WEEK1 ASSIGNMENT

NAME: LIKHITH R

SRN: PES1UG20CS659

SECTION: K

ROLL NUMBER: 12

SEMESTER: 4th

Task 1: Linux Interface Configuration (ifconfig / IP command)

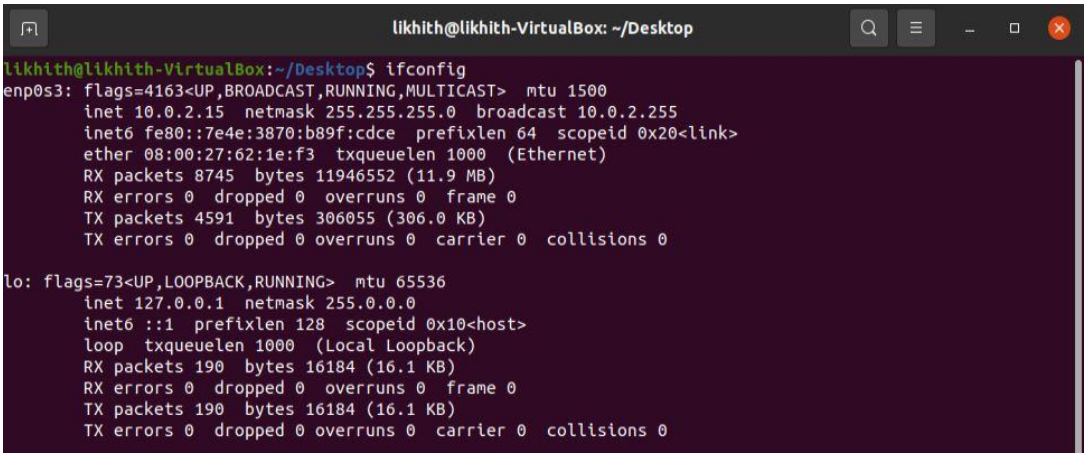
Step 1: To display status of all active network interfaces.

ifconfig (or) **ip addr show**

Analyze and fill the following table:

ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address
enp0s3	10.0.2.15 fe80::7e4e:3870:b89f:cdce	08:00:27:62:1e:f3
lo	127.0.0.1/: :1	00:00:00:00:00:00



Step 2: To assign an IP address to an interface, use the following command.

sudo ifconfig interface_name 10.0.your_section.your_sno netmask 255.255.255.0 (or)

sudo ip addr add 10.0.your_section.your_sno /24 dev interface_name

```
likhith@likhith-VirtualBox:~/Desktop$ sudo ifconfig enp0s3 10.0.11.12 netmask 255.255.255.0
[sudo] password for likhith:
likhith@likhith-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.11.12  netmask 255.255.255.0  broadcast 10.0.11.255
        inet6 fe80::7e4e:3870:b89f:cdce  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:62:1e:f3  txqueuelen 1000  (Ethernet)
        RX packets 8754  bytes 11947322 (11.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4612  bytes 308773 (308.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 196  bytes 16648 (16.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 196  bytes 16648 (16.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 3: To activate / deactivate a network interface, type.

sudo ifconfig interface_name down

```
likhith@likhith-VirtualBox:~/Desktop$ sudo ifconfig enp0s3 down
likhith@likhith-VirtualBox:~/Desktop$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 202  bytes 17112 (17.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 202  bytes 17112 (17.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

sudo ifconfig interface_name up

```
likhith@likhith-VirtualBox:~/Desktop$ sudo ifconfig enp0s3 up
likhith@likhith-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::7e4e:3870:b89f:cdce  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:62:1e:f3  txqueuelen 1000  (Ethernet)
        RX packets 8773  bytes 11949569 (11.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4660  bytes 314540 (314.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 226  bytes 19012 (19.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 226  bytes 19012 (19.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 4: To show the current neighbor table in kernel, type

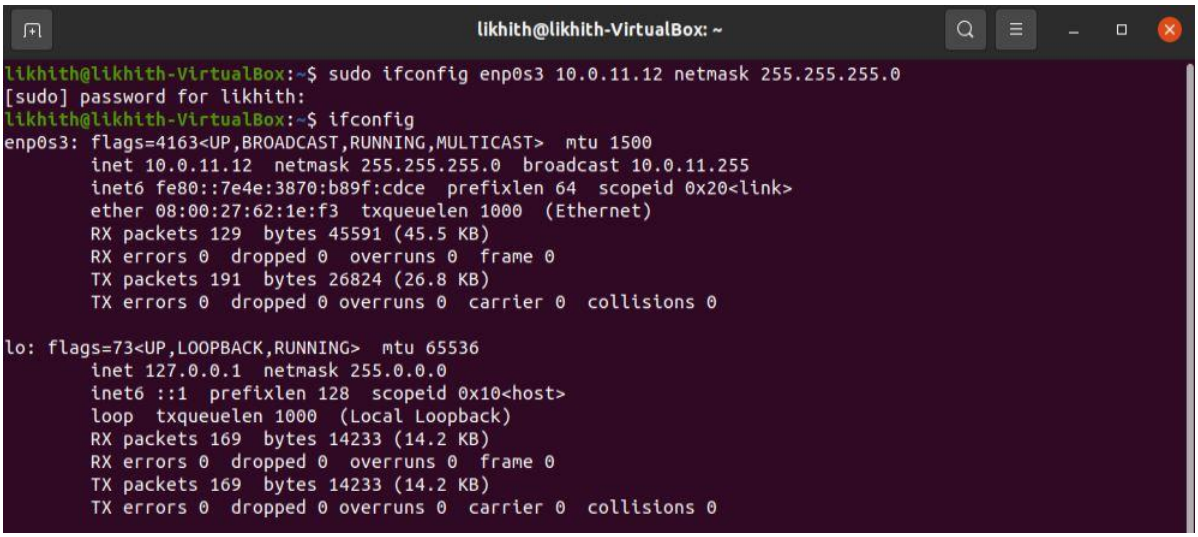
ip neigh

