COMPUTER NETWORKS LAB WEEK #1

Name: Laxmikant Bhujang Gurav

SRN: PES1UG20CS658

Roll no : 55

Section : K

 # Installing necessary utilities

sudo apt-get install ***name_of_the_tool***

```
laxmikant@MyPC:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 178 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Fetched 196 kB in 1s (257 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 184471 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
```

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	IPv4: 10.0.2.15 IPv6: fe80::e1d4:bcb3:7fd5:10e6	08:00:27:cb:5a:f6
lo	IPv4: 127.0.0.1 IPv6: ::1	00:00:00:00:00:00

```
laxmikant@MyPC:~$ 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::e1d4:bcb3:7fd5:10e6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:cb:5a:f6  txqueuelen 1000  (Ethernet)
        RX packets 8808  bytes 11979229 (11.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2898  bytes 204889 (204.8 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 205  bytes 17620 (17.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 205  bytes 17620 (17.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
laxmikant@MyPC:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cb:5a:f6 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 83581sec preferred_lft 83581sec
    inet6 fe80::e1d4:bcb3:7fd5:10e6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

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**

```
laxmikant@MyPC:~$ sudo ifconfig enp0s3 10.0.11.58 netmask 255.255.255.0
laxmikant@MyPC:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.11.58  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::e1d4:bcb3:7fd5:10e6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:cb:5a:f6  txqueuelen 1000  (Ethernet)
        RX packets 18452  bytes 20460358 (20.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8817  bytes 1081374 (1.0 MB)
        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 1290  bytes 132818 (132.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1290  bytes 132818 (132.8 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

```
laxmikant@MyPC:~$ sudo ifconfig lo down
laxmikant@MyPC:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.11.58  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::e1d4:bcb3:7fd5:10e6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:cb:5a:f6  txqueuelen 1000  (Ethernet)
        RX packets 18665  bytes 20522727 (20.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8963  bytes 1097905 (1.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

sudo ifconfig interface_name up

```
laxmikant@MyPC:~$ sudo ifconfig lo up
laxmikant@MyPC:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.11.58  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::e1d4:bcb3:7fd5:10e6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:cb:5a:f6  txqueuelen 1000  (Ethernet)
        RX packets 18690  bytes 20525874 (20.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8986  bytes 1100406 (1.1 MB)
        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 1522  bytes 157203 (157.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1522  bytes 157203 (157.2 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

```
laxmikant@MyPC:~$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 REACHABLE
laxmikant@MyPC:~$
```

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.

```
laxmikant@MyPC:~$ sudo ifconfig enp0s3 10.0.11.58 netmask 255.255.255.0
[sudo] password for laxmikant:
laxmikant@MyPC:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.11.58  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::e1d4:bcb3:7fd5:10e6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:cb:5a:f6  txqueuelen 1000  (Ethernet)
        RX packets 20025  bytes 20687420 (20.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10088  bytes 1186774 (1.1 MB)
        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 1635  bytes 168864 (168.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1635  bytes 168864 (168.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

laxmikant@MyPC:~$
```

Step 2: Launch Wireshark and select ‘any’ interface

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

Observations to be made

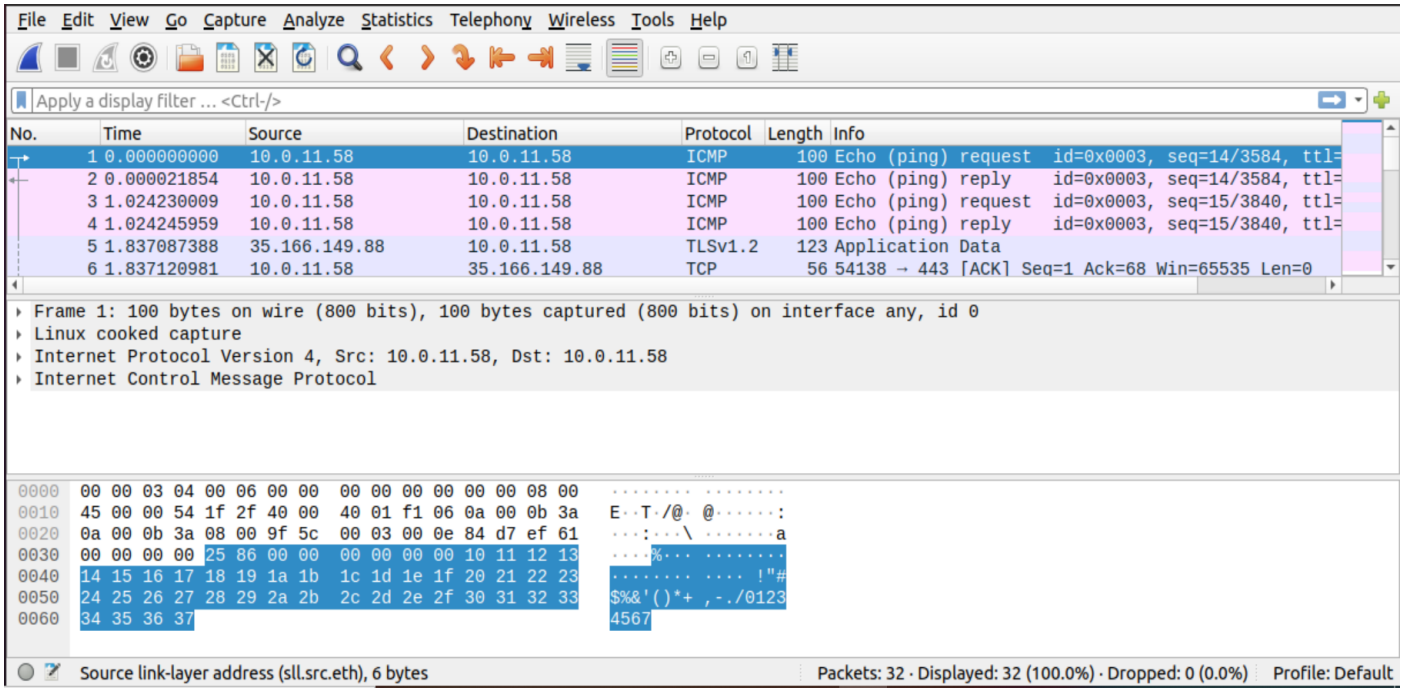
Step 4: Analyze the following in Terminal

