**CSE 232: Assignment 2**

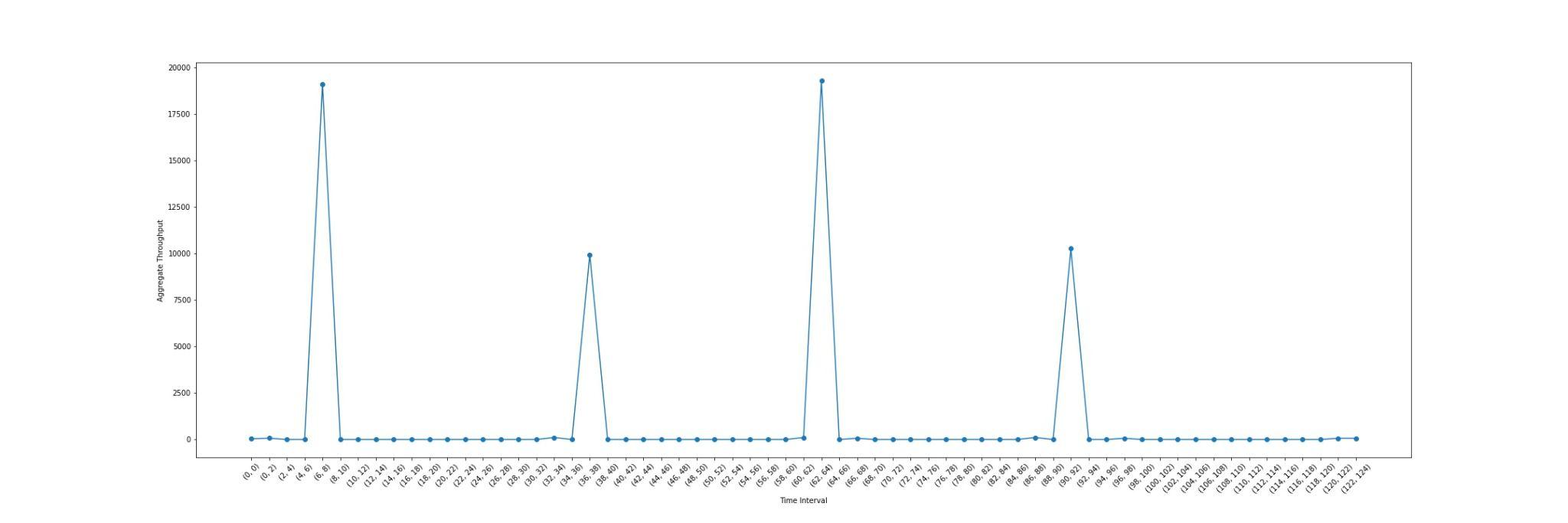
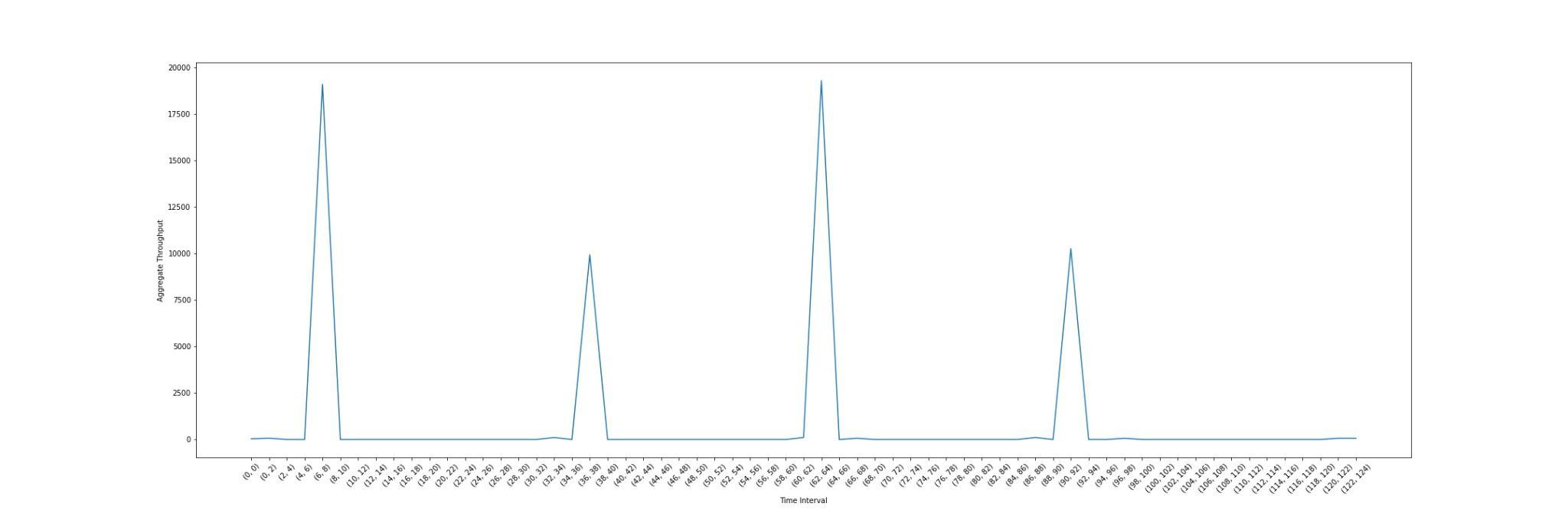
**Kesar Shrivastava**

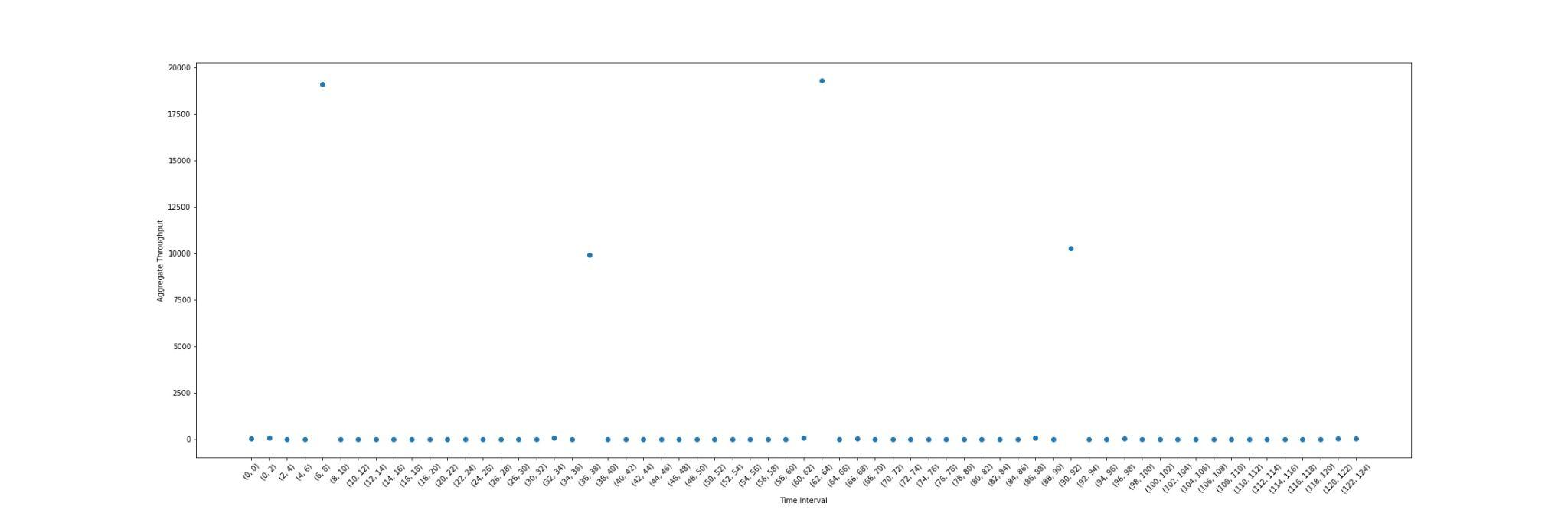
**2019051**

**Q1**

**Packet capture:** The packets are captured from the loopback interface since both the client and the server are running on the same machine. From the display filter of Wireshark, the packets are filtered using tcp.port (according to the port number in the program). It is then exported into csv format and using a python script the plot is made.

**Analysis:** Since there are three to four clients running in parallel which communicate with the server we see spikes when the packets are sent in between both. There are also some bumps in between due to input output.