```
likhith@likhith-VirtualBox:~/Desktop$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 REACHABLE
likhith@likhith-VirtualBox:~/Desktop$
```

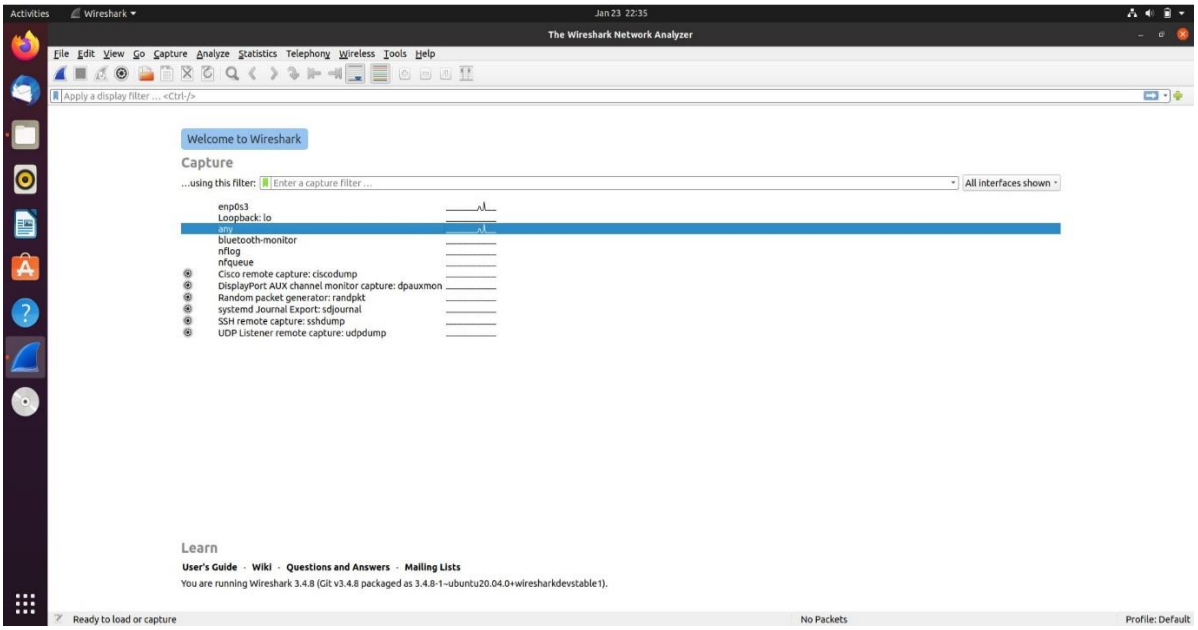
Task 2: Ping PDU (Packet Data Units or Packets) Capture

Step 1: Assign an IP address to the system (Host).

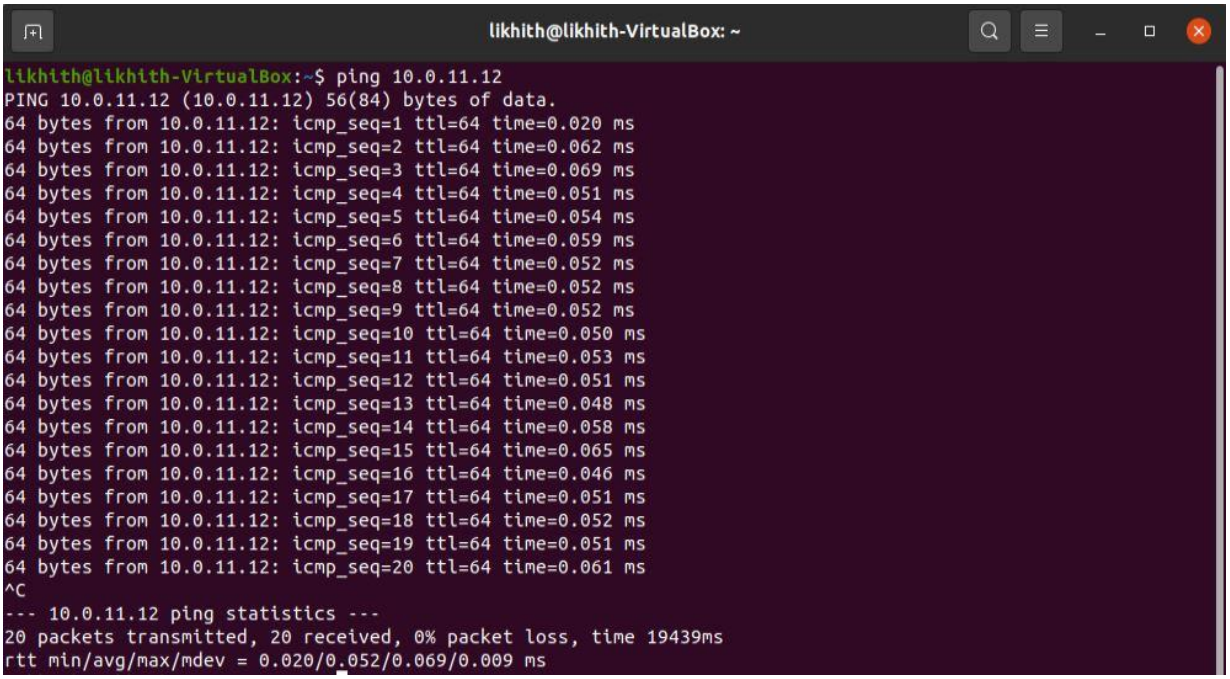
Note: IP address of your system should be 10.0.your_section.your_sno.



Step 2: Launch Wireshark and select ‘any’ interface



Step 3: In terminal, type **ping 10.0.your_section.your_sno**