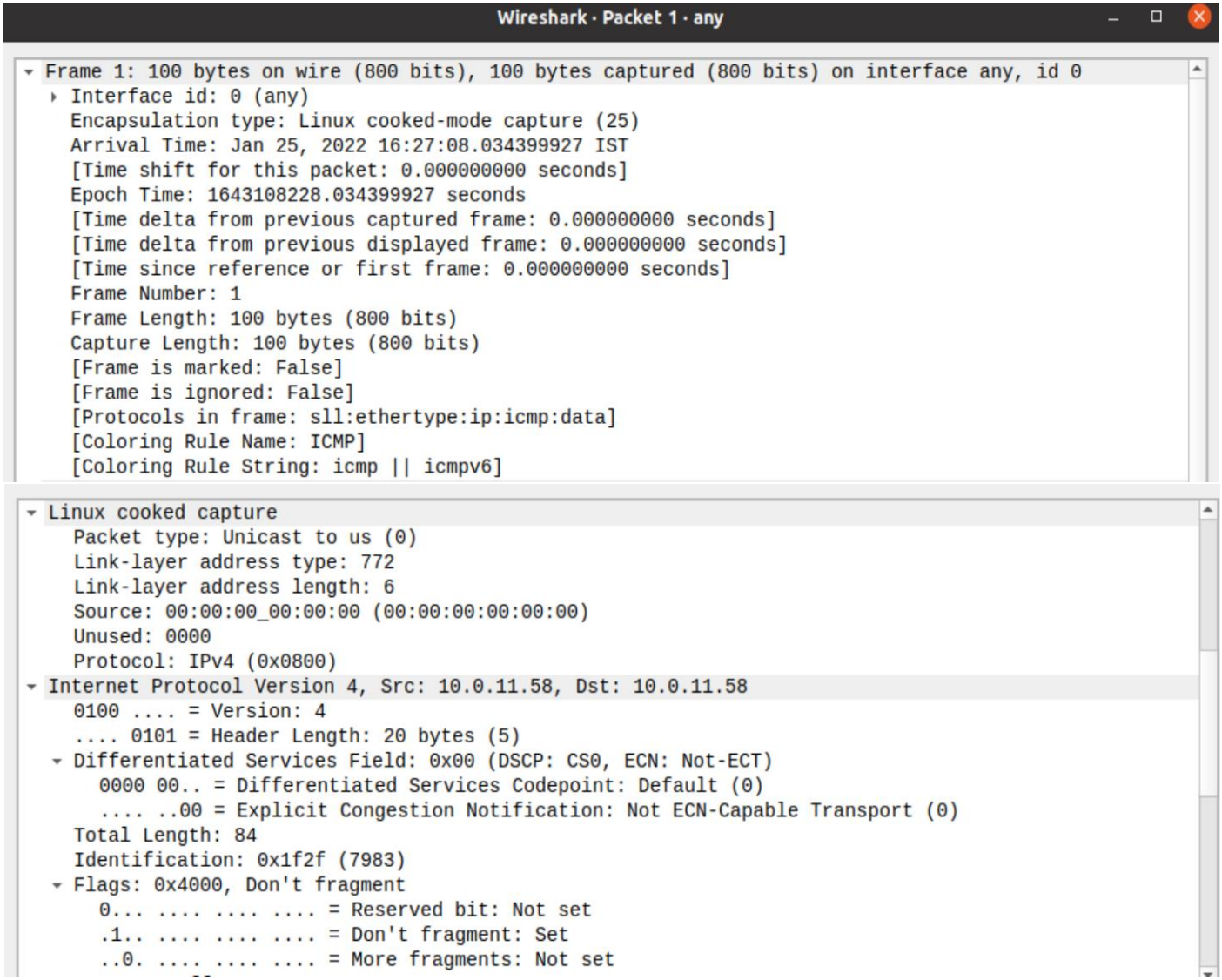
- TTL
- Protocol used by ping
- Time

```
laxmikant@MyPC:~$ ping 10.0.11.58
PING 10.0.11.58 (10.0.11.58) 56(84) bytes of data.
64 bytes from 10.0.11.58: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 10.0.11.58: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.0.11.58: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from 10.0.11.58: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.0.11.58: icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from 10.0.11.58: icmp_seq=6 ttl=64 time=0.072 ms
64 bytes from 10.0.11.58: icmp_seq=7 ttl=64 time=0.056 ms
```

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.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.11.58 10.0.11.58 ICMP 100 Echo (ping) request id=0x0003, seq=14/3584, ttl=
2 0.000021854 10.0.11.58 10.0.11.58 ICMP 100 Echo (ping) reply id=0x0003, seq=14/3584, ttl=
3 1.024230009 10.0.11.58 10.0.11.58 ICMP 100 Echo (ping) request id=0x0003, seq=15/3840, ttl=
4 1.024245959 10.0.11.58 10.0.11.58 ICMP 100 Echo (ping) reply id=0x0003, seq=15/3840, ttl=
5 1.837087388 35.166.149.88 10.0.11.58 TLSv1.2 123 Application Data
6 1.837120981 10.0.11.58 35.166.149.88 TCP 56 54138 → 443 [ACK] Seq=1 Ack=68 Win=65535 Len=0
Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.11.58, Dst: 10.0.11.58
Internet Control Message Protocol
0000 00 00 03 04 00 06 00 00 00 00 00 00 00 08 00
0010 45 00 00 54 1f 2f 40 00 40 01 f1 06 0a 00 0b 3a
0020 0a 00 0b 3a 08 00 9f 5c 00 03 00 0e 84 d7 ef 61
0030 00 00 00 00 25 86 00 00 00 00 00 00 10 11 12 13
0040 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
0050 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33
0060 34 35 36 37
Source link-layer address (sll.src.eth), 6 bytes
Packets: 32 · Displayed: 32 (100.0%) · Dropped: 0 (0.0%)
Profile: Default


Wireshark · Packet 1 · any
Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0
Interface id: 0 (any)
Encapsulation type: Linux cooked-mode capture (25)
Arrival Time: Jan 25, 2022 16:27:08.034399927 IST
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1643108228.034399927 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 100 bytes (800 bits)
Capture Length: 100 bytes (800 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: sll:ethertype:ip:icmp:data]
[Coloring Rule Name: ICMP]
[Coloring Rule String: icmp || icmpv6]
Linux cooked capture
Packet type: Unicast to us (0)
Link-layer address type: 772
Link-layer address length: 6
Source: 00:00:00_00:00:00 (00:00:00:00:00:00)
Unused: 0000
Protocol: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.11.58, Dst: 10.0.11.58
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 84
Identification: 0x1f2f (7983)
Flags: 0x4000, Don't fragment
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment: Set
..0. .... = More fragments: Not set