**The plots:** They are provided as pdf in the zip file too.

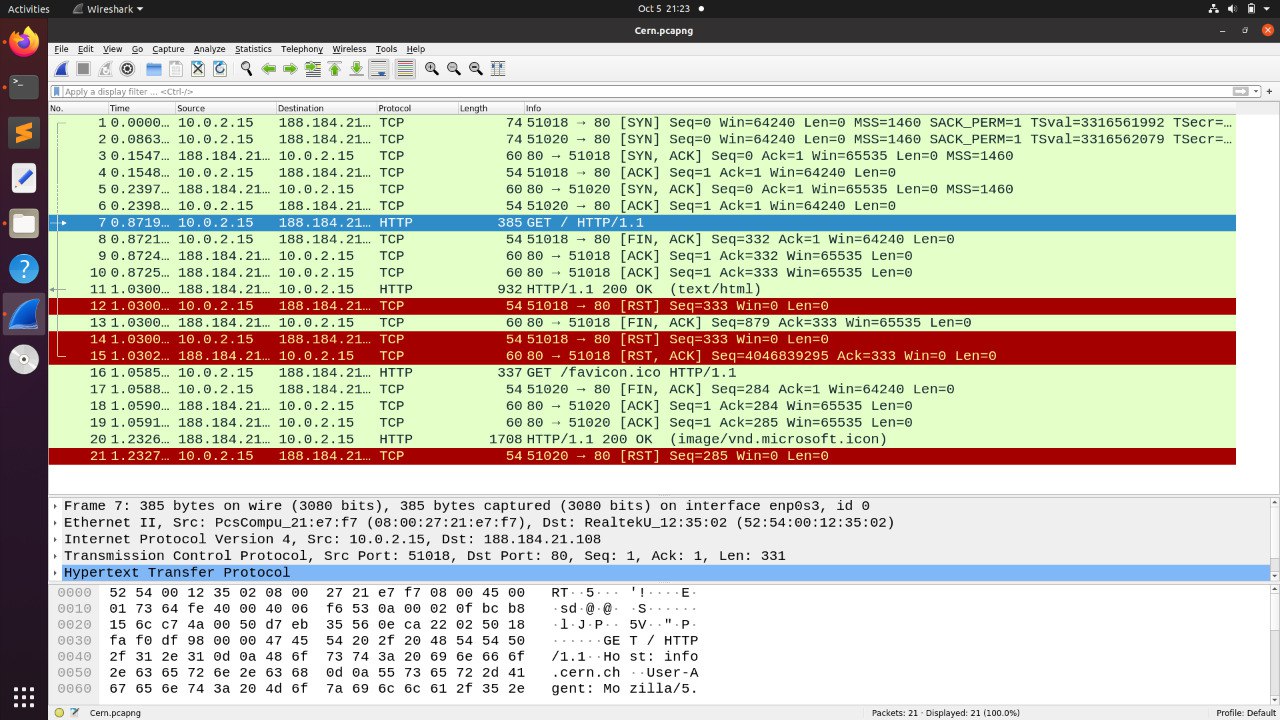


**Q2**

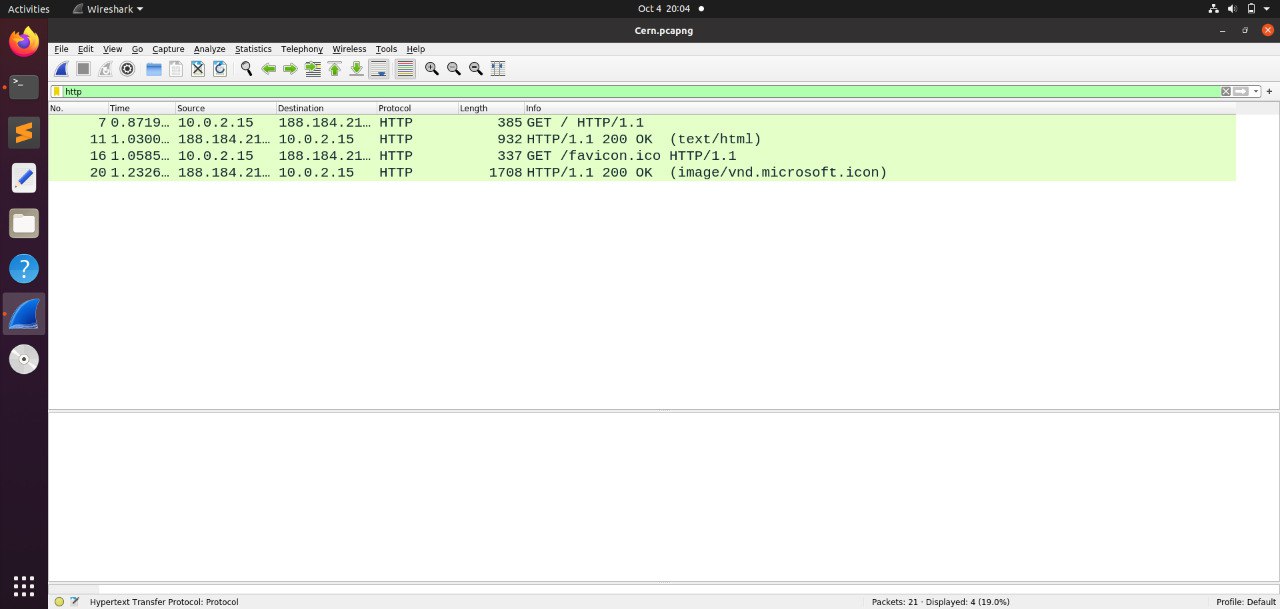
Capture filter: host info.cern.ch

The packets are captured from the ethernet interface.

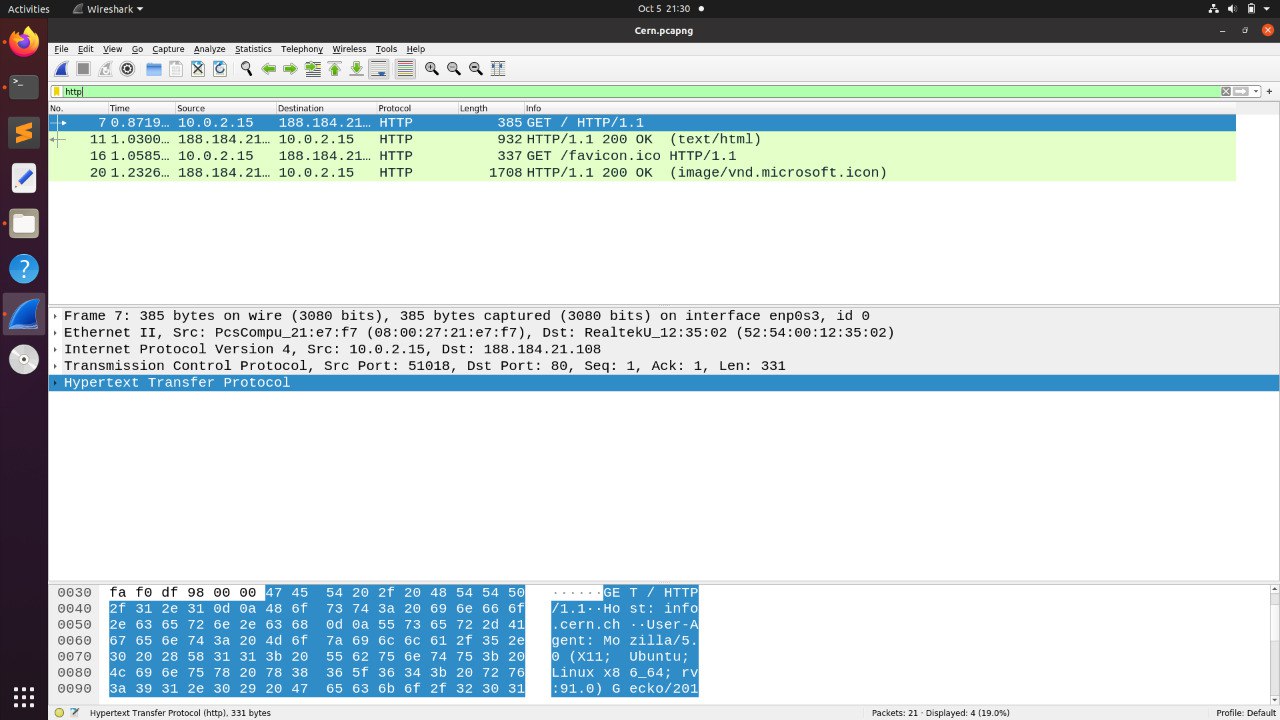
**The complete packet capture:**

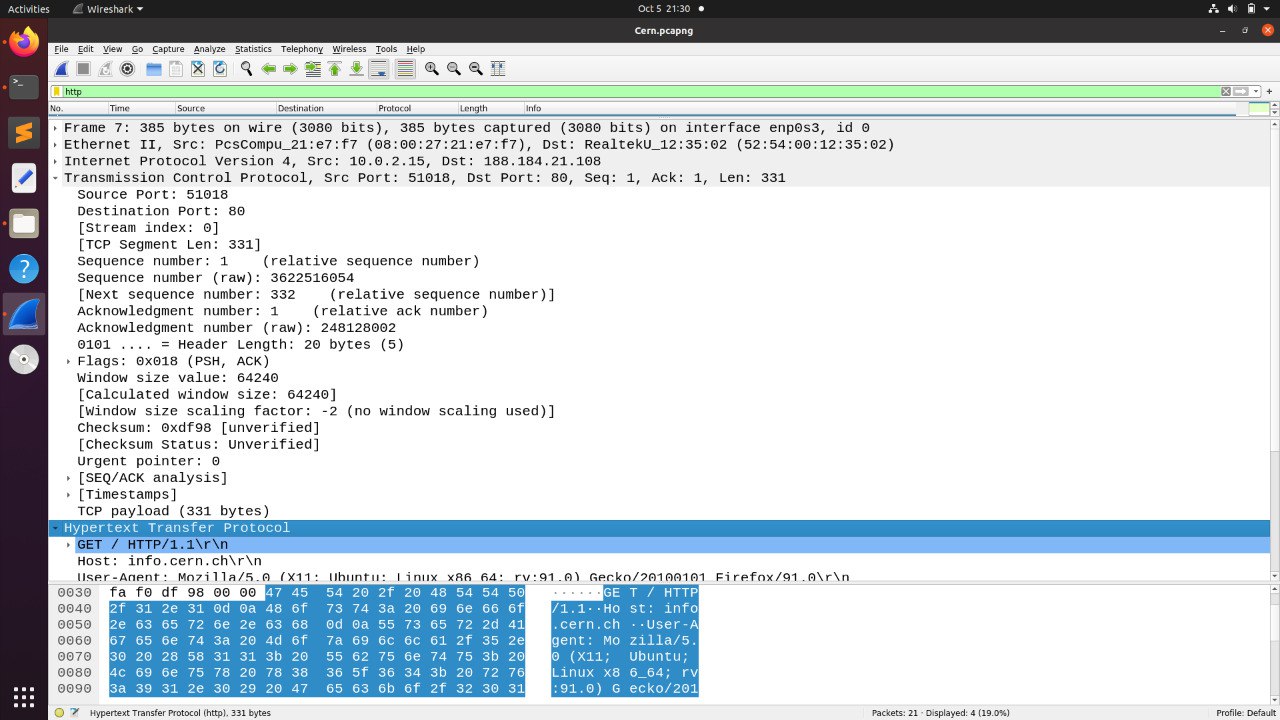


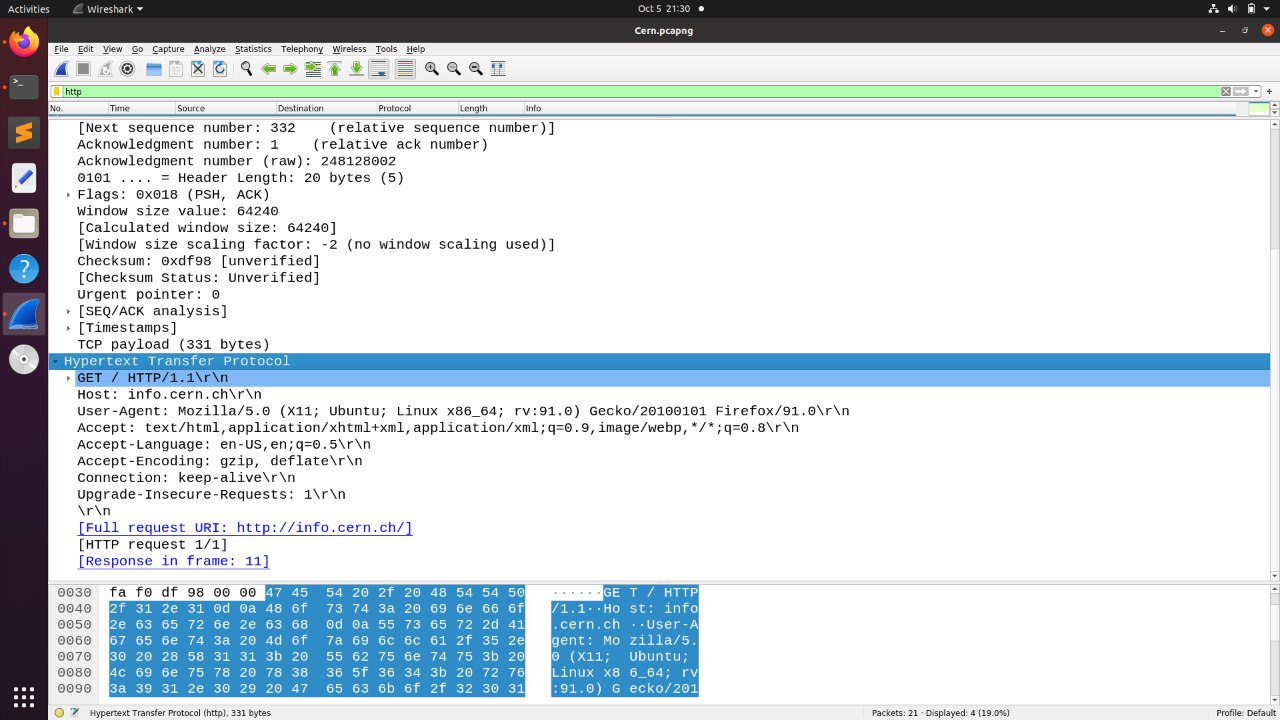
HTTP filter is applied:

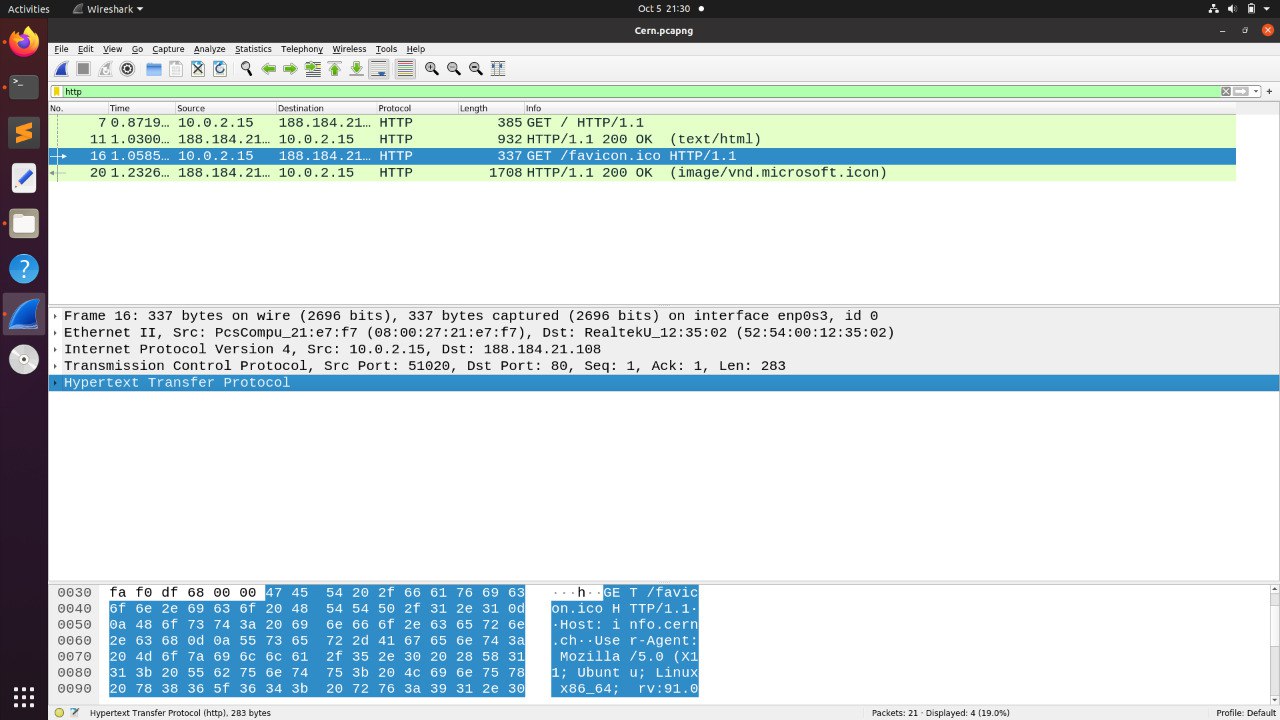


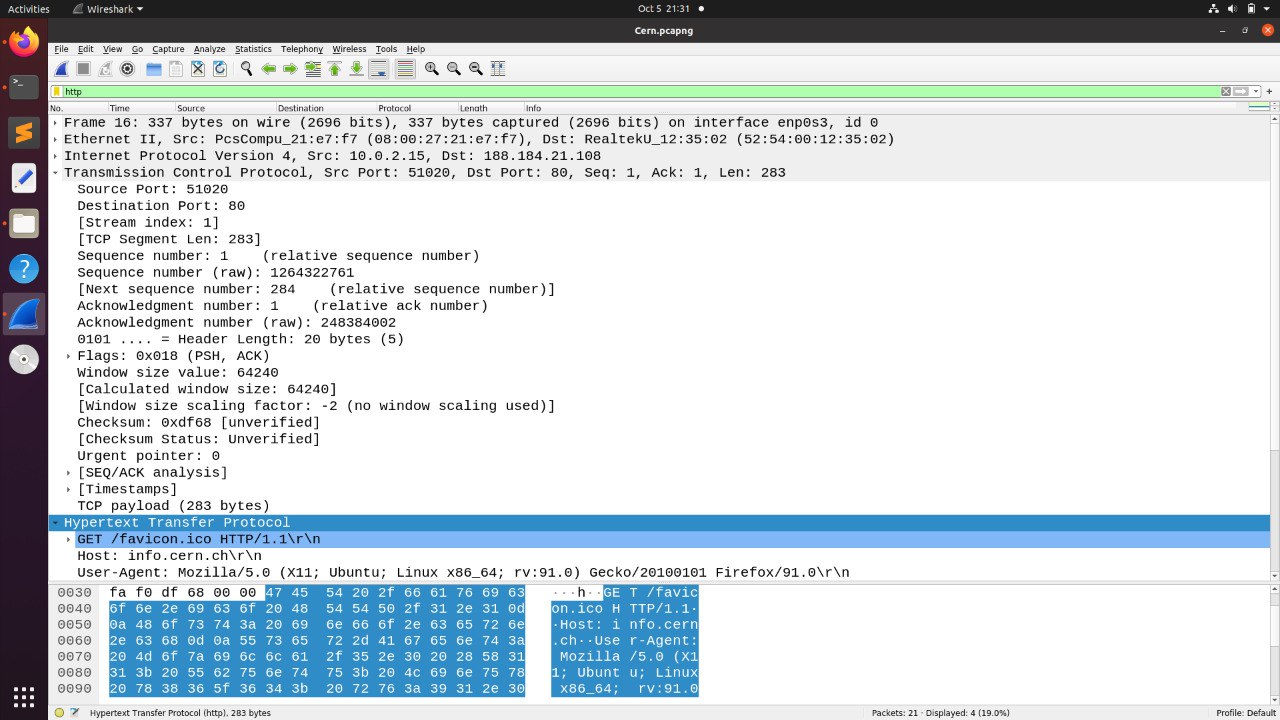
**The HTTP request messages:**

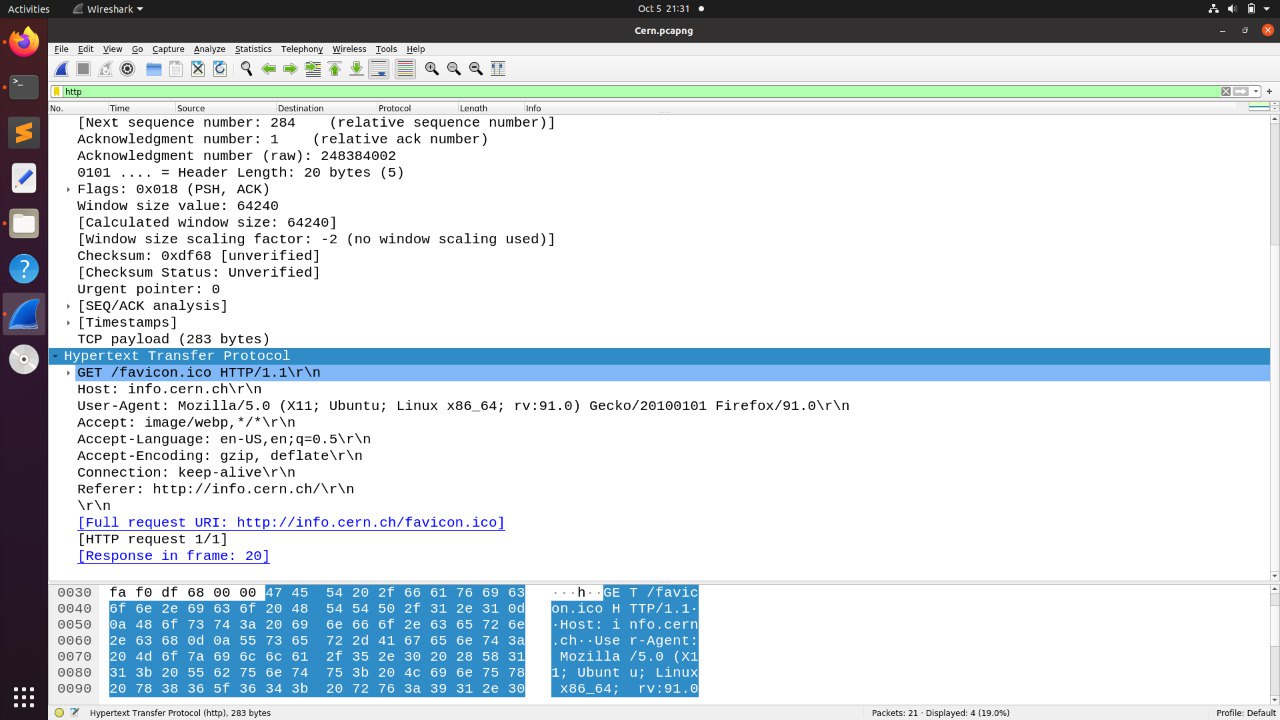




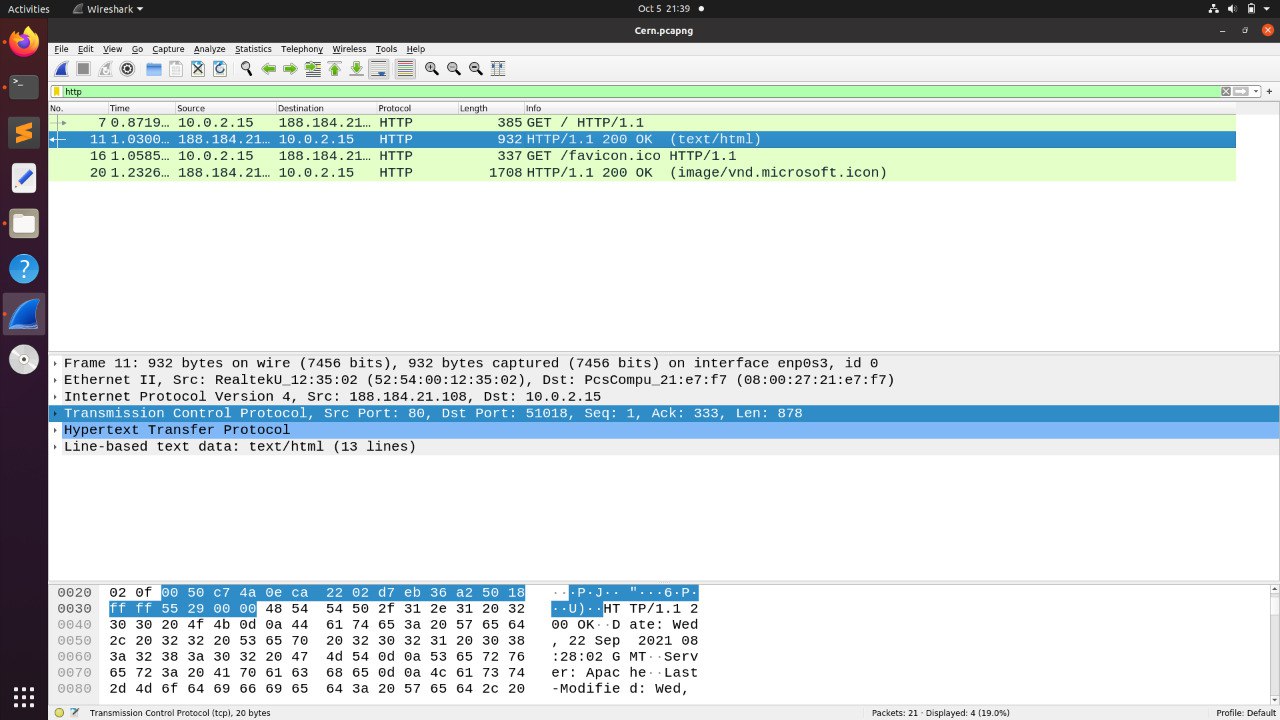


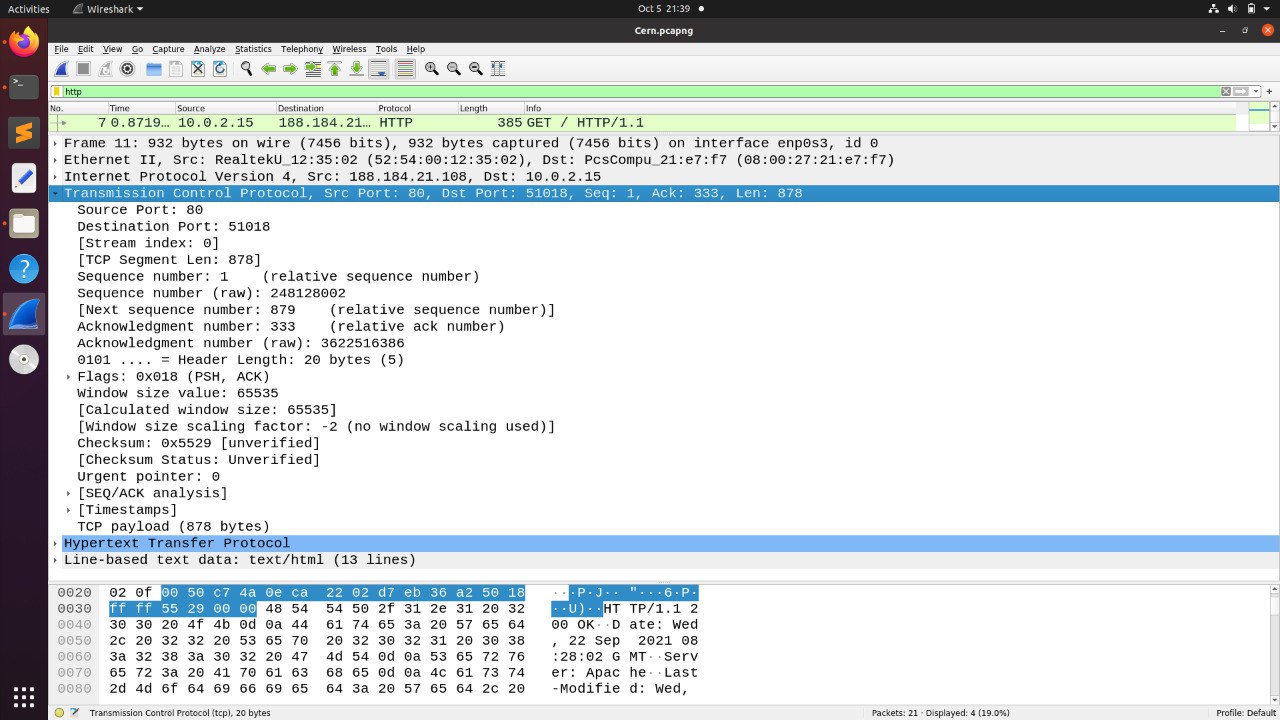


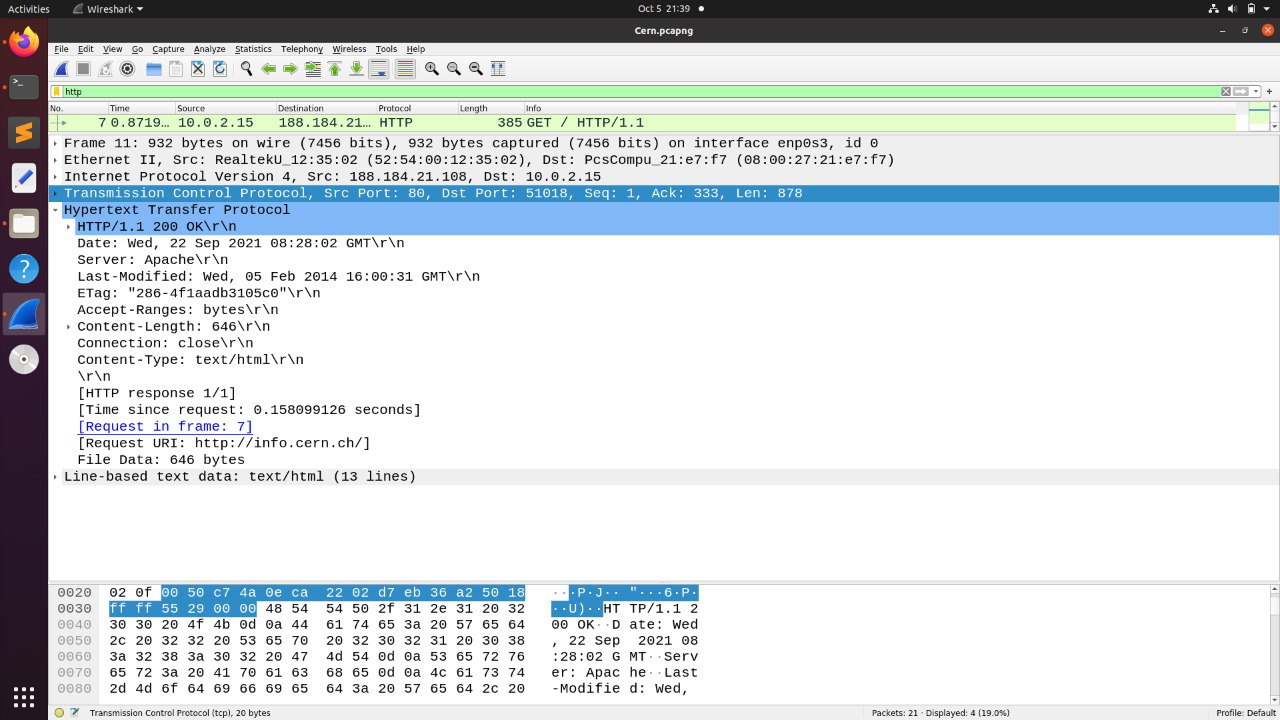


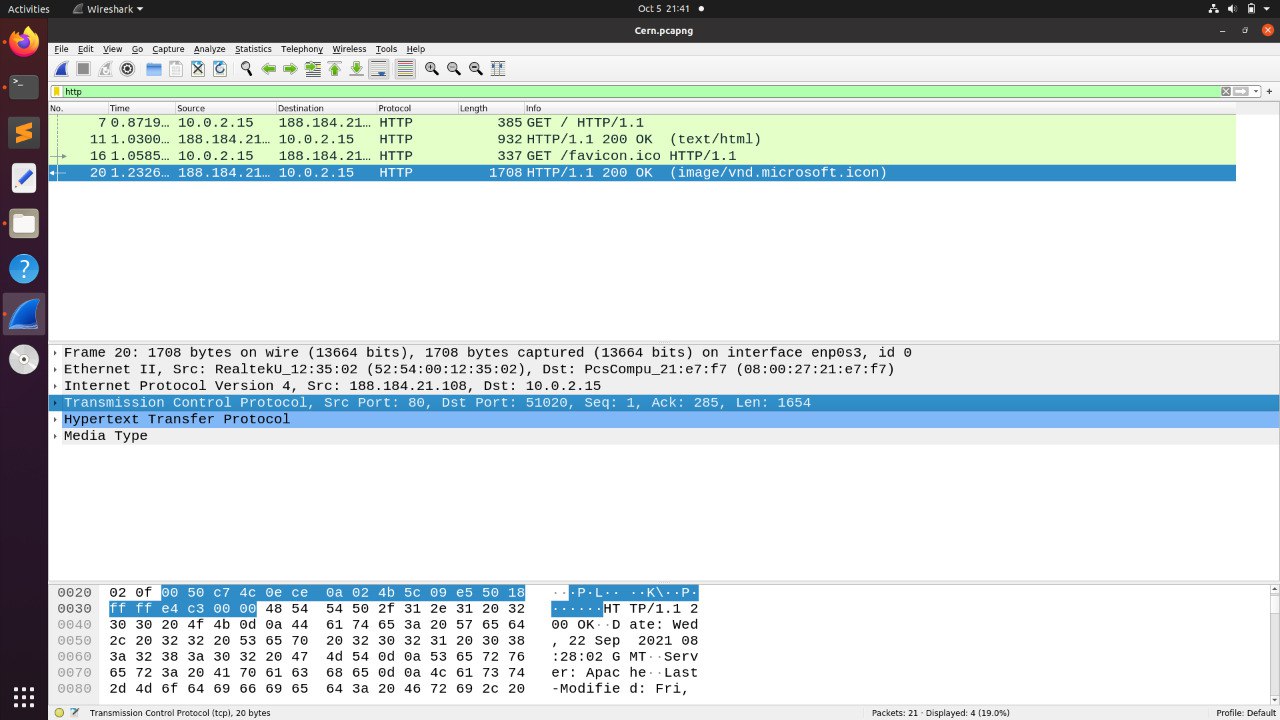


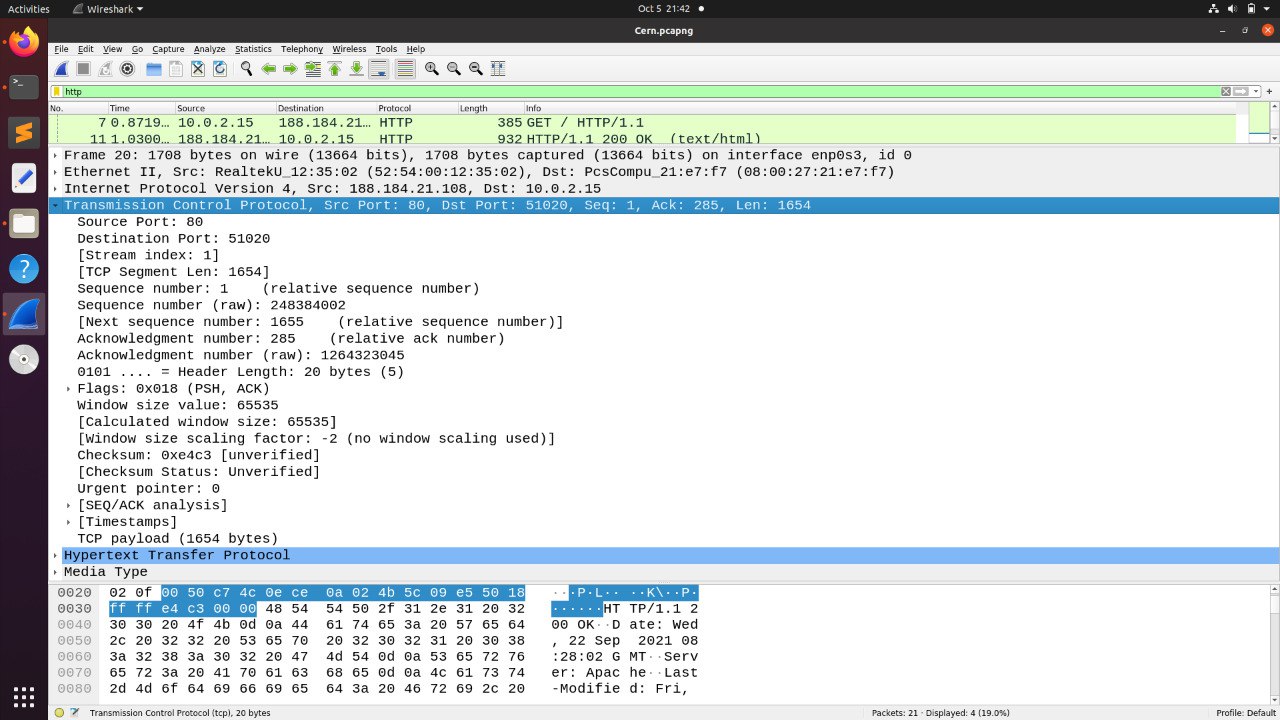
**The HTTP response messages:**

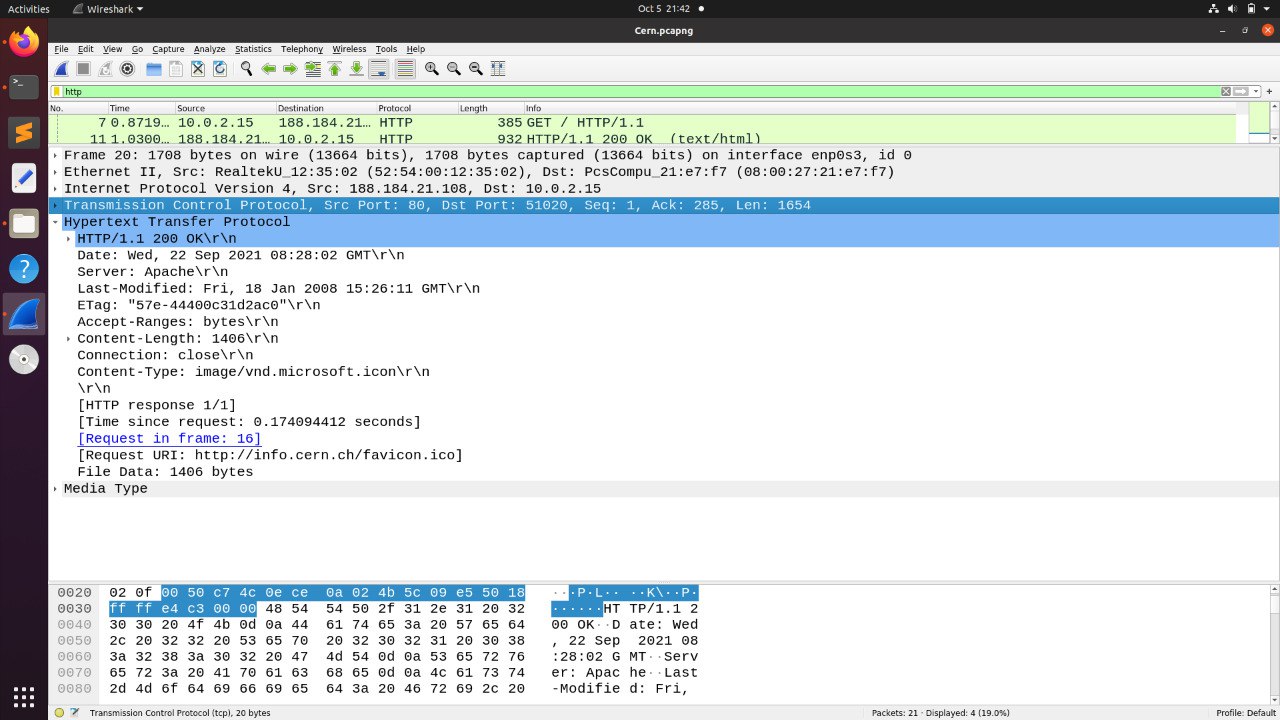












Packet by frame number:

**Frame 7:**

**HTTP packet type:** Request

**HTTP Request type:** GET

**User Agent Type**: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n

**HTTP request packet’s URL:** <http://info.cern.ch/>

**Frame 11:**

**HTTP packet type:** Response

**HTTP response code:** 200

**HTTP response description:** OK

**Name and version of the webserver:** Apache\r\n

**Frame 16:**

**HTTP packet type:** Request

**HTTP Request type:** GET

**User Agent Type**: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n

**HTTP request packet’s URL:** <http://info.cern.ch/favicon.ico>

**Frame 20:**

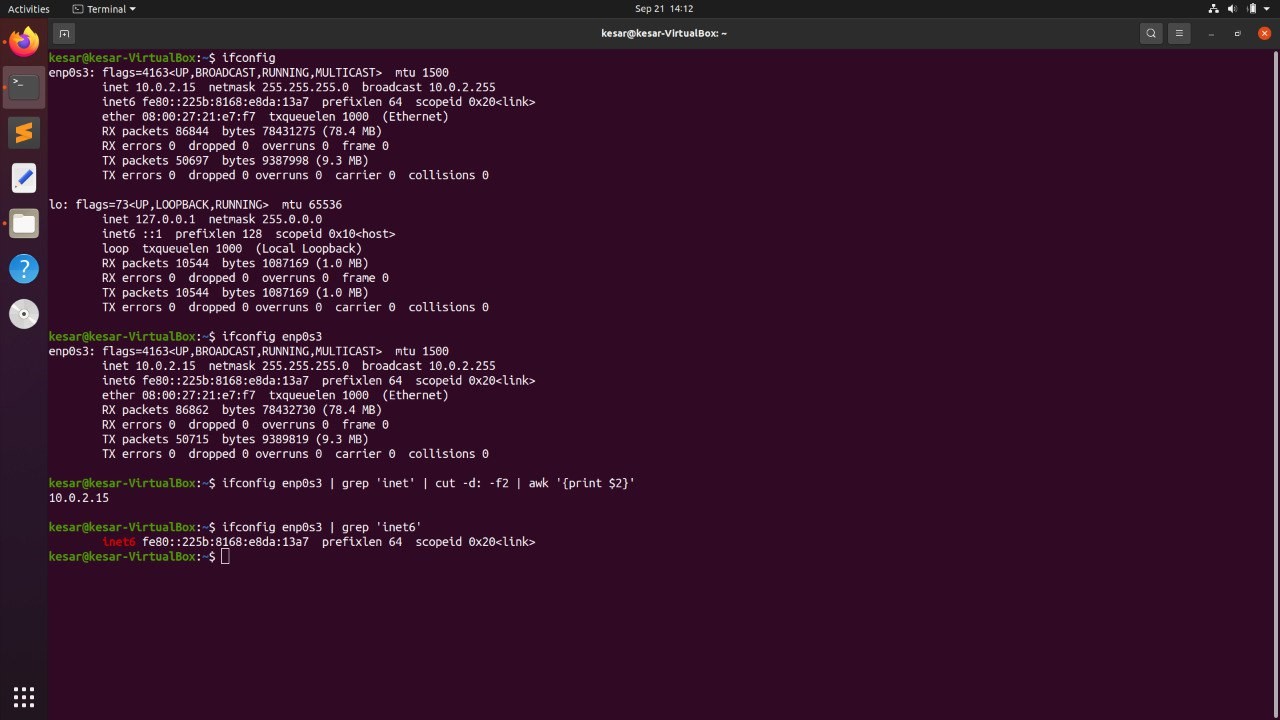
**HTTP packet type:** Response

**HTTP response code:** 200

**HTTP response description:** OK

**Name and version of the webserver:** Apache\r\n

**Q3 (a)**



ifconfig command is used to configure the kernel-resident network interfaces. It shows two interfaces: enp0s3 and lo. enp0s3 represents the active network interface. The inet and inet6 represent IPv4 and IPv6 addressing respectively.

