a58b:2c55:7348:330c  prefixlen 64  scopeid 0x20<link>
        inet6 2401:4900:2e9f:5a0c:53c2:6bdd:40f4:72de  prefixlen 64  scopeid 0x0<global>
        inet6 2401:4900:2e9f:5a0c:1c45:3096:4397:45ed  prefixlen 64  scopeid 0x0<global>
        ether 50:5b:c2:c6:5c:c7  txqueuelen 1000  (Ethernet)
        RX packets 560486  bytes 587166601 (587.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 274234  bytes 64164071 (64.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- LAN IPv4 address: 192.168.128.86
- LAN IPv6 addresses: fe80::a58b:2c55:7348:330c

b)

As listed on whatismyip.com, my public IPv4 address (at the time) is 106.223.22.38 and public IPv6 address (at the time) is 2401:4900:2e9f:5a0c:1c45:3096:4397:45ed.

The addresses listed via `ifconfig` are IP addresses of the Local Area Network created by the modem/router. These addresses are used to communicate between devices connected to the same local network.

The addresses listed on whatismyip.com are addresses for the Wide Area Network that are publicly visible and used to communicate with the *internet* and devices connected to the internet.

Question 4.

a)

To ping with a packet of MTU size 3000 bytes, we must send 3000 bytes - 8 bytes (ICMP header) - 20 bytes (IP header) = 2972 bytes of data.

Command: `ping -c 1 -s 2972 www.iiitd.ac.in`.
Explanation: Ping with 1 packet of data size 2972 bytes (total size 3000 bytes) to www.iiitd.ac.in

```
❯ ping -c 1 -s 2972 www.iiitd.ac.in
PING iiitd.ac.in (103.25.231.30) 2972(3000) bytes of data.

--- iiitd.ac.in ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

As we see, there is complete packet loss: the ping with one packet could not be done successfully. This is because the configured maximum MTU size for the connection through this network is capped at 1500 bytes. Since we are attempting to ping with an MTU > 1500 bytes, it fails.

The maximum MTU size can be noted using the `ifconfig` command.

```
~
❯ ifconfig wlp1s0
wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.128.86  netmask 255.255.255.0  broadcast 192.168.128.255
        inet6 fe80::a58b:2c55:7348:330c  prefixlen 64  scopeid 0x20<link>
        inet6 2401:4900:2e9f:5a0c:53c2:6bdd:40f4:72de  prefixlen 64  scopeid 0x0<global>
        inet6 2401:4900:2e9f:5a0c:1c45:3096:4397:45ed  prefixlen 64  scopeid 0x0<global>
        ether 50:5b:c2:c6:5c:c7  txqueuelen 1000  (Ethernet)
        RX packets 560486  bytes 587166601 (587.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 274234  bytes 64164071 (64.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

b)

To display all **connected** TCP connections on my machines, with the process PIDs:

`netstat --tcp --programs` or `netstat -tp`

```
~
❯ netstat --tcp --programs
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:55170         localhost:5054          ESTABLISHED 2518/python3
tcp        0      0 plibither8-linux:60736  91.108.23.100:https     ESTABLISHED 959256/telegram-des
tcp        0      0 localhost:5054          localhost:55166         ESTABLISHED 2211/python3
tcp        0      0 localhost:5054          localhost:55156         ESTABLISHED 2211/python3
tcp        0      0 plibither8-linux:54782  ec2-54-187-118-20:https ESTABLISHED -
tcp        0      0 localhost:55158         localhost:5054          ESTABLISHED 2542/python3
tcp        0      0 localhost:55162         localhost:5054          ESTABLISHED 2538/python3
tcp        0      0 localhost:5054          localhost:55134         ESTABLISHED 2211/python3
tcp        0      0 localhost:55166         localhost:5054          ESTABLISHED 2535/python3
tcp        0      1 plibither8-linux:40240  ec2-34-226-161-24:https SYN_SENT    915821/code
tcp        0      0 localhost:5054          localhost:55158         ESTABLISHED 2211/python3
tcp        0      0 localhost:55144         localhost:5054          ESTABLISHED 2526/python3
tcp        0      0 localhost:55140         localhost:5054          ESTABLISHED 2540/python3
tcp        0      0 localhost:5054          localhost:55170         ESTABLISHED 2211/python3
tcp        0      0 plibither8-linux:34862  stackoverflow.com:https ESTABLISHED 909295/brave --type
tcp        0      0 plibither8-linux:44384  91.108.56.199:https     ESTABLISHED 959256/telegram-des
```