```
likhith@likhith-VirtualBox: ~
likhith@likhith-VirtualBox:~$ ping 10.0.11.12
PING 10.0.11.12 (10.0.11.12) 56(84) bytes of data.
64 bytes from 10.0.11.12: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 10.0.11.12: icmp_seq=2 ttl=64 time=0.062 ms
64 bytes from 10.0.11.12: icmp_seq=3 ttl=64 time=0.069 ms
64 bytes from 10.0.11.12: icmp_seq=4 ttl=64 time=0.051 ms
64 bytes from 10.0.11.12: icmp_seq=5 ttl=64 time=0.054 ms
64 bytes from 10.0.11.12: icmp_seq=6 ttl=64 time=0.059 ms
64 bytes from 10.0.11.12: icmp_seq=7 ttl=64 time=0.052 ms
64 bytes from 10.0.11.12: icmp_seq=8 ttl=64 time=0.052 ms
64 bytes from 10.0.11.12: icmp_seq=9 ttl=64 time=0.052 ms
64 bytes from 10.0.11.12: icmp_seq=10 ttl=64 time=0.050 ms
64 bytes from 10.0.11.12: icmp_seq=11 ttl=64 time=0.053 ms
64 bytes from 10.0.11.12: icmp_seq=12 ttl=64 time=0.051 ms
64 bytes from 10.0.11.12: icmp_seq=13 ttl=64 time=0.048 ms
64 bytes from 10.0.11.12: icmp_seq=14 ttl=64 time=0.058 ms
64 bytes from 10.0.11.12: icmp_seq=15 ttl=64 time=0.065 ms
64 bytes from 10.0.11.12: icmp_seq=16 ttl=64 time=0.046 ms
64 bytes from 10.0.11.12: icmp_seq=17 ttl=64 time=0.051 ms
64 bytes from 10.0.11.12: icmp_seq=18 ttl=64 time=0.052 ms
64 bytes from 10.0.11.12: icmp_seq=19 ttl=64 time=0.051 ms
64 bytes from 10.0.11.12: icmp_seq=20 ttl=64 time=0.061 ms
^C
--- 10.0.11.12 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19439ms
rtt min/avg/max/mdev = 0.020/0.052/0.069/0.009 ms
```

Observations to be made

Step 4: Analyze the following in Terminal

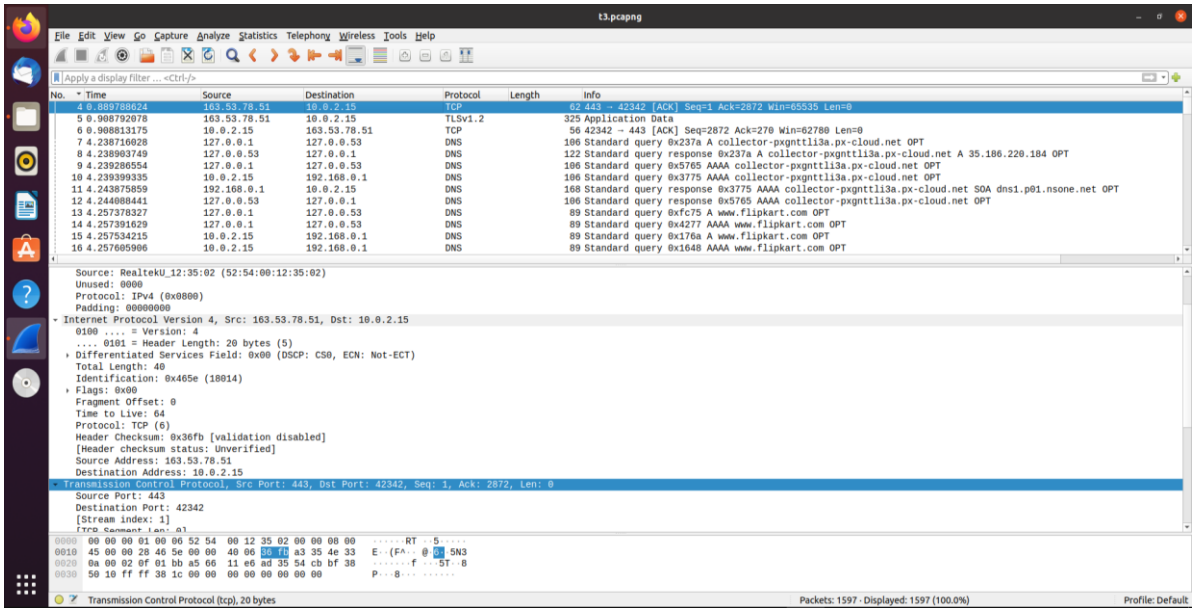
- TTL
- Protocol used by ping
- Time

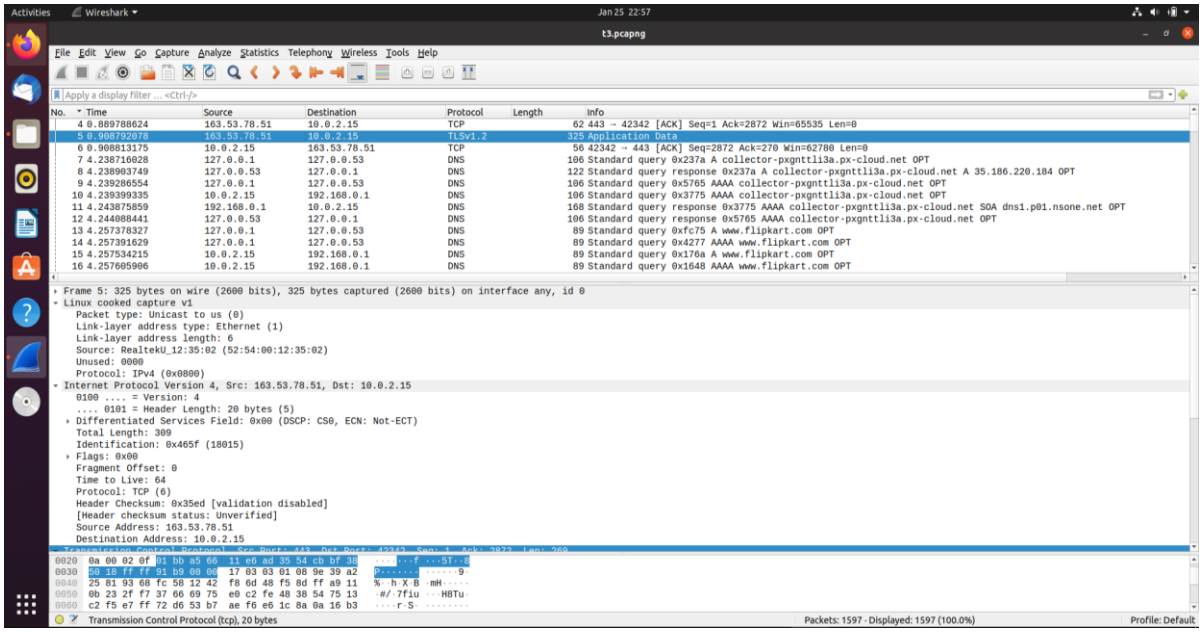
Step 5: Analyze the following in Wireshark

On Packet List Pane, select the first echo packet on the list. On Packet Details Pane, click on each of the four “+” to expand the information. Analyze the frames with the first echo request and