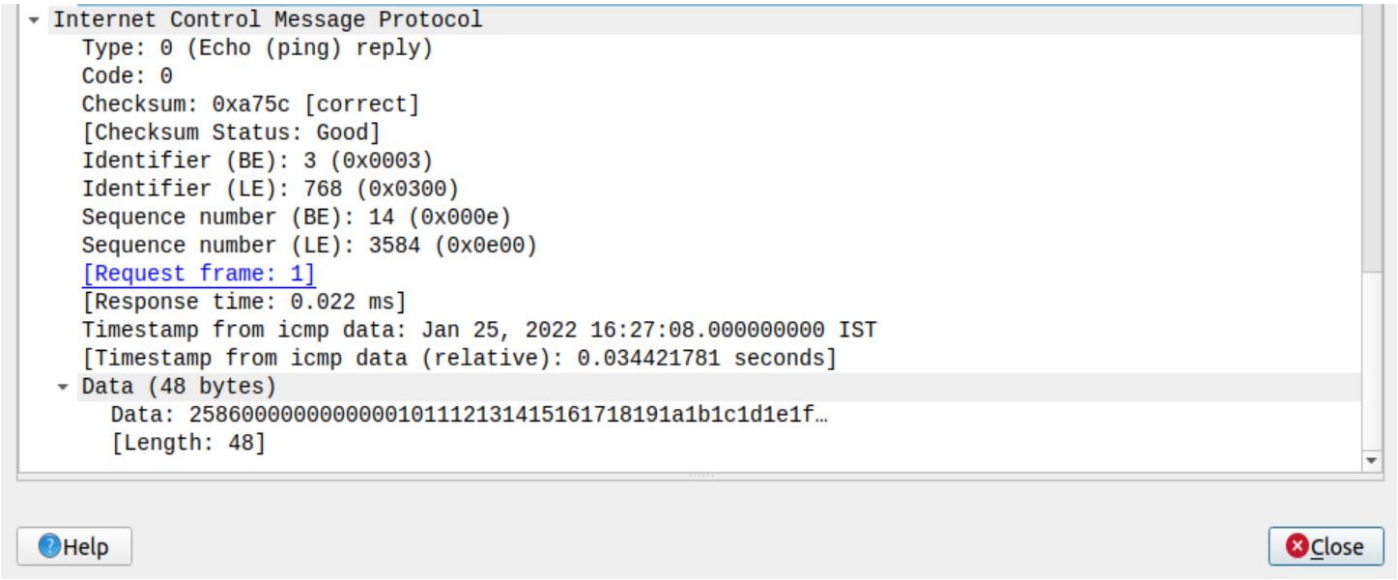
```
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0xf106 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.11.58
    Destination: 10.0.11.58
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x9f5c [correct]
    [Checksum Status: Good]
    Identifier (BE): 3 (0x0003)
    Identifier (LE): 768 (0x0300)
    Sequence number (BE): 14 (0x000e)
    Sequence number (LE): 3584 (0x0e00)
    [Response frame: 2]
    Timestamp from icmp data: Jan 25, 2022 16:27:08.000000000 IST
    [Timestamp from icmp data (relative): 0.034399927 seconds]
  ▸ Data (48 bytes)
```

Help Close

Wireshark · Packet 2 · any

```
▾ Frame 2: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0
  ▸ Interface id: 0 (any)
    Encapsulation type: Linux cooked-mode capture (25)
    Arrival Time: Jan 25, 2022 16:27:08.034421781 IST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1643108228.034421781 seconds
    [Time delta from previous captured frame: 0.000021854 seconds]
    [Time delta from previous displayed frame: 0.000021854 seconds]
    [Time since reference or first frame: 0.000021854 seconds]
    Frame Number: 2
    Frame Length: 100 bytes (800 bits)
    Capture Length: 100 bytes (800 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: sll:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
▾ Linux cooked capture
    Packet type: Unicast to us (0)
```

```
    Link-layer address type: 772
    Link-layer address length: 6
    Source: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Unused: 0000
    Protocol: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 10.0.11.58, Dst: 10.0.11.58
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x1f30 (7984)
  ▸ Flags: 0x0000
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x3106 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.11.58
    Destination: 10.0.11.58
```



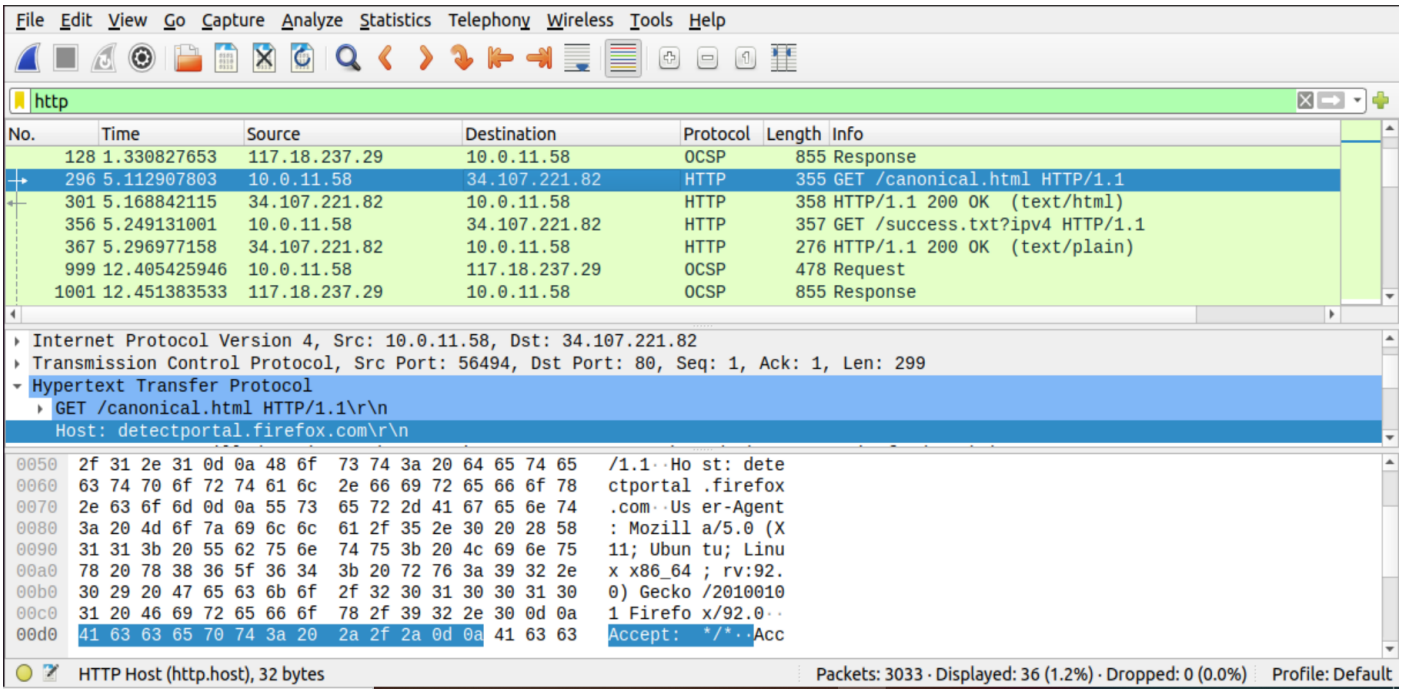
Details	First Echo Request	First Echo Reply
Frame Number	1	2
Source IP address	10.0.11.58	10.0.11.58
Destination IP address	10.0.11.58	10.0.11.58
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	00.00.00.00.00.00	00.00.00.00.00.00
Destination Ethernet Address	00.00.00.00.00.00	00.00.00.00.00.00
Internet Protocol Version	4	4
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