To display all **listening** TCP connections on my machines, with the process PIDs:

`netstat --tcp --programs --listening` or `netstat -tpl`

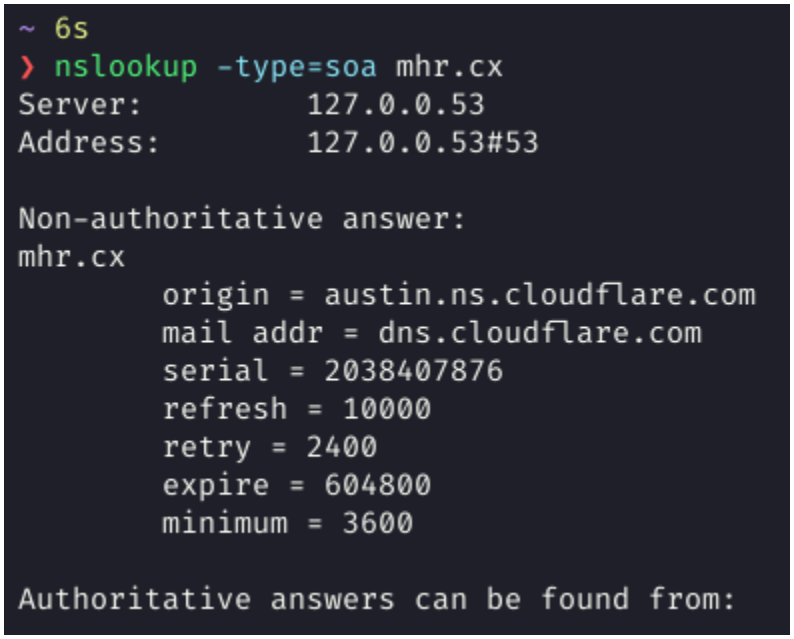
```
~ 18s
❯ netstat --tcp --programs -l
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN      -
tcp        0      0 plibither8-linux:domain 0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:5054          0.0.0.0:*               LISTEN      2211/python3
tcp6       0      0 [::]:16587              [::]:*                  LISTEN      2228/rescuetime
tcp6       0      0 [::]:1716               [::]:*                  LISTEN      2409/gjs
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN      -
```

Question 5.

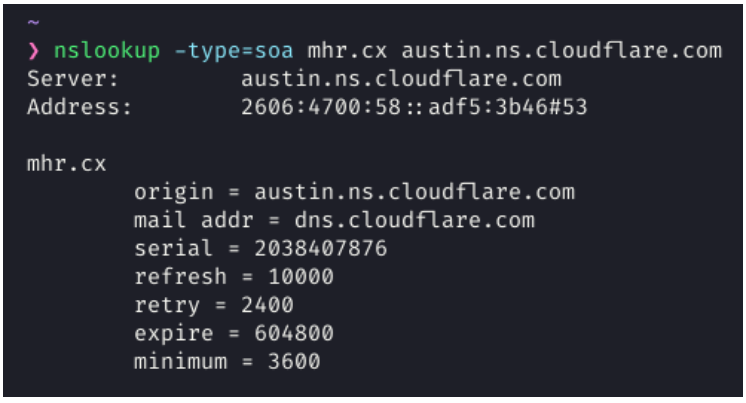
a) `nslookup` tool for most websites tested by me displays an empty “Authoritative answers” block.

My test domain is `mhr.cx`.