echo reply and complete the table below.

Details	First Echo Request	First Echo Reply
Frame Number	4	5
Source IP address	163.53.78.51	163.53.78.51
Destination IP address	10.0.2.15	10.0.2.15
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	00:00:00:00:00:00	00:00:00:00:00:00
Destination Ethernet Address	00:00:00:00:00:00	00:00:00:00:00:00
Internet Protocol Version	IPV4	IPV4
Time To Live (TTL) Value	64	64



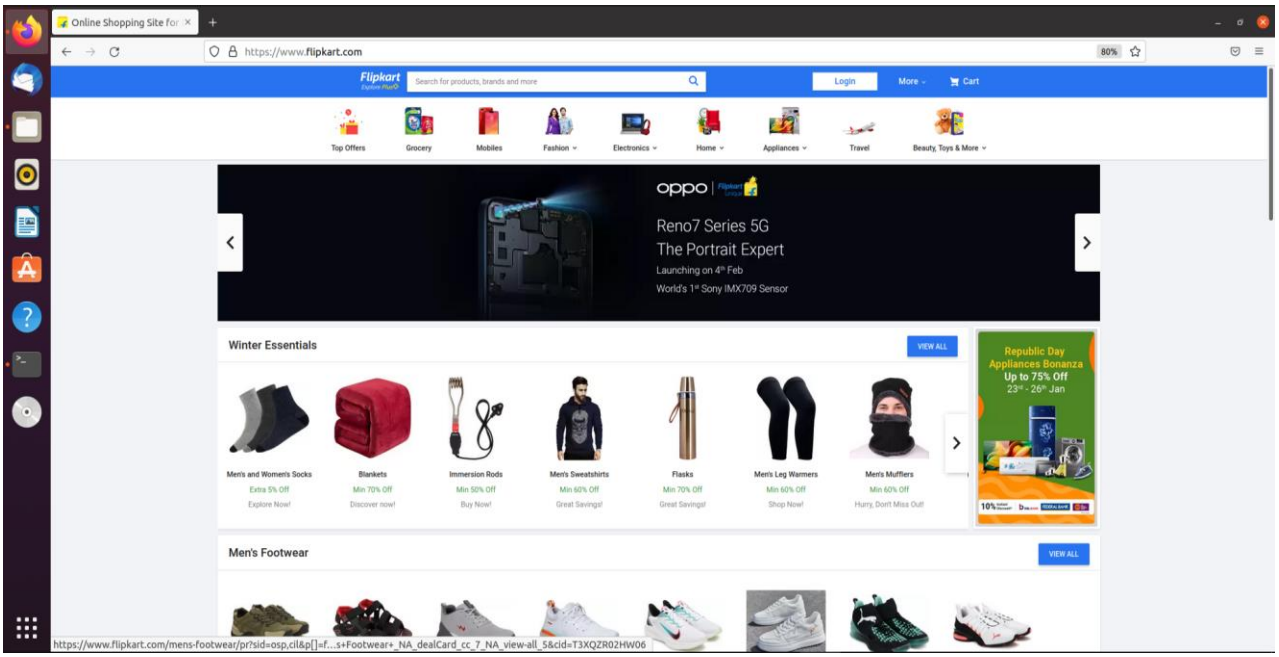


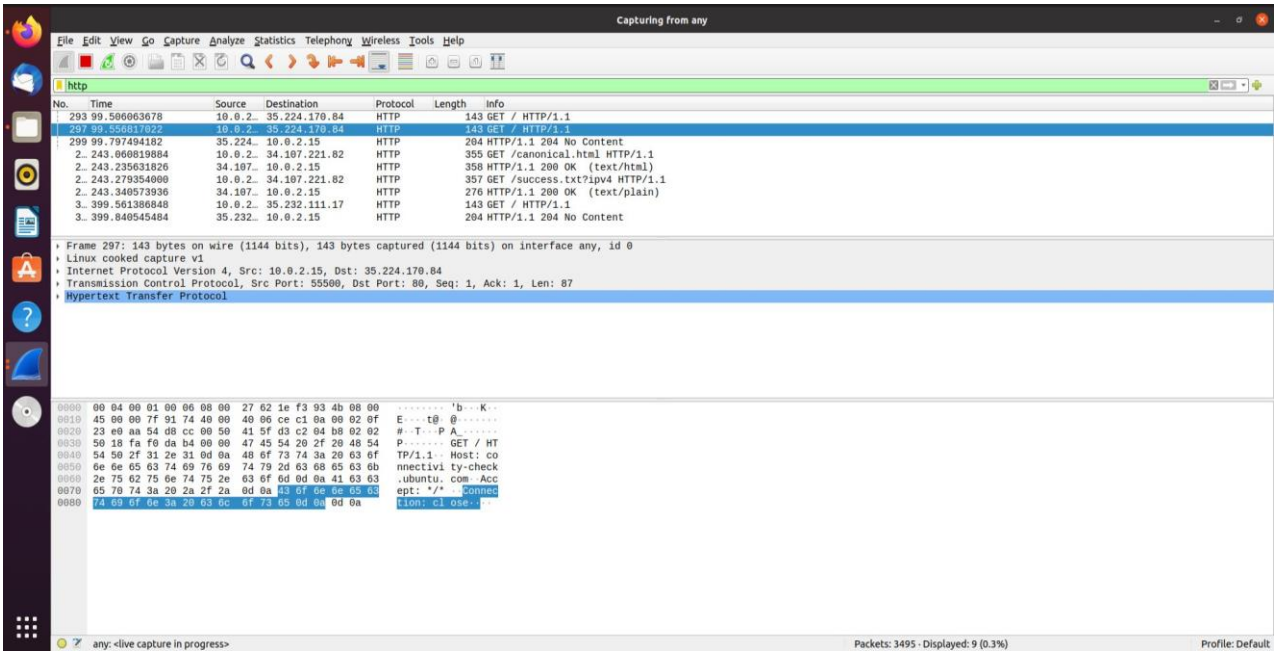
Task 3: HTTP PDU Capture

Using Wireshark's Filter feature

Step 1: Launch Wireshark and select 'any' interface. On the Filter toolbar, type-in 'http' and press enter

Step 2: Open Firefox browser, and browse www.flipkart.com



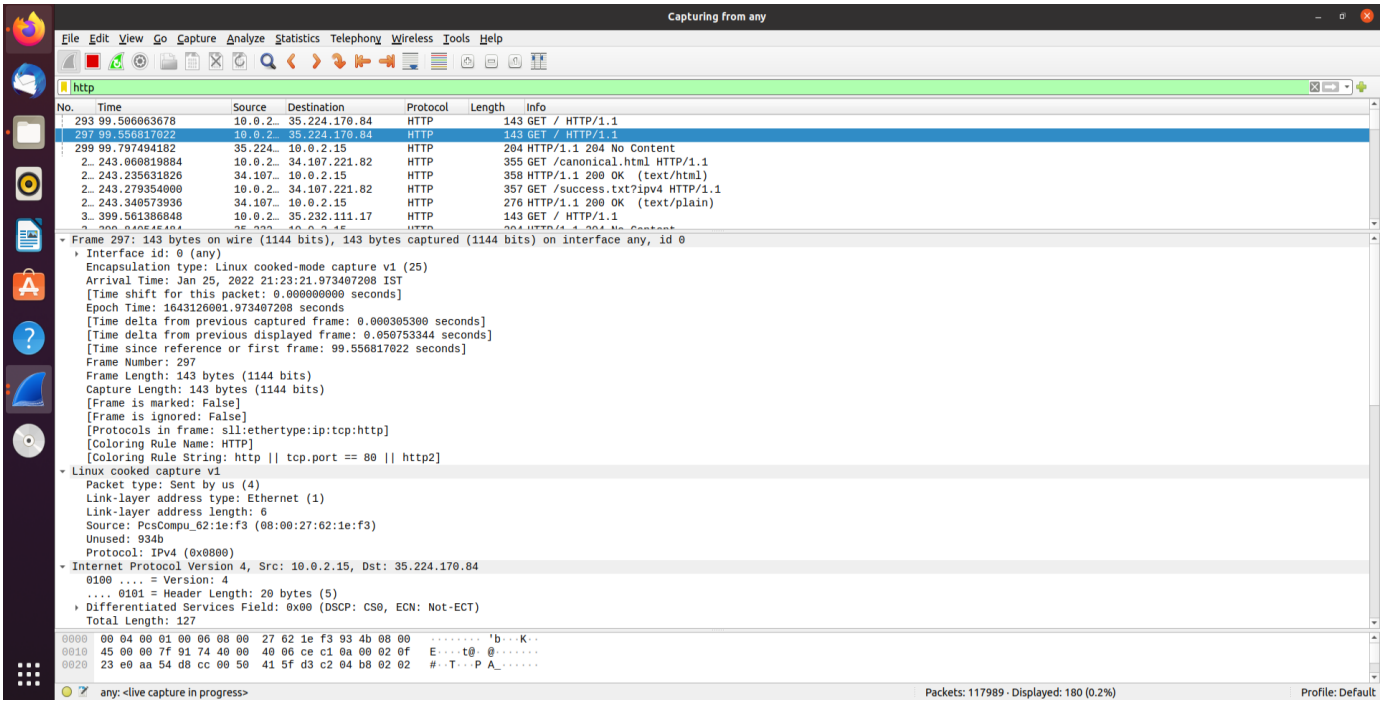


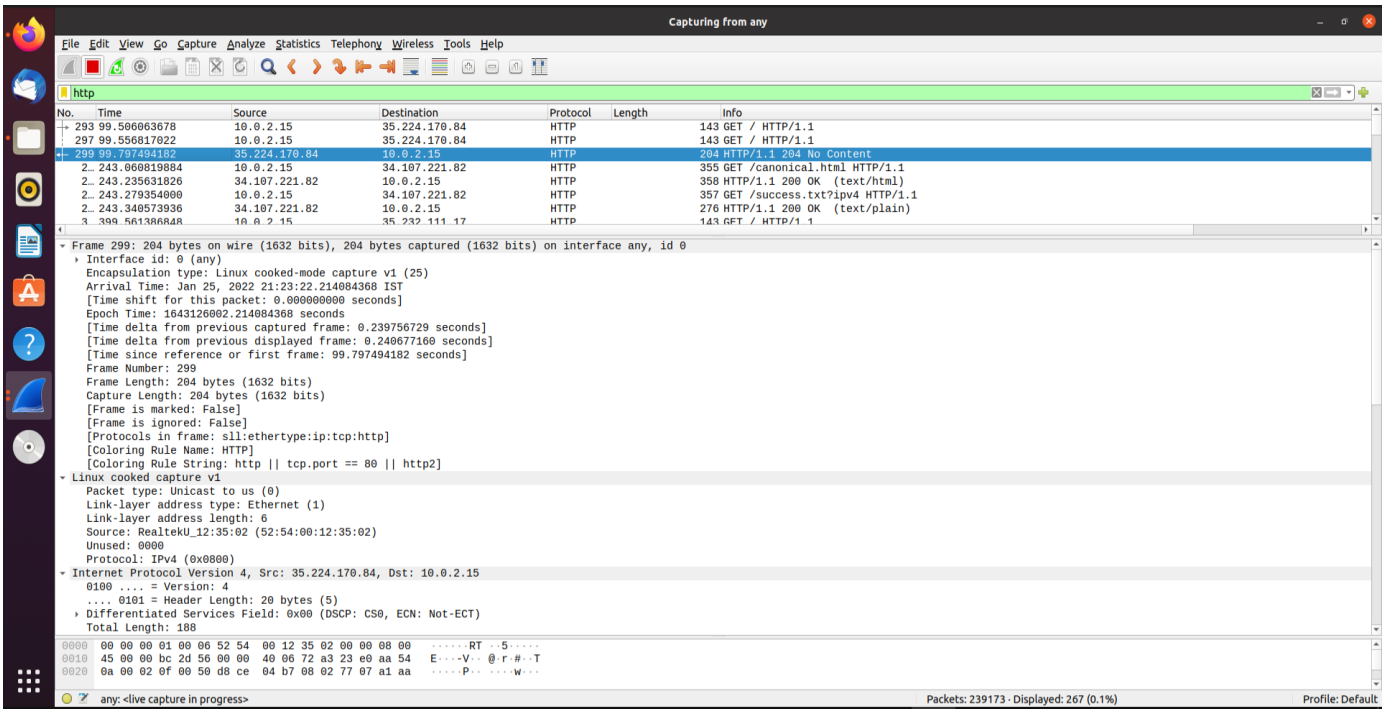
Observations to be made

Step 3: Analyze the first (interaction of host to the web server) and second frame (responseof server to the client). By analyzing the filtered frames, complete the table below:

Details	First Echo Request	First Echo Reply
Frame Number	297	299
Source Port	55502	80
Destination Port	80	55502
Source IP address	10.0.2.15	35.224.170.84
Destination IP address	35.224.170.84	10.0.2.15
Source Ethernet Address	52:54:00:12:35:02	52:54:00:12:35:02
Destination Ethernet Address	52:54:00:12:35:02	00:0c:29:f8:f2:8e

Step 4: Analyze the HTTP request and response and complete the table below.





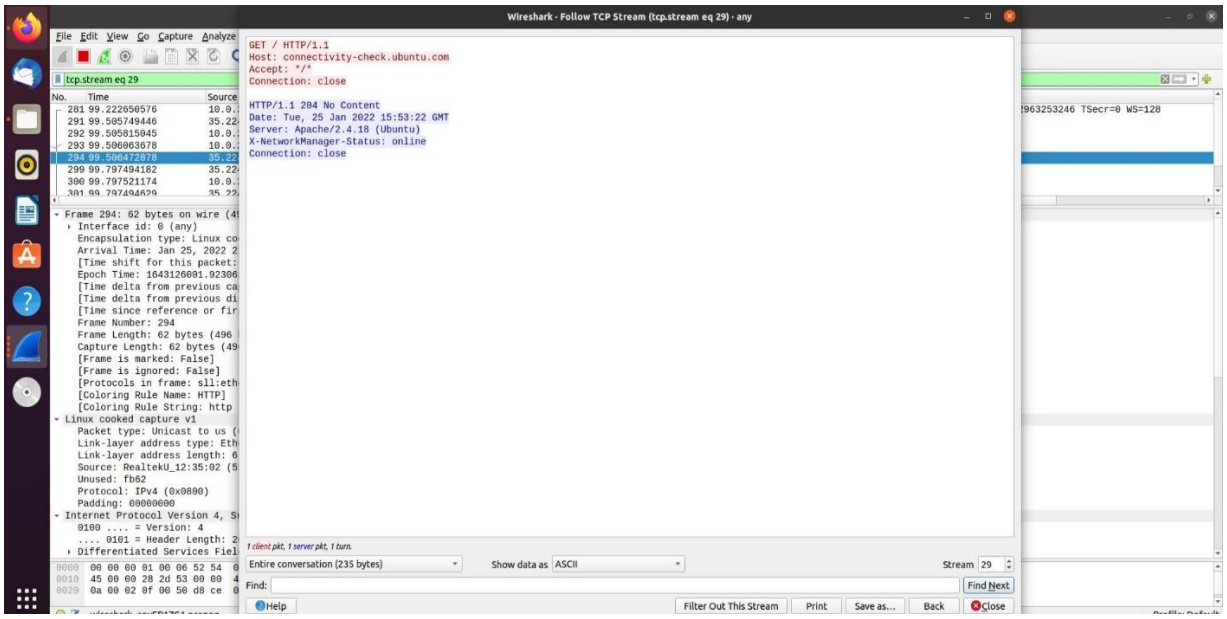
HTTP Request		**HTTP Response**	
Get	[GET / HTTP/1.1\r\n]	Server	HTTP/1.1
Host	www.flipkart.com	Content-Type	text/html
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0)	Date	Date: Mon, 24 Jan 2022 22:18:04 GMT

	Gecko/20100101 Firefox/96.0		
Accept-Language	en-US,en;q=0.5	Location	https://www.flipkart.com
Accept-Encoding	gzip, deflate	Content-Length	188
Connection	Keep-alive	Connection	keep-alive

Using Wireshark's Follow TCP Stream