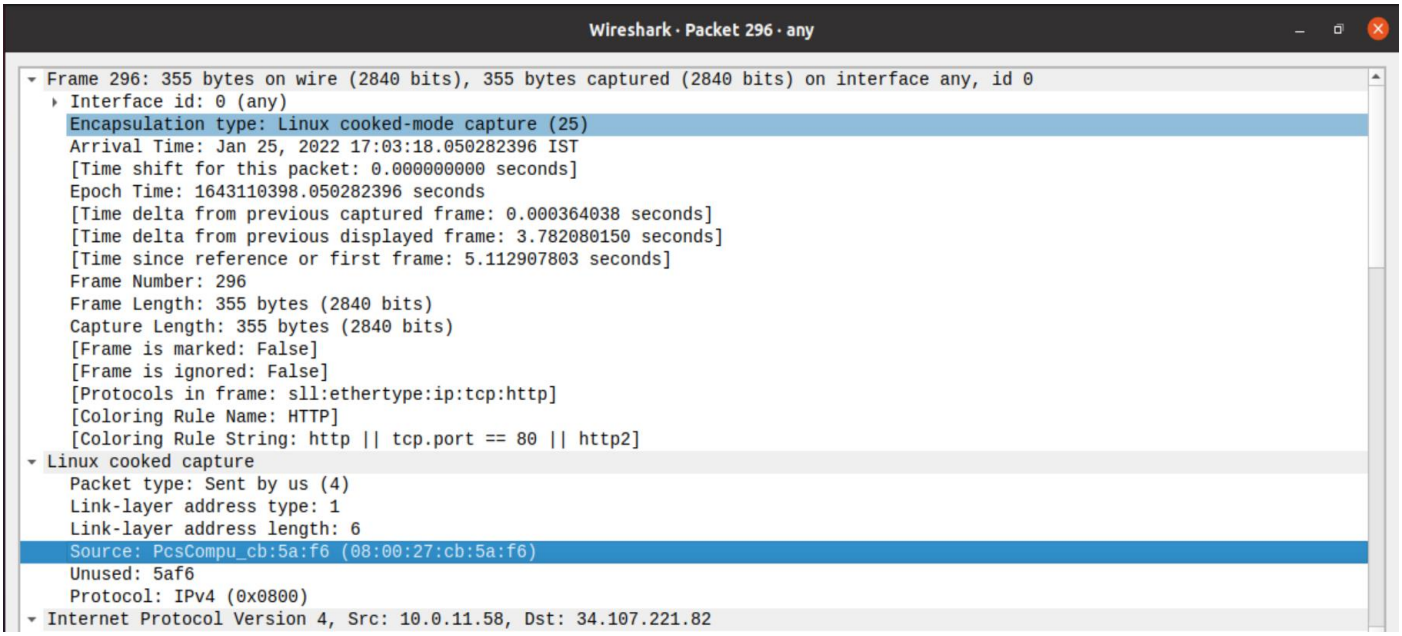
Step 2: Open Firefox browser, and browse www.flipkart.com



Observations to be made

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

Details	First Echo Request	First Echo Reply
Frame Number	296	301
Source Port	56494	80
Destination Port	80	56494
Source IP address	10.0.11.58	34.107.221.82
Destination IP address	34.107.221.82	10.0.11.58
Source Ethernet Address	08.00.27.cb.5a.f6	52.54.00.12.35.02
Destination Ethernet Address	52.54.00.12.35.02	08.00.27.cb.5a.f6

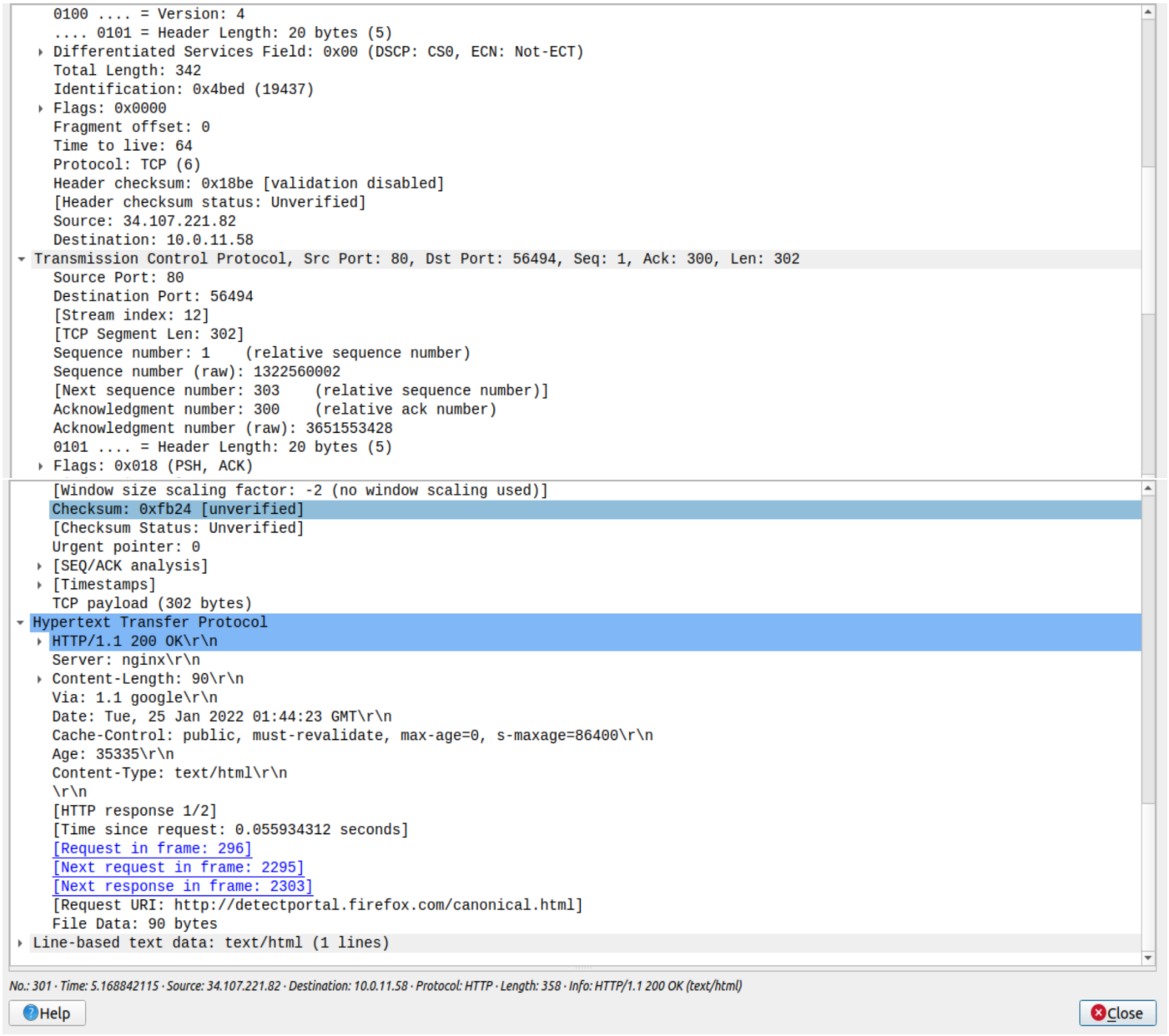


```
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 339
    Identification: 0xe3e8 (58344)
  ▸ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x40c5 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.11.58
    Destination: 34.107.221.82
▾ Transmission Control Protocol, Src Port: 56494, Dst Port: 80, Seq: 1, Ack: 1, Len: 299
    Source Port: 56494
    Destination Port: 80
    [Stream index: 12]
    [TCP Segment Len: 299]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 3651553129
    [Next sequence number: 300    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 1322560002
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 64240
    [Calculated window size: 64240]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x163d [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (299 bytes)
▾ Hypertext Transfer Protocol
  ▸ GET /canonical.html HTTP/1.1\r\n
    Host: detectportal.firefox.com\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
    Cache-Control: no-cache\r\n
    Pragma: no-cache\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://detectportal.firefox.com/canonical.html]
    [HTTP request 1/2]
    [Response in frame: 301]
    [Next request in frame: 2295]
```