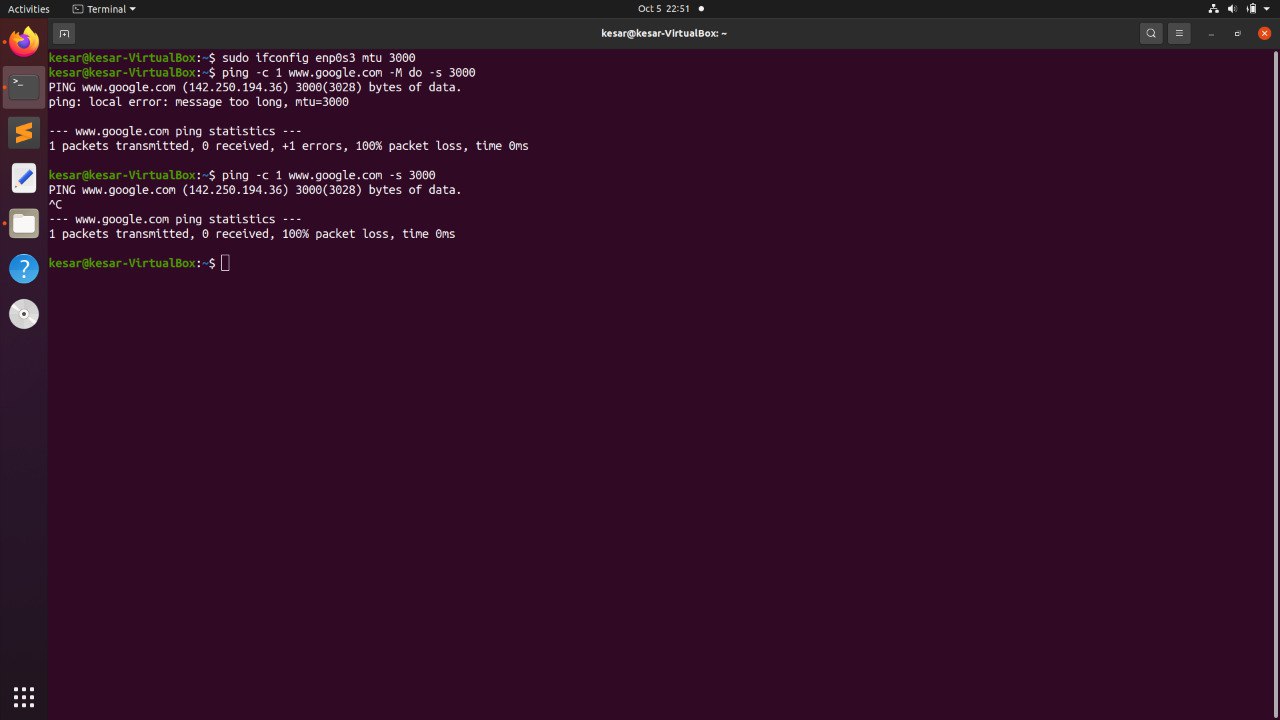
**Q3 (b)**



The IP address figured out in (a) is different from what is shown on the given webpage. This happens due to the categorization of the IP addresses into public and private.

A private IP address is used to communicate within the same network while the public IP address which used to communicate outside the network. Ifconfig gives us the private IP address.

**Q4 (a)**



MTU(Maximum Transmission Unit) is the largest packet or frame size specified in bytes that can be sent over a network connection.

MTU 3000 means that we are sending a packet of size 3000 bytes over the network.

Command to test whether we can send a packet of size 3000 bytes to [www.google.com](https://www.google.com/):

ping [www.google.com](https://www.google.com/) -M do -s 3000

First we change the mtu of the ethernet interface using the ifconfig command. Ping command sends ICMP echo request to network hosts. It takes URL or the ip address of the host and sends it a data packet. Using different options we can modify the packet size to be sent.

-s: specifies the number of data bytes to be sent.

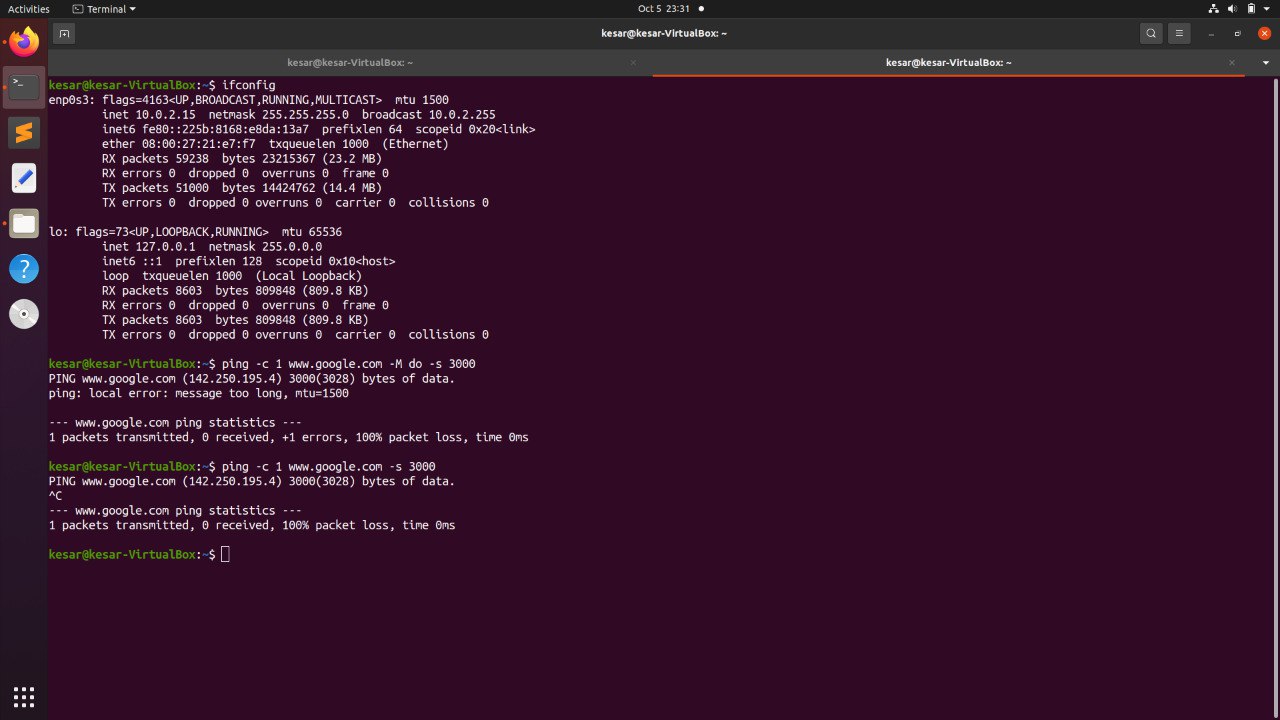
-M do: prohibits the fragmentation of the data packet.

-c: specifies the number of packets to be sent.

Since the packet cannot be fragmented any router that receives the packet will analyze the header and check for the Don't Fragment flag. Since the flag is on and the packet exceeds the MTU (due to the addition of 28 bytes extra header), the router then drops the packet instead of fragmenting it.

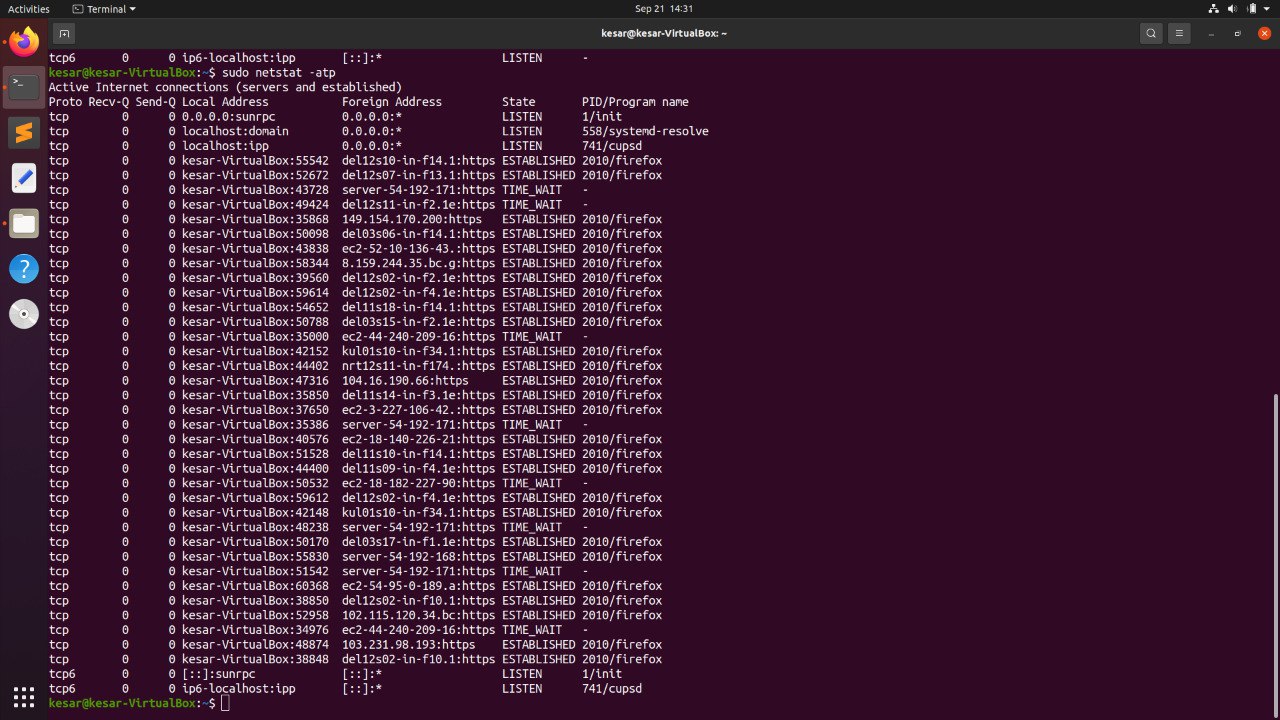
Also, a command is run when we are not mentioning the -M do option. But after this too we see a 100% packet loss. This is because even after fragmentation the packet size is too big to be transmitted.

The maximum transmission unit is 552 bytes. 3000 > 552, and hence we see 100% packet loss.



In the above screenshot we are not changing the mtu and the reason why we see packet loss is similar to the first case: the packet size is greater than the mtu.