Step 1: Make sure the filter is blank. Right-click any packet inside the Packet List Pane, then select 'Follow TCP Stream'. For demo purpose, a packet containing the HTTP GET request "GET / HTTP / 1.1" can be selected.

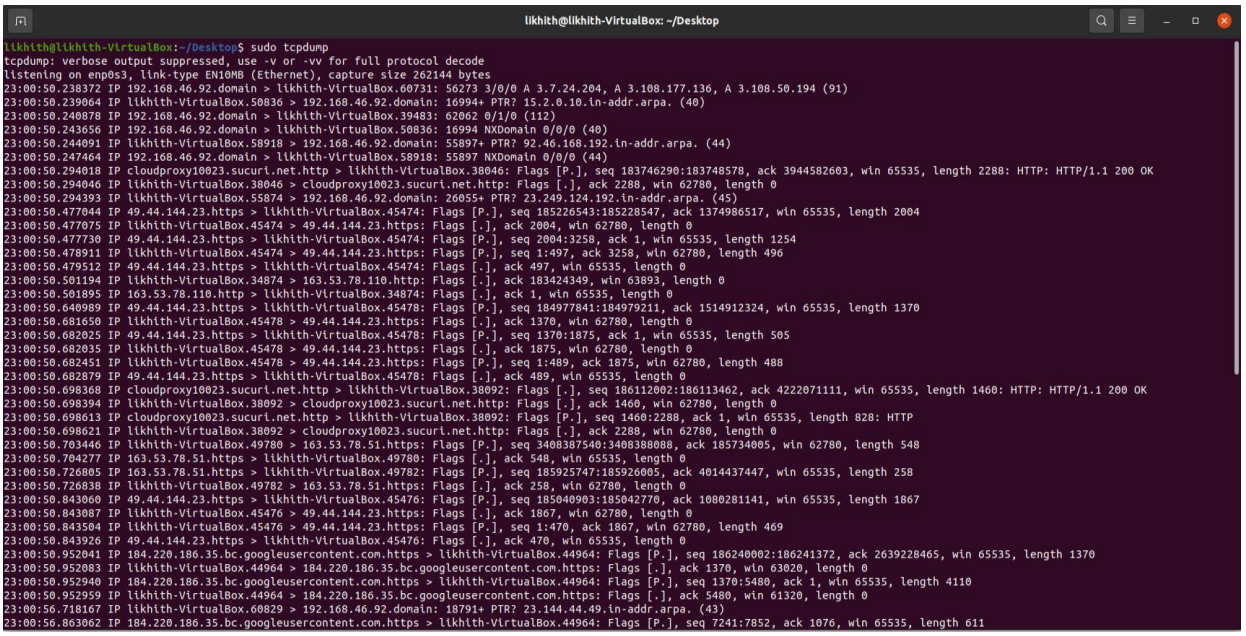
Step 2: Upon following a TCP stream, screenshot the whole window.



Task 4: Capturing packets with tcpdump

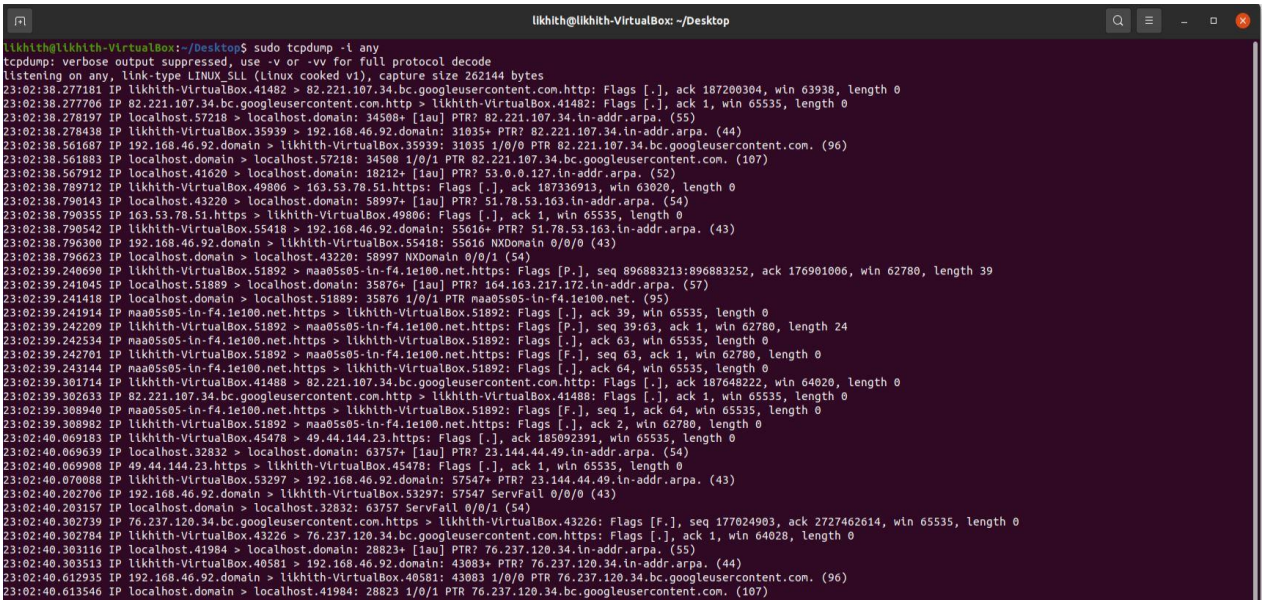
Step 1: Use the command **tcpdump -D** to see which interfaces are available for capture.

sudo tcpdump -D



Step 2: Capture all packets in any interface by running this command:

sudo tcpdump -i any



Note: Perform some pinging operation while giving above command.

Also typewww.google.com in browser.

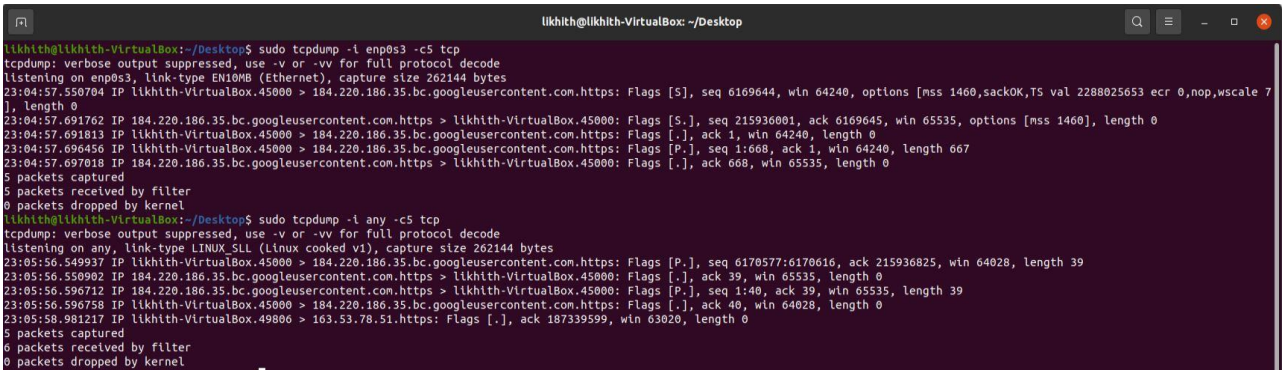
Observation

Step 3: Understand the output format.

Step 4: To filter packets based on protocol, specifying the protocol in the command line.

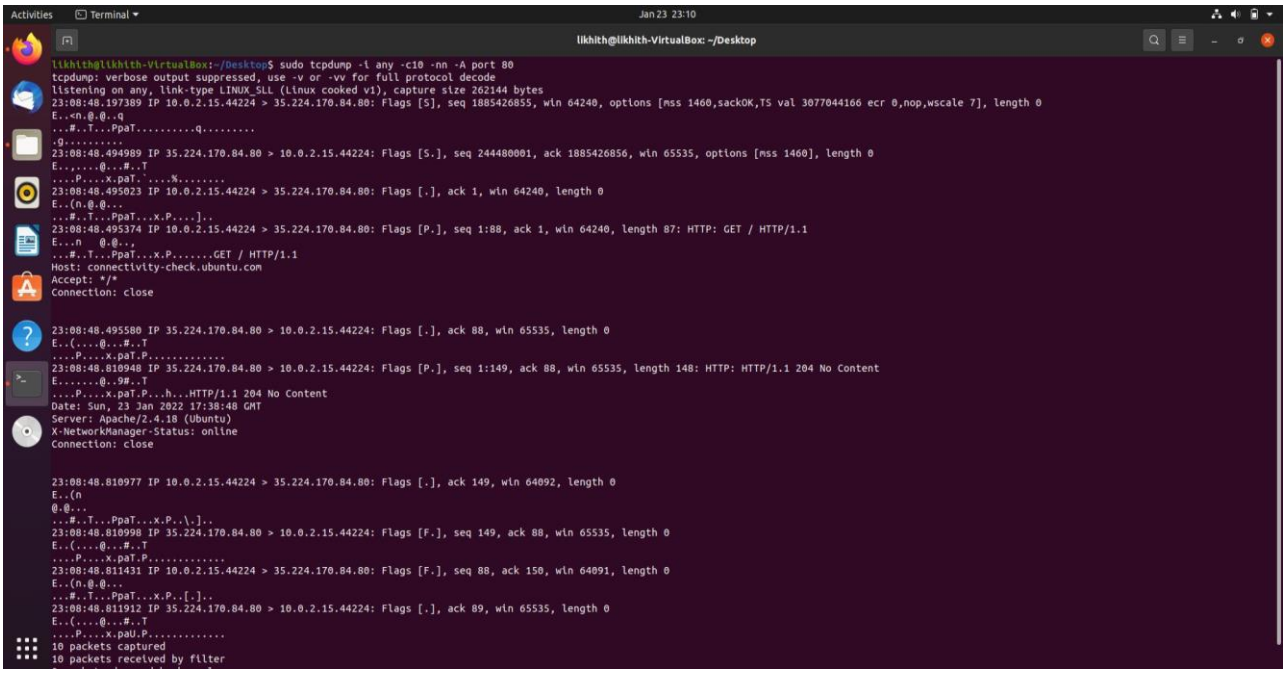
Forexample, capture ICMP packets only by using this command:

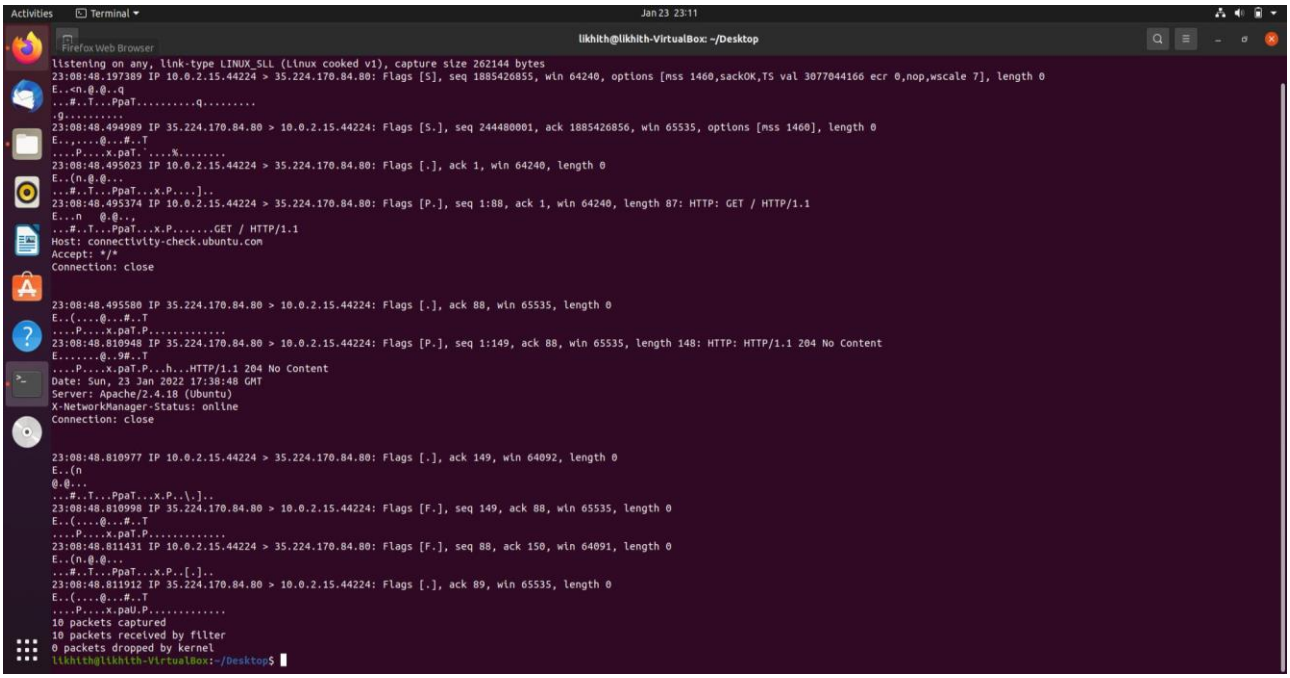
sudo tcpdump -i any -c5 icmp



Step 5: Check the packet content. For example, inspect the HTTP content of a web request like this:

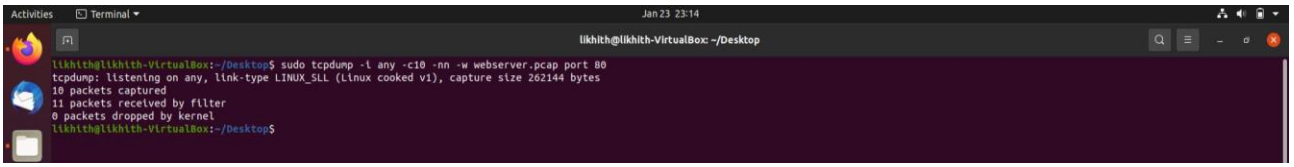
sudo tcpdump -i any -c10 -nn -A port 80





Step 6: To save packets to a file instead of displaying them on screen, use the option -w:

sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80

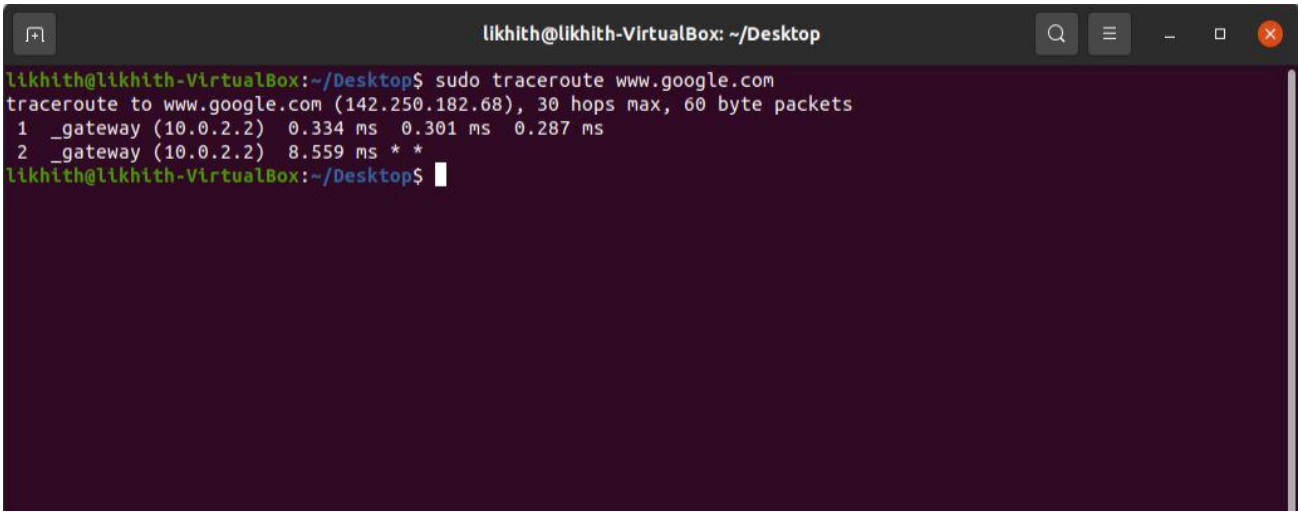


Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com

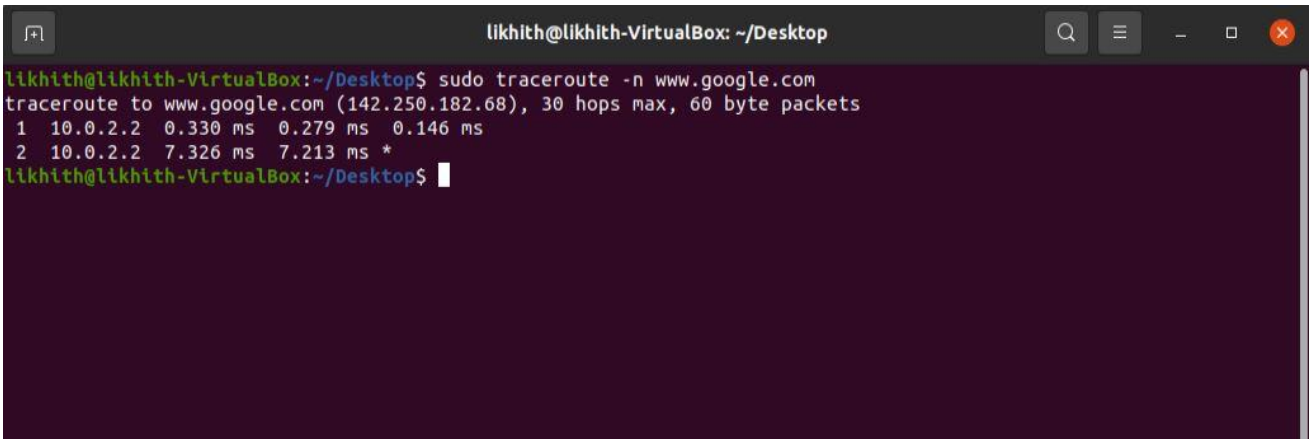
Step 2: Analyze destination address of google.com and no. of hops



Step 3: To speed up the process, you can disable the mapping of IP

addresses with hostnamesby using the *-n* option

sudo traceroute -n www.google.com



Step 4: The -I option is necessary so that the traceroute uses ICMP.

sudo traceroute -I www.google.com