Help Close

Wireshark · Packet 301 · any

```
▾ Frame 301: 358 bytes on wire (2864 bits), 358 bytes captured (2864 bits) on interface any, id 0
  ▸ Interface id: 0 (any)
    Encapsulation type: Linux cooked-mode capture (25)
    Arrival Time: Jan 25, 2022 17:03:18.106216708 IST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1643110398.106216708 seconds
    [Time delta from previous captured frame: 0.000698326 seconds]
    [Time delta from previous displayed frame: 0.055934312 seconds]
    [Time since reference or first frame: 5.168842115 seconds]
    Frame Number: 301
    Frame Length: 358 bytes (2864 bits)
    Capture Length: 358 bytes (2864 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: sll:ethertype:ip:tcp:http:data-text-lines]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
▾ Linux cooked capture
    Packet type: Unicast to us (0)
    Link-layer address type: 1
    Link-layer address length: 6
    Source: RealtekU_12:35:02 (52:54:00:12:35:02)
    Unused: e842
    Protocol: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 34.107.221.82, Dst: 10.0.11.58
```



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

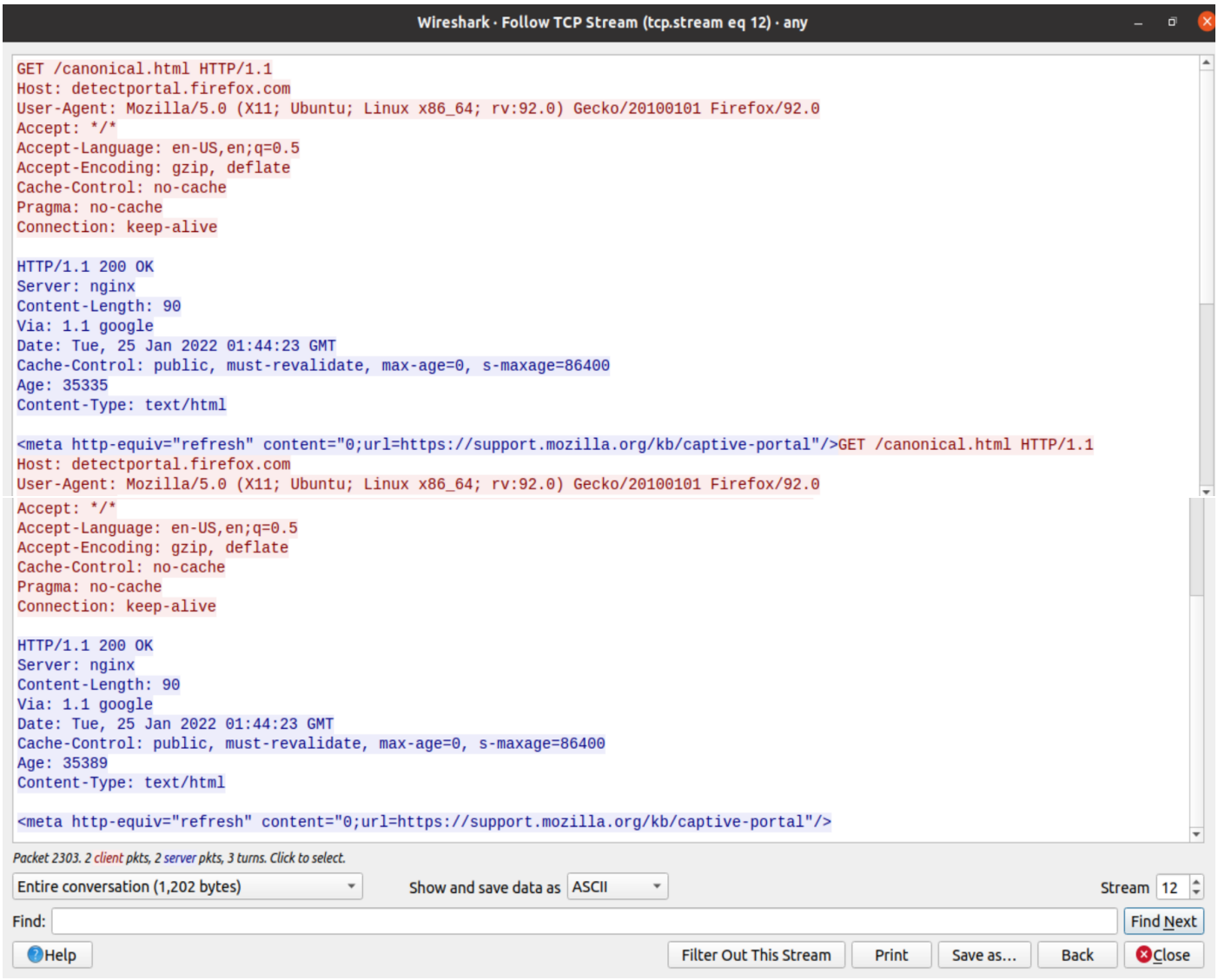
HTTP Request		**HTTP Response**	
Get	/canonical.html HTTP/1.1	Server	nginx
Host	detectportal.firefox.com	Content-Type	Text/html
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko 20100101 Firefox/92.0	Date	Tue, 25 Jan 2022 01:44:23 GMT
Accept-Language	en-US, en ; q=0.5	Location	https://www.flipkart.com/
Accept-Encoding	gzip , deflate	Content-Length	90
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

```
laxmikant@MyPC:~$ sudo tcpdump -D
1.enp0s3 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
laxmikant@MyPC:~$
```