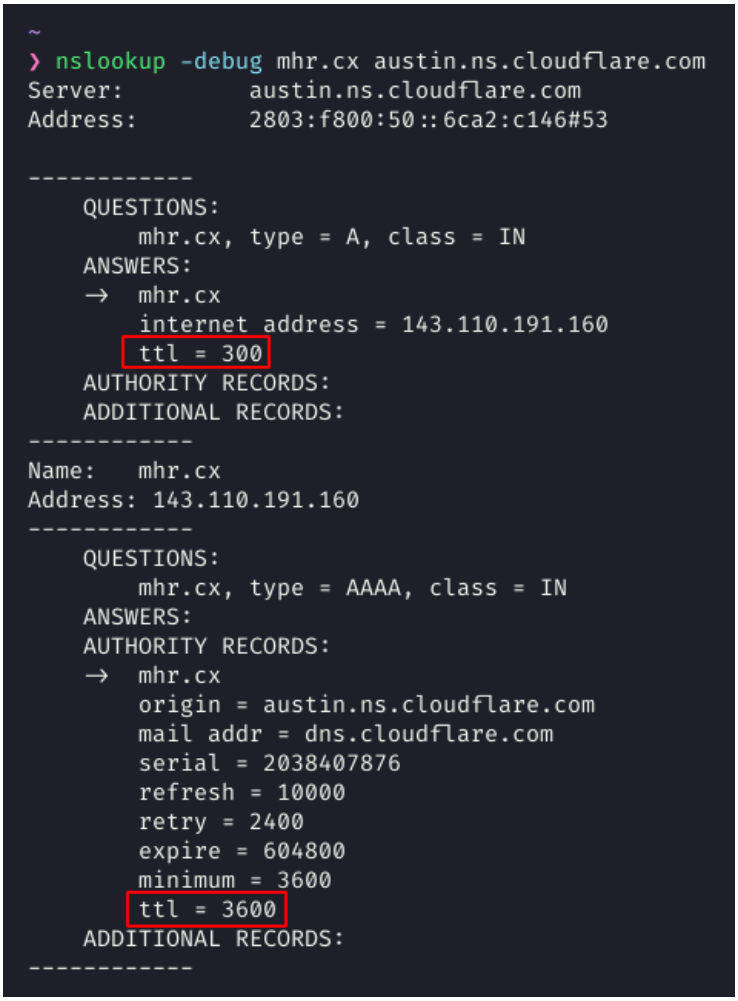
Using the following command, I determine the Start-of-authority record for the domain:
`$ nslookup --type=soa mhr.cx`:



Now that we have the first non-authoritative DNS server, austin.ns.cloudflare.com, we connect to mhr.cx through this DNS server to verify authoritativeness:
`$ nslookup -type=soa mhr.cx austin.ns.cloudflare.com`:



b) To receive the TTL (Time-to-live) information for the domain `mhr.cx`, I queried the domain using `nslookup` through its authoritative DNS server with the `-debug` flag. `$ nslookup -debug mhr.cx austin.ns.cloudflare.com`:



Here we receive verbose information for the `A` and `AAAA` records, that point to the IPv4 and IPv6 addresses of `mhr.cx`.

- For the A record: TTL = 3600 seconds
- For the AAAA record: TTL = 3600 seconds

The TTL field dictates for how long this record can *live* in the cache of the DNS servers before having to query the authoritative name servers again.

Question 6.

a)

```
root@services ~
❯ traceroute www.iiith.ac.in
traceroute to www.iiith.ac.in (196.12.53.50), 30 hops max, 60 byte packets
 1  * * *
 2  10.66.6.190 (10.66.6.190)  2.879 ms 10.66.6.148 (10.66.6.148)  2.860 ms 10.66.6.210 (10.66.6.210)  2.828 ms
 3  138.197.249.14 (138.197.249.14)  2.803 ms  2.780 ms 138.197.249.22 (138.197.249.22)  2.755 ms
 4  * 5.101.108.193 (5.101.108.193)  4.001 ms  3.984 ms
 5  * * *
 6  * * *
 7  * * *
 8  196.12.53.50 (196.12.53.50)  28.000 ms  26.953 ms  27.145 ms
```

- In this traceroute:
  - **8 hops** are made
  - From source to destination, the route goes through **6 intermediate hosts.**
  - For 5 hops, including source, we see asterisks (*), indicating most probably that the host does not support the ICMP protocol. However, traffic does pass.
-

Hop #	Host(s)	Average latency (ms)
2	10.66.6.190, 10.66.6.148	2.855
3	138.197.249.14, 138.197.249.22	2.779
4	5.101.108.193	3.9925
8	196.12.53.50	27.366

b) `ping -c 100 www.iiith.ac.in`:

```
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=91 ttl=58 time=27.1 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=92 ttl=58 time=27.3 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=93 ttl=58 time=27.1 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=94 ttl=58 time=27.1 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=95 ttl=58 time=27.2 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=96 ttl=58 time=27.2 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=97 ttl=58 time=27.2 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=98 ttl=58 time=27.2 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=99 ttl=58 time=27.2 ms
64 bytes from 196.12.53.50 (196.12.53.50): icmp_seq=100 ttl=58 time=32.8 ms

--- www.iiit.ac.in ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99138ms
rtt min/avg/max/mdev = 26.955/29.474/74.849/8.110 ms
```

From the above, the average latency is 29.474ms.

c)

- Sum of ping latencies of intermediate hosts in traceroute = 36.9925 ms
- Average latency in (b) = 29.474ms

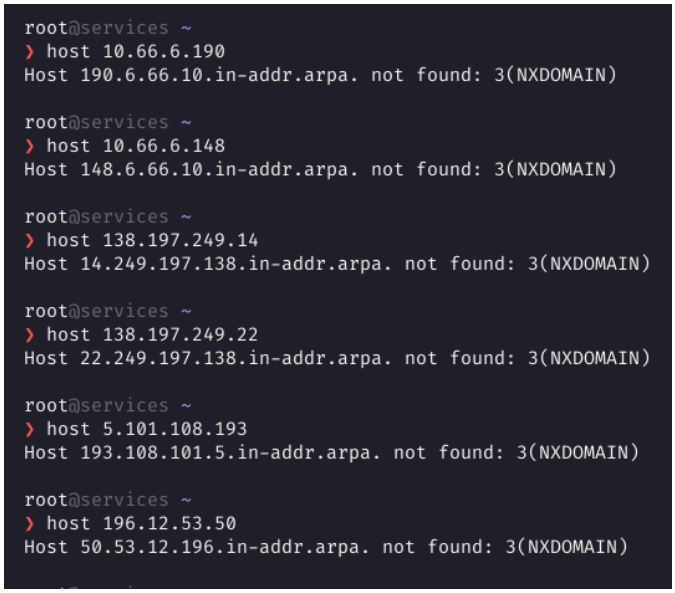
- The average latency of traceroute packets is about 25% greater than `ping`ing directly.
- This is because, even though traceroute also uses the ICMP protocol (same as `ping`) for pinging intermediate hosts, for each intermediate host it probes, by default, a total of three times. It probes multiple times to achieve greater accuracy.
- Thus, due to a greater number of ICMP requests, the average traceroute latency is greater than the average ping latency.

d)

- The maximum ping latency of intermediate hosts is of Hop #8 = 27.366ms
- The average latency in (b) = 29.474ms
- These two are almost equal because Hop #8 is the host for `www.iiith.ac.in`, and in (b), we are directly pinging the host for `www.iiith.ac.in` without multiple pings to intermediate hosts. Since ICMP requests are being sent to the same hosts, the latency is very similar.

e) I attempted to get the hostname of the IP address by performing reverse DNS lookups using the `dig` and `host` commands. The IP address do not have their PTR (pointer) records that associate them to a hostname in any registry/registrar's tables. Thus, their hostname cannot be looked up.

IP Address	**Hostname**
10.66.6.190	None found
10.66.6.148	None found
138.197.249.14	None found
138.197.249.22	None found
5.101.108.193	None found
196.12.53.50	None found



Question 7.

To make the ping command fail for 127.0.0.1 (the loopback address), we can disable the loopback interface altogether. Doing this, `ping` won't be able to send a successful request to the loopback address, 127.0.0.1, and so this will cause 100% packet loss.

To disable the loopback interface: `$ sudo ifconfig lo down`

Then when we do `$ ping 127.0.0.1`:

```
~
❯ sudo ifconfig lo down

~
❯ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
^C
--- 127.0.0.1 ping statistics ---
14 packets transmitted, 0 received, 100% packet loss, time 13323ms
```