```
likhith@likhith-VirtualBox: ~/Desktop
likhith@likhith-VirtualBox:~/Desktop$ sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.182.68), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.579 ms  0.545 ms  0.539 ms
 2  192.168.102.108 (192.168.102.108)  10.318 ms  10.860 ms  10.854 ms
 3  * * *
 4  10.72.169.67 (10.72.169.67)  88.406 ms 10.72.169.3 (10.72.169.3)  94.366 ms  99.782 ms
 5  192.168.61.46 (192.168.61.46)  93.807 ms  99.926 ms 192.168.61.44 (192.168.61.44)  101.319 ms
 6  192.168.61.45 (192.168.61.45)  101.311 ms  92.509 ms  93.156 ms
 7  172.26.74.84 (172.26.74.84)  98.469 ms  91.925 ms  91.893 ms
 8  172.26.74.99 (172.26.74.99)  97.454 ms  30.065 ms  45.924 ms
 9  192.168.61.18 (192.168.61.18)  44.939 ms  50.245 ms 192.168.61.20 (192.168.61.20)  50.661 ms
10  192.168.61.19 (192.168.61.19)  49.263 ms  54.492 ms  50.288 ms
11  172.31.2.63 (172.31.2.63)  55.551 ms  56.385 ms  55.973 ms
12  74.125.51.4 (74.125.51.4)  55.292 ms  55.616 ms  55.980 ms
13  209.85.142.223 (209.85.142.223)  41.011 ms  40.982 ms  48.338 ms
14  142.251.55.247 (142.251.55.247)  59.537 ms  60.001 ms  54.711 ms
15  maa05s20-in-f4.1e100.net (142.250.182.68)  54.190 ms  48.307 ms  47.911 ms
likhith@likhith-VirtualBox:~/Desktop$
```

Step 5: By default, traceroute uses icmp (ping) packets. If you'd rather test a TCP connectionto gather data more relevant to web server, you can use the -T flag.

sudo traceroute -T www.google.com