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

sudo tcpdump -i any

```
laxmikant@MyPC:~$ sudo tcpdump -i any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
11:11:11.844268 IP localhost.60304 > localhost.domain: 60712+ [1au] A? detectportal.firefox.com. (53)
11:11:11.844719 IP localhost.46209 > localhost.domain: 45912+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
11:11:11.845029 IP MyPC.39388 > 192.168.43.1.domain: 62791+ A? detectportal.firefox.com. (42)
11:11:11.845225 IP localhost.domain > localhost.46209: 45912 1/0/1 PTR localhost. (75)
11:11:11.845424 IP localhost.42283 > localhost.domain: 14635+ [1au] PTR? 1.43.168.192.in-addr.arpa. (54)
11:11:11.845663 IP MyPC.44259 > 192.168.43.1.domain: 2599+ PTR? 1.43.168.192.in-addr.arpa. (43)
11:11:11.850567 IP localhost.60304 > localhost.domain: 56598+ [1au] AAAA? detectportal.firefox.com. (53)
11:11:11.950627 IP 192.168.43.1.domain > MyPC.39388: 62791 3/0/0 CNAME detectportal.prod.mozaws.net., CNAME prod.detectportal.prod.c
loudops.mozgcp.net., A 34.107.221.82 (153)
11:11:11.951014 IP localhost.domain > localhost.60304: 60712 3/0/1 CNAME detectportal.prod.mozaws.net., CNAME prod.detectportal.prod
.cloudops.mozgcp.net., A 34.107.221.82 (164)
11:11:12.034830 IP localhost.52236 > localhost.domain: 12651+ [1au] A? content-signature-2.cdn.mozilla.net. (64)
11:11:12.035086 IP MyPC.55844 > 192.168.43.1.domain: 54793+ A? d2nxq2uap88usk.cloudfront.net. (47)
11:11:12.035792 IP localhost.52236 > localhost.domain: 42837+ [1au] AAAA? content-signature-2.cdn.mozilla.net. (64)
11:11:12.035977 IP MyPC.42226 > 192.168.43.1.domain: 52656+ AAAA? d2nxq2uap88usk.cloudfront.net. (47)
11:11:12.095111 IP 192.168.43.1.domain > MyPC.55844: 54793 4/4/0 A 13.249.208.2, A 13.249.208.5, A 13.249.208.102, A 13.249.208.89 (
248)
11:11:12.095304 IP localhost.domain > localhost.52236: 12651 5/0/1 CNAME d2nxq2uap88usk.cloudfront.net., A 13.249.208.2, A 13.249.20
8.5, A 13.249.208.102, A 13.249.208.89 (168)
11:11:12.102375 IP 192.168.43.1.domain > MyPC.42226: 52656 8/4/0 AAAA 2600:9000:215c:f800:a:da5e:7900:93a1, AAAA 2600:9000:215c:f600
:a:da5e:7900:93a1, AAAA 2600:9000:215c:ac00:a:da5e:7900:93a1, AAAA 2600:9000:215c:fa00:a:da5e:7900:93a1, AAAA 2600:9000:215c:b600:a:
da5e:7900:93a1, AAAA 2600:9000:215c:de00:a:da5e:7900:93a1, AAAA 2600:9000:215c:d800:a:da5e:7900:93a1, AAAA 2600:9000:215c:4200:a:da5
e:7900:93a1 (408)
```

Note: Perform some pinging operation while giving above command. Also type www.google.com in browser.

```
11:12:00.395610 IP maa05s12-in-f3.1e100.net.http > MyPC.46768: Flags [.], ack 1, win 65535, length 0
11:12:00.740986 IP localhost.38633 > localhost.domain: 750+ [1au] A? www.google.com. (43)
11:12:00.741235 IP localhost.domain > localhost.38633: 750 1/0/1 A 142.250.67.132 (59)
11:12:00.741873 IP localhost.38633 > localhost.domain: 18923+ [1au] AAAA? www.google.com. (43)
11:12:00.741976 IP localhost.domain > localhost.38633: 18923 1/0/1 AAAA 2404:6800:4009:811::2004 (71)
11:12:00.744459 IP MyPC.38540 > bom12s06-in-f4.1e100.net.443: UDP, length 1357
11:12:00.767390 IP localhost.60615 > localhost.domain: 26736+ [1au] A? www.google.com. (43)
11:12:00.767590 IP localhost.domain > localhost.60615: 26736 1/0/1 A 142.250.67.132 (59)
11:12:00.767800 IP localhost.39464 > localhost.domain: 7283+ [1au] AAAA? www.google.com. (43)
11:12:00.767887 IP localhost.domain > localhost.39464: 7283 1/0/1 AAAA 2404:6800:4009:811::2004 (71)
11:12:00.768210 IP localhost.56723 > localhost.domain: 31878+ [1au] A? www.gstatic.com. (44)
11:12:00.768347 IP MyPC.57551 > 192.168.43.1.domain: 64356+ A? www.gstatic.com. (33)
11:12:00.768660 IP localhost.53002 > localhost.domain: 46021+ [1au] A? encrypted-tbn0.gstatic.com. (55)
11:12:00.768761 IP MyPC.60210 > 192.168.43.1.domain: 5590+ A? encrypted-tbn0.gstatic.com. (44)
11:12:00.823446 IP 192.168.43.1.domain > MyPC.57551: 64356 1/0/0 A 142.250.192.3 (49)
11:12:00.823655 IP localhost.domain > localhost.56723: 31878 1/0/1 A 142.250.192.3 (60)
11:12:00.823815 IP localhost.45488 > localhost.domain: 7040+ [1au] AAAA? www.gstatic.com. (44)
11:12:00.823944 IP MyPC.53343 > 192.168.43.1.domain: 65003+ AAAA? www.gstatic.com. (33)
11:12:00.826878 IP 192.168.43.1.domain > MyPC.60210: 5590 1/0/0 A 142.250.182.46 (60)
11:12:00.827078 IP localhost.domain > localhost.53002: 46021 1/0/1 A 142.250.182.46 (71)
11:12:00.827245 IP localhost.41117 > localhost.domain: 10694+ [1au] AAAA? encrypted-tbn0.gstatic.com. (55)
11:12:00.827368 IP localhost.domain > localhost.41117: 10694 1/0/1 AAAA 2404:6800:4009:82a::200e (83)
11:12:00.847755 IP bom12s06-in-f4.1e100.net.443 > MyPC.38540: UDP, length 1357
11:12:00.848329 IP MyPC.38540 > bom12s06-in-f4.1e100.net.443: UDP, length 1357
```

After typing www.pes.edu in browser.

```
11:12:14.622793 IP localhost.56010 > localhost.domain: 4186+ [1au] A? www.pes.edu. (40)
11:12:14.623058 IP MyPC.37888 > 192.168.43.1.domain: 46554+ A? pesuniversity.azurewebsites.net. (49)
11:12:14.703168 IP localhost.45390 > localhost.domain: 16290+ [1au] A? www.pes.edu. (40)
11:12:14.716165 IP localhost.59254 > localhost.domain: 31776+ [1au] A? pes.edu. (36)
11:12:14.716335 IP localhost.domain > localhost.59254: 31776 1/0/1 A 52.172.204.196 (52)
11:12:14.716462 IP localhost.43261 > localhost.domain: 8229+ [1au] AAAA? pes.edu. (36)
11:12:14.716555 IP MyPC.37036 > 192.168.43.1.domain: 28824+ AAAA? pes.edu. (25)
11:12:14.730559 IP MyPC.36522 > 117.18.237.29.http: Flags [.], ack 1598, win 63920, length 0
11:12:14.731136 IP 117.18.237.29.http > MyPC.36522: Flags [.], ack 423, win 65535, length 0
11:12:14.782639 IP 192.168.43.1.domain > MyPC.37888: 46554 3/0/0 CNAME waws-prod-pn1-007.sip.azurewebsites.windows.net., CNAME waws-
prod-pn1-007.cloudapp.net., A 52.172.204.196 (164)
11:12:14.782968 IP localhost.domain > localhost.56010: 4186 4/0/1 CNAME pesuniversity.azurewebsites.net., CNAME waws-prod-pn1-007.si
p.azurewebsites.windows.net., CNAME waws-prod-pn1-007.cloudapp.net., A 52.172.204.196 (200)
11:12:14.783041 IP localhost.domain > localhost.45390: 16290 4/0/1 CNAME pesuniversity.azurewebsites.net., CNAME waws-prod-pn1-007.s
ip.azurewebsites.windows.net., CNAME waws-prod-pn1-007.cloudapp.net., A 52.172.204.196 (200)
11:12:14.783290 IP localhost.36308 > localhost.domain: 16607+ [1au] AAAA? www.pes.edu. (40)
11:12:14.783505 IP MyPC.59605 > 192.168.43.1.domain: 60575+ AAAA? waws-prod-pn1-007.cloudapp.net. (48)
11:12:14.783641 IP localhost.56810 > localhost.domain: 27998+ [1au] AAAA? www.pes.edu. (40)
11:12:14.986591 IP MyPC.32954 > a23-48-226-168.deploy.static.akamaitechnologies.com.http: Flags [.], ack 1778, win 64008, length 0
11:12:14.987419 IP a23-48-226-168.deploy.static.akamaitechnologies.com.http > MyPC.32954: Flags [.], ack 422, win 65535, length 0
11:12:15.070273 IP 192.168.43.1.domain > MyPC.37036: 28824 0/1/0 (100)
11:12:15.070766 IP localhost.domain > localhost.43261: 8229 0/0/1 (36)
```

Observation

Step 3: Understand the output format.

Step 4: To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using this command:

sudo tcpdump -i any -c5 icmp

```
laxmikant@MyPC:~$ sudo tcpdump -i any -c5 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
11:21:14.883878 IP localhost > localhost: ICMP localhost udp port 50799 unreachable, length 138
11:21:14.883905 IP localhost > localhost: ICMP localhost udp port 54902 unreachable, length 138
11:21:17.819423 IP localhost > localhost: ICMP localhost udp port 48234 unreachable, length 116
11:21:19.116390 IP localhost > localhost: ICMP localhost udp port 56092 unreachable, length 91
11:21:22.874463 IP localhost > localhost: ICMP localhost udp port 37803 unreachable, length 119
5 packets captured
10 packets received by filter
0 packets dropped by kernel
laxmikant@MyPC:~$
```

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

sudo tcpdump -i any -c6 -nn -A port 80

```
laxmikant@MyPC:~$ sudo tcpdump -i any -c6 -nn -A port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
11:24:13.578579 IP 10.0.2.15.38412 > 54.182.1.18.80: Flags [.], ack 371906012, win 63252, length 0
E..(.6@.@...
...6.....P.....*..P...C...
11:24:13.578652 IP 10.0.2.15.38414 > 54.182.1.18.80: Flags [.], ack 372034013, win 63315, length 0
E..(..@.@.H.
...6.....P.....,..P..SC...
11:24:13.579702 IP 54.182.1.18.80 > 10.0.2.15.38412: Flags [.], ack 1, win 65535, length 0
E..(m%..@...6...
....P...*......P...).........
11:24:13.579702 IP 54.182.1.18.80 > 10.0.2.15.38414: Flags [.], ack 1, win 65535, length 0
E..(m&..@...6...
....P...,......P.............
11:24:23.818651 IP 10.0.2.15.38412 > 54.182.1.18.80: Flags [.], ack 1, win 63252, length 0
E..(.7@.@...
...6.....P.....*..P...C...
11:24:23.818681 IP 10.0.2.15.38414 > 54.182.1.18.80: Flags [.], ack 1, win 63315, length 0
E..(..@.@.H.
...6.....P.....,..P..SC...
6 packets captured
6 packets received by filter
0 packets dropped by kernel
laxmikant@MyPC:~$
```

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

```
laxmikant@MyPC:~$ sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
10 packets captured
12 packets received by filter
0 packets dropped by kernel
laxmikant@MyPC:~$
```

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

```
laxmikant@MyPC:~$ sudo traceroute www.google.com
traceroute to www.google.com (142.250.193.132), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.695 ms  0.637 ms  0.626 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
laxmikant@MyPC:~$
```

Traceroute in windows: tracert www.google.com

```
Select Windows PowerShell (x86)
PS C:\Users\Laxmikant> tracert www.google.com

Tracing route to www.google.com [142.250.67.132]
over a maximum of 30 hops:

  1     5 ms     1 ms     1 ms  192.168.43.1
  2    58 ms    47 ms    69 ms  10.206.141.10
  3     *        *        *     Request timed out.
  4    40 ms     *       53 ms  10.206.143.189
  5     *        *        *     Request timed out.
  6   100 ms    53 ms    76 ms  182.79.239.197
  7    73 ms    47 ms    70 ms  72.14.208.234
  8    75 ms    77 ms    77 ms  216.239.51.91
  9    77 ms    75 ms    80 ms  108.170.253.120
 10    75 ms    70 ms    86 ms  72.14.232.34
 11    91 ms    75 ms    90 ms  108.170.248.161
 12   598 ms    92 ms    75 ms  142.250.227.71
 13    70 ms    55 ms    64 ms  bom12s06-in-f4.1e100.net [142.250.67.132]

Trace complete.
PS C:\Users\Laxmikant>
```

Step 3: To speed up the process, you can disable the mapping of IP addresses with hostnames by using the -*n* option

sudo traceroute -n www.google.com

```
laxmikant@MyPC:~$ sudo traceroute -n www.google.com
traceroute to www.google.com (142.250.193.132), 30 hops max, 60 byte packets
 1  10.0.2.2  0.647 ms  0.616 ms  0.599 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
laxmikant@MyPC:~$
```