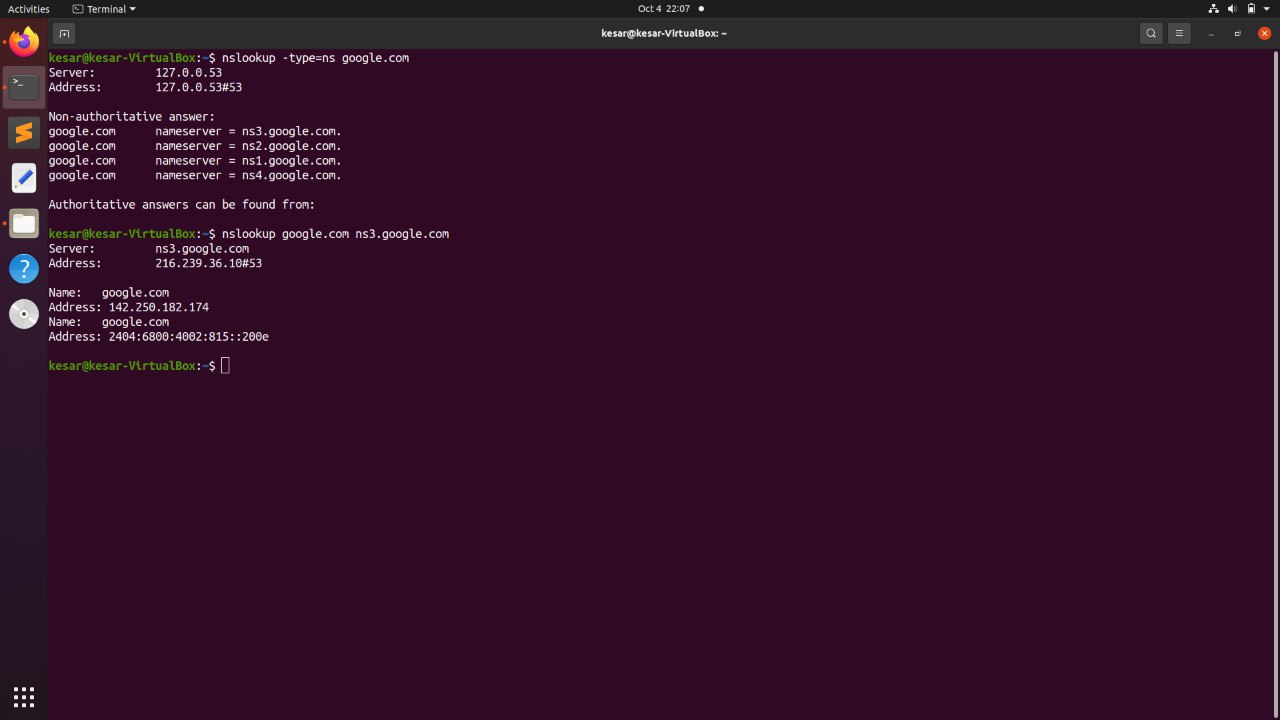
**Q4 (b)**

****

netstat command prints information about the Linux networking subsystem. The required command is 'netstat -atp'.

-a lists all the currently active connections. -t filters out all the tcp connections and -p shows the pid of the program to which each socket belongs. We use sudo so that we get the root privileges and all the connections are shown.

**Q5 (a)**



nslookup is the command that queries the name servers interactively. Using this command we can get the authoritative result. The option -type=ns helps get all name servers that are authoritative for that domain.

Thus, in the screenshot attached 'nslookup -type=ns [google.com](https://google.com/)' gives all its name servers.

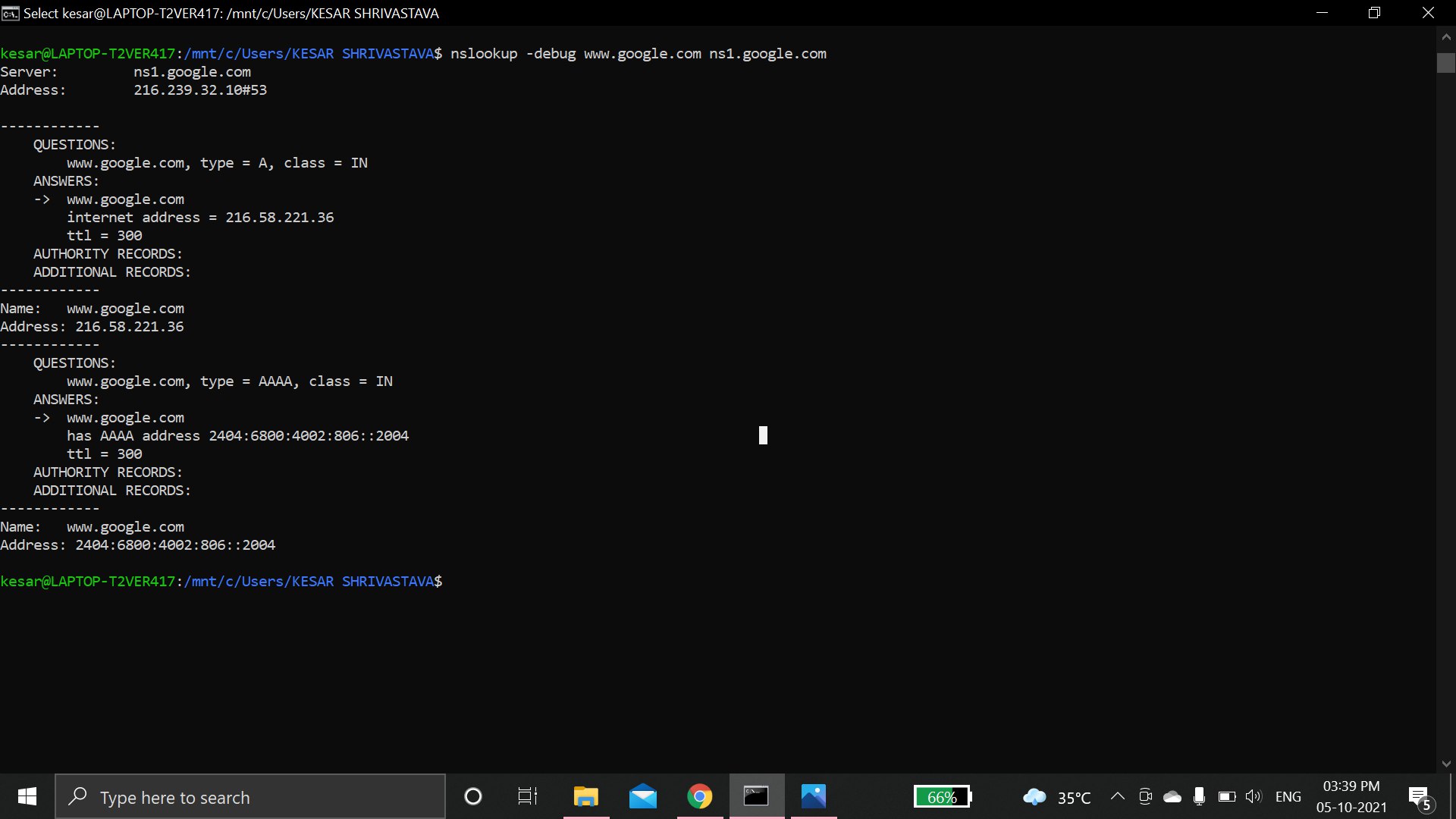
By default, nslookup queries the same DNS the system is configured to use for all network operations. We can specify a custom DNS to query. The command 'nslookup [google.com](https://google.com/) [ns3.google.com](https://ns3.google.com/)' provides us with the authoritative answer to our previous query of [google.com](https://google.com/)

**Q5 (b)**



As seen from the screenshot, the TTL(time to live) is **40 seconds** so the entry would expire by 40 Seconds. Time to live (TTL) refers to the amount of time that a packet is set to exist inside a network before being discarded by a router.

Below screenshot shows TTL for www.google.com using authoritative DNS server



TTL: 300 seconds

**Q6 (a)**



There are 8 intermediate hosts. 7 intermediate hosts if we ignore the Asterix host.

The IP address and average latency to each intermediate hosts is:

1. 192.168.1.1 (2+2+1)/3 ms = 1.67 ms

2. 182.78.219.41 (5+6+5)/3 ms = 5.33 ms

3. 116.119.61.117 (48+46+47)/3 ms = 47 ms

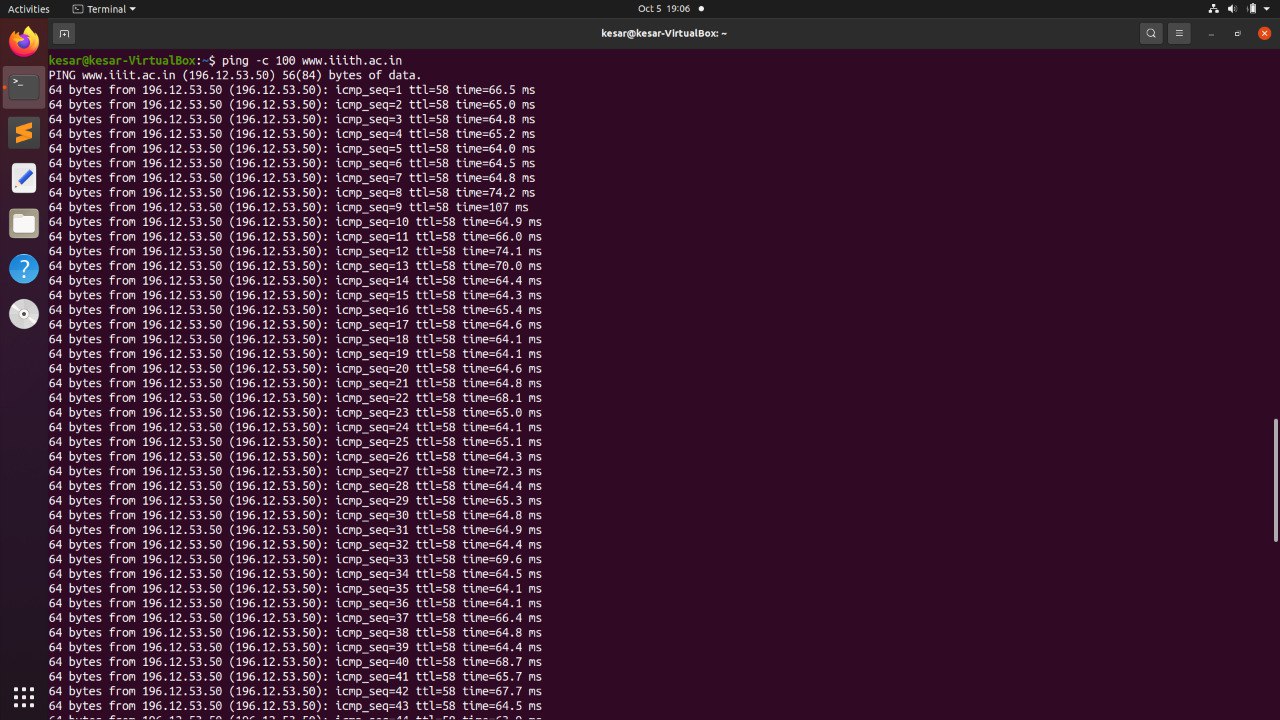
4. 49.44.220.188 (46+53+46)/3 ms = 48.33 ms

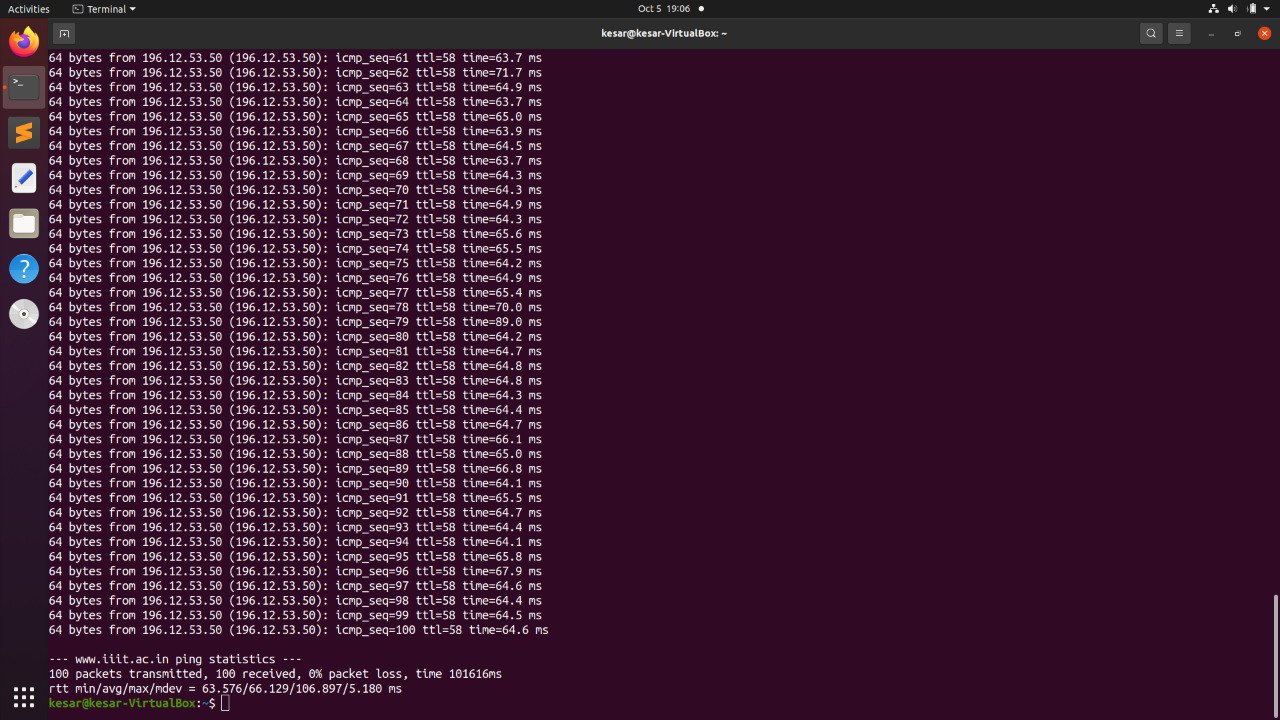
6. 115.242.184.26 (56+57+56)/3 ms = 56.33 ms

7. 196.12.34.76 (62+62+62)/3 ms = 62 ms

8. 196.12.53.50 (63+63+65)/3 ms = 63.67 ms

**Q6 (b)**





**The average latency is 66.129 ms**

**Q6 (c)**

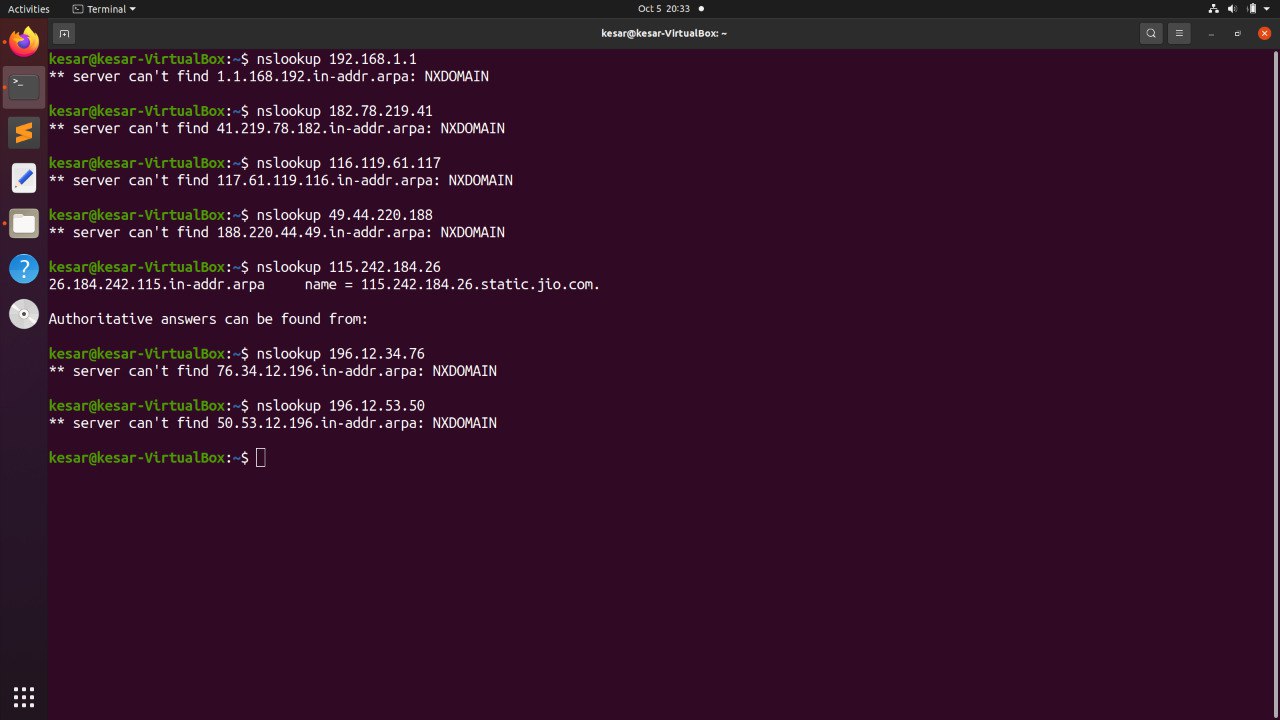
After adding up all the average ping latency of the intermediate hosts, we get: 284.33 ms = (1.67+5.33+47+48.33+56.33+62+63.67) ms. This is greater than the (b) that is 66.129 ms.

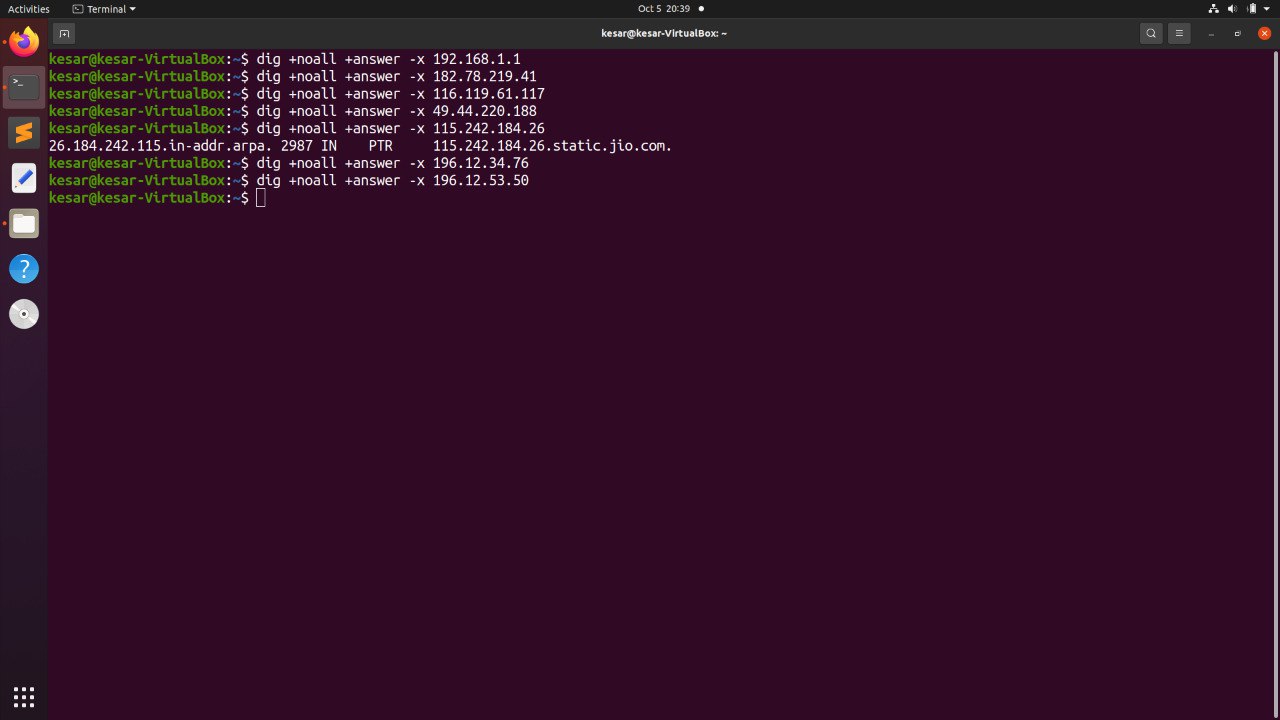
We see a greater number here because during tracert we wait for the timeout response of each of the intermediate host and the average latency we get is the round trip time between our machine and the intermediate host as seen in the tracert output. It thus doubles up the time. However, ping is just forwarded where we just forward a single packet and hence it is less than the addition.

**Q6 (d)**

The maximum average latency in (a) part is from the 8th host that is 63.67ms is comparable to (b) part (66.129 ms). The variation in the latency is due to the fact that packet traverses through different nodes for different times. The variation arises because the traffic in the network varies from time to time. This matches to the maximum latency in the tracert output because the average latency for ping will be dominated by the slowest node in between.

**Q6 (e)**





Nslookup and dig both can be used to perform the reverse DNS lookups. dig -x is used to perform reverse DNS lookup for dig command.

The hostname in the screenshots comes only for one intermediate IP address that is 115.242.184.26.

In nslookup screenshot the name field gives the hostname of this IP address and in the dig screenshot too 115.242.184.26.static.jio.com is the hostname.

There are not any aliases.

**Q7**

Ifconfig is a command used to configure kernel-resident network interfaces. [127.0.0.1](https://127.0.0.1/) is the IP address of one such interface called loopback. It is a virtual network device used to access network services locally.

Ifconfig has an option down that is used to deactivate the driver for the given interface.

Hence the command: sudo ifconfig lo down would deactivate loopback making it unreachable. Thus, the ping command on this IP address would fail with 100% packet loss.

sudo needs to be added as it lets us have the root user privileges. Without this, the loopback would not get deactivated.