```
likhith@likhith-VirtualBox: ~/Desktop
likhith@likhith-VirtualBox:~/Desktop$ sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.195.228), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.405 ms  0.355 ms  0.336 ms
 2  maa03s43-in-f4.1e100.net (142.250.195.228)  133.368 ms  145.073 ms  144.618 ms
likhith@likhith-VirtualBox:~/Desktop$
```

Task 6: Explore an entire network for information (Nmap)

Step 1: You can scan a host using its host name or IP address, for instance.

nmap www.pes.edu

```
likhith@likhith-VirtualBox: ~/Desktop
likhith@likhith-VirtualBox:~/Desktop$ nmap www.pes.edu
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-24 22:32 IST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.075s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 8.09 seconds
```

Step 2: Alternatively, use an IP address to scan.

nmap 163.53.78.128

```
likhith@likhith-VirtualBox: ~/Desktop
likhith@likhith-VirtualBox:~/Desktop$ nmap -Pn 10.0.11.12
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-24 22:47 IST
Nmap scan report for 10.0.11.12
Host is up.
All 1000 scanned ports on 10.0.11.12 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 201.44 seconds
likhith@likhith-VirtualBox:~/Desktop$
```

Step 3: Scan multiple IP address or subnet (IPv4)

nmap 192.168.1.1 192.168.1.2 192.168.1.3

