

# COMPUTER NETWORKS LAB – WEEK1

Name: TUSHAR Y S

SRN: PES1UG19CS545

## Task 1: Linux Interface Configuration

1.1 To display status of all active network interfaces.

Command- ip addr show

```
tushar@tushar:~$ 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 08:00:27:95:92:47 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 86348sec preferred_lft 86348sec
    inet6 fe80::33a5:4f63:f6e2:563e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
tushar@tushar:~$
```

Ip address table:

Interface Name	IP address(IPv4/IPv6)	MAC address
lo	127.0.0.1/::1	00:00:00:00:00:00
enp0s3	10.0.2.15/fe80::33a5:4f63:f6e2:563e	08:00:27:95:92:47

1.2 Assigning an IP address to an interface.

Command- sudo ip addr add 10.0.9.12/24 dev enp0s3

(Section I – 9, Roll No.:12)

```
tushar@tushar:~$ sudo ip addr add 10.0.9.12/24 dev enp0s3
[sudo] password for tushar:
tushar@tushar:~$ 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 08:00:27:95:92:47 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 86098sec preferred_lft 86098sec
    inet 10.0.9.12/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::33a5:4f63:f6e2:563e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
tushar@tushar:~$
```

### 1.3 To activate and deactivate a network interface.

#### 1.3.1 Deactivating an interface(enp0s3)

Command- sudo ifconfig enp0s3 down

```
tushar@tushar-VirtualBox:~$ sudo ifconfig enp0s3 down
tushar@tushar-VirtualBox:~$ 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 317 bytes 27371 (27.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 317 bytes 27371 (27.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
tushar@tushar-VirtualBox:~$
```

#### 1.3.2 Activating an interface(enp0s3)

Command- sudo ifconfig enp0s3 up

```
tushar@tushar:~$ sudo ifconfig enp0s3 up
tushar@tushar:~$ 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::33a5:4f63:f6e2:563e prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:95:92:47 txqueuelen 1000 (Ethernet)
    RX packets 139149 bytes 189994959 (189.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 33103 bytes 2054582 (2.0 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 321 bytes 27872 (27.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 321 bytes 27872 (27.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tushar@tushar:~$
```

1.4 To show the current neighbor table in kernel.

Command- ip neigh

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

## **Task 2: Ping PDU (Packet Data Units) Capture**

## Command- ping 10.0.9.12

```
tushar@tushar:~$ ping 10.0.9.12
PING 10.0.9.12 (10.0.9.12) 56(84) bytes of data.
64 bytes from 10.0.9.12: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 10.0.9.12: icmp_seq=2 ttl=64 time=0.084 ms
64 bytes from 10.0.9.12: icmp_seq=3 ttl=64 time=0.112 ms
64 bytes from 10.0.9.12: icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from 10.0.9.12: icmp_seq=5 ttl=64 time=0.057 ms
64 bytes from 10.0.9.12: icmp_seq=6 ttl=64 time=0.110 ms
64 bytes from 10.0.9.12: icmp_seq=7 ttl=64 time=0.451 ms
64 bytes from 10.0.9.12: icmp_seq=8 ttl=64 time=0.288 ms
64 bytes from 10.0.9.12: icmp_seq=9 ttl=64 time=0.061 ms
64 bytes from 10.0.9.12: icmp_seq=10 ttl=64 time=0.089 ms
64 bytes from 10.0.9.12: icmp_seq=11 ttl=64 time=0.240 ms
64 bytes from 10.0.9.12: icmp_seq=12 ttl=64 time=0.109 ms
64 bytes from 10.0.9.12: icmp_seq=13 ttl=64 time=0.141 ms
64 bytes from 10.0.9.12: icmp_seq=14 ttl=64 time=0.122 ms
64 bytes from 10.0.9.12: icmp_seq=15 ttl=64 time=0.080 ms
64 bytes from 10.0.9.12: icmp_seq=16 ttl=64 time=0.106 ms
64 bytes from 10.0.9.12: icmp_seq=17 ttl=64 time=0.103 ms
64 bytes from 10.0.9.12: icmp_seq=18 ttl=64 time=0.168 ms
64 bytes from 10.0.9.12: icmp_seq=19 ttl=64 time=0.132 ms
64 bytes from 10.0.9.12: icmp_seq=20 ttl=64 time=0.090 ms
64 bytes from 10.0.9.12: icmp_seq=21 ttl=64 time=0.057 ms
64 bytes from 10.0.9.12: icmp_seq=22 ttl=64 time=0.105 ms
64 bytes from 10.0.9.12: icmp_seq=23 ttl=64 time=0.071 ms
64 bytes from 10.0.9.12: icmp_seq=24 ttl=64 time=0.196 ms
64 bytes from 10.0.9.12: icmp_seq=25 ttl=64 time=0.056 ms
64 bytes from 10.0.9.12: icmp_seq=26 ttl=64 time=0.112 ms
64 bytes from 10.0.9.12: icmp_seq=27 ttl=64 time=0.045 ms
64 bytes from 10.0.9.12: icmp_seq=28 ttl=64 time=0.057 ms
64 bytes from 10.0.9.12: icmp_seq=29 ttl=64 time=0.062 ms
64 bytes from 10.0.9.12: icmp_seq=30 ttl=64 time=0.048 ms
64 bytes from 10.0.9.12: icmp_seq=31 ttl=64 time=0.097 ms
64 bytes from 10.0.9.12: icmp_seq=32 ttl=64 time=0.099 ms
```

Observation:

TTL	64
Protocol used by ping	ICMP
Time	In the order of $10^{-2}$ ms

## Echo Request Packet:

Wireshark · Packet 1 · any