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

sudo traceroute -I www.google.com

```
laxmikant@MyPC:~$ sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.182.100), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.959 ms  0.922 ms  0.916 ms
 2  192.168.43.1 (192.168.43.1)  26.302 ms  27.646 ms  27.640 ms
 3  10.206.141.10 (10.206.141.10)  241.913 ms  244.179 ms  244.169 ms
 4  * * *
 5  10.206.143.221 (10.206.143.221)  244.135 ms * *
 6  dsl-tn-dynamic-081.219.22.125.airtelbroadband.in (125.22.219.81)  246.623 ms  204.004 ms  203.948 ms
 7  * * *
 8  72.14.216.192 (72.14.216.192)  210.693 ms  68.126 ms  70.274 ms
 9  142.251.227.213 (142.251.227.213)  78.494 ms  76.130 ms  78.389 ms
10  142.251.55.241 (142.251.55.241)  78.372 ms  78.358 ms  68.851 ms
11  maa05s21-in-f4.1e100.net (142.250.182.100)  114.328 ms  110.615 ms  106.087 ms
laxmikant@MyPC:~$
```

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

sudo traceroute -T www.google.com

```
laxmikant@MyPC:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.182.100), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.753 ms  0.714 ms  0.702 ms
 2  maa05s21-in-f4.1e100.net (142.250.182.100)  152.098 ms  152.072 ms  152.062 ms
laxmikant@MyPC:~$
```

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.pesuacademy.com

```
laxmikant@MyPC:~$ nmap www.pesuacademy.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-25 18:28 IST
Nmap scan report for www.pesuacademy.com (15.206.0.45)
Host is up (0.15s latency).
Other addresses for www.pesuacademy.com (not scanned): 3.108.154.180 65.2.118.206
rDNS record for 15.206.0.45: ec2-15-206-0-45.ap-south-1.compute.amazonaws.com
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 29.29 seconds
laxmikant@MyPC:~$
```

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

nmap 15.206.0.45

```
laxmikant@MyPC:~$ nmap 15.206.0.45
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-25 18:42 IST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 0.05 seconds
laxmikant@MyPC:~$ nmap -Pn 15.206.0.45
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-25 18:42 IST
Nmap scan report for 15.206.0.45
Host is up (0.0041s latency).
All 1000 scanned ports on 15.206.0.45 are in ignored states.
Not shown: 1000 filtered tcp ports (host-unreach)

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
laxmikant@MyPC:~$
```

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

nmap 192.168.1.1 192.168.1.2 192.168.1.3

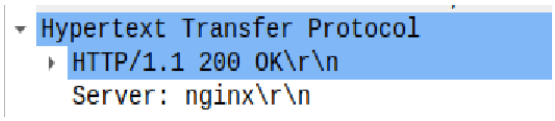
```
laxmikant@MyPC:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-25 18:44 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 0.06 seconds
laxmikant@MyPC:~$
```

Questions on above observations:

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

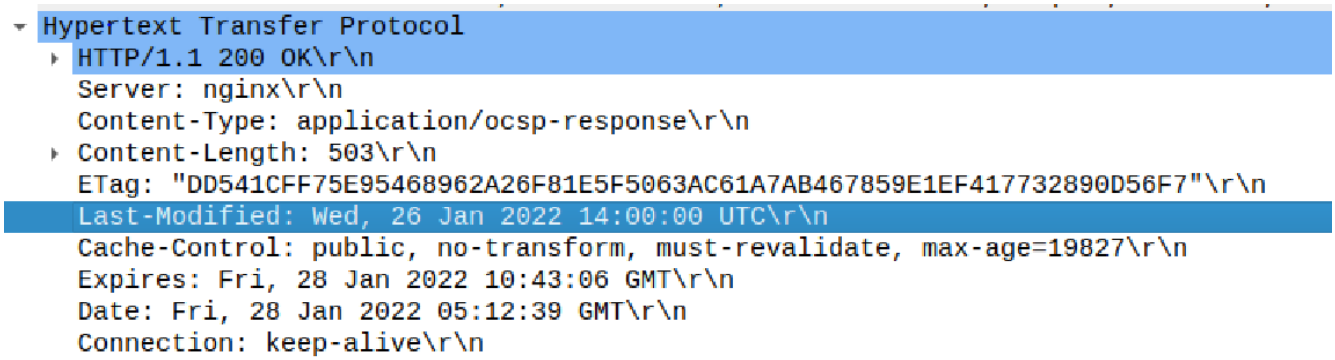
ANS:- HTTP version 1.1

The browser is running on HTTP version 1.1 .



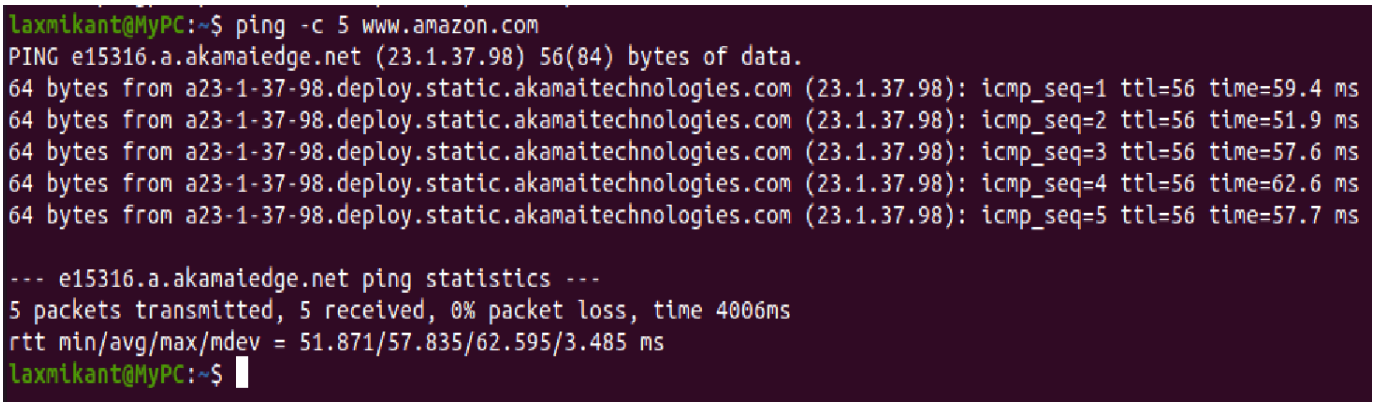
2) When was the HTML file that you are retrieving last modified at the server?

ANS:- The last modified field of the OCSP response gives the last modified time of the HTML file at the server.



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

ANS:- By using the command : ping -c ***no_of_packets IP_address/ URL***



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

ANS:- Using nmap command:

nmap -O -v ***IP_address/URL***

This will scan the network and gives the information about host apps and OS.