```
likhith@likhith-VirtualBox: ~/Desktop
likhith@likhith-VirtualBox:~/Desktop$ nmap 10.0.1.1 10.0.1.2 10.0.1.3
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-24 22:59 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.06 seconds
likhith@likhith-VirtualBox:~/Desktop$
```

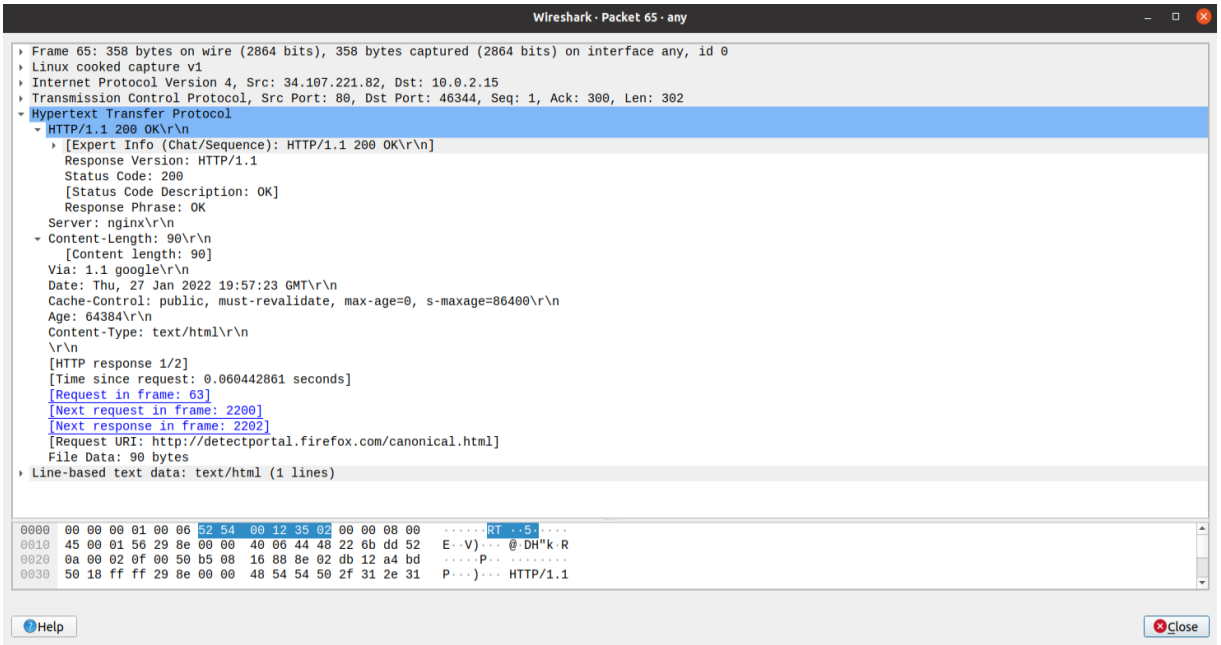
<u>Questions on above observations:</u>

1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?

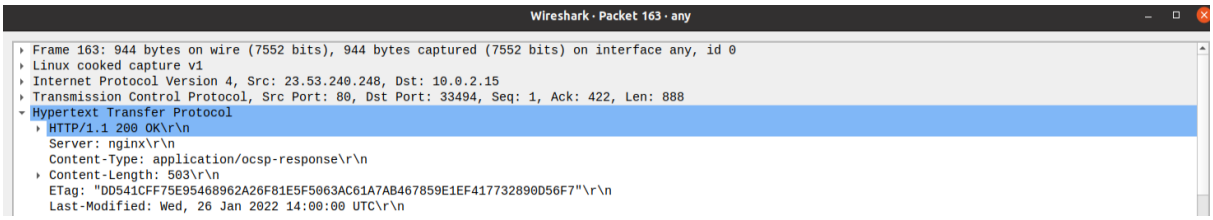
Ans: Version of HTTP of the server is 1.1

```
Wireshark · Packet 159 · any
> Frame 159: 477 bytes on wire (3816 bits), 477 bytes captured (3816 bits) on interface any, id 0
> Linux cooked capture v1
> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 23.53.240.248
> Transmission Control Protocol, Src Port: 33494, Dst Port: 80, Seq: 1, Ack: 1, Len: 421
v Hypertext Transfer Protocol
  > POST / HTTP/1.1\r\n
    Host: r3.o.lencr.org\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0\r\n
    Accept: */*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
```

Server HTTP version is also 1.1



2) When was the HTML file that you are retrieving last modified at the server?



3) How to tell ping to exit after a specified number of ECHO_REQUEST packets?

Ans: By using the ping -c <<number of packets>> <<URL or IP address>>

```
likhith@likhith-VirtualBox: ~/Desktop
likhith@likhith-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::7e4e:3870:b89f:cdce  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:62:1e:f3  txqueuelen 1000  (Ethernet)
        RX packets 67  bytes 14618 (14.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 117  bytes 12701 (12.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 152  bytes 12792 (12.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 152  bytes 12792 (12.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

likhith@likhith-VirtualBox:~/Desktop$ ping -c 5 www.flipkart.com
PING flipkart.com (163.53.76.86) 56(84) bytes of data.
64 bytes from 163.53.76.86 (163.53.76.86): icmp_seq=1 ttl=54 time=188 ms
64 bytes from 163.53.76.86 (163.53.76.86): icmp_seq=2 ttl=54 time=60.6 ms
64 bytes from 163.53.76.86 (163.53.76.86): icmp_seq=3 ttl=54 time=55.5 ms
64 bytes from 163.53.76.86 (163.53.76.86): icmp_seq=4 ttl=54 time=61.7 ms
64 bytes from 163.53.76.86 (163.53.76.86): icmp_seq=5 ttl=54 time=60.3 ms

--- flipkart.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 55.516/85.176/187.722/51.316 ms
likhith@likhith-VirtualBox:~/Desktop$
```

4) How will you identify remote host apps and OS?

Ans: using “nmap” command in terminal to probe the remote computer and based on it's responses to TCP packets, “nmap” can infer what operating system it is using. By typing the above command you can identify remote host apps and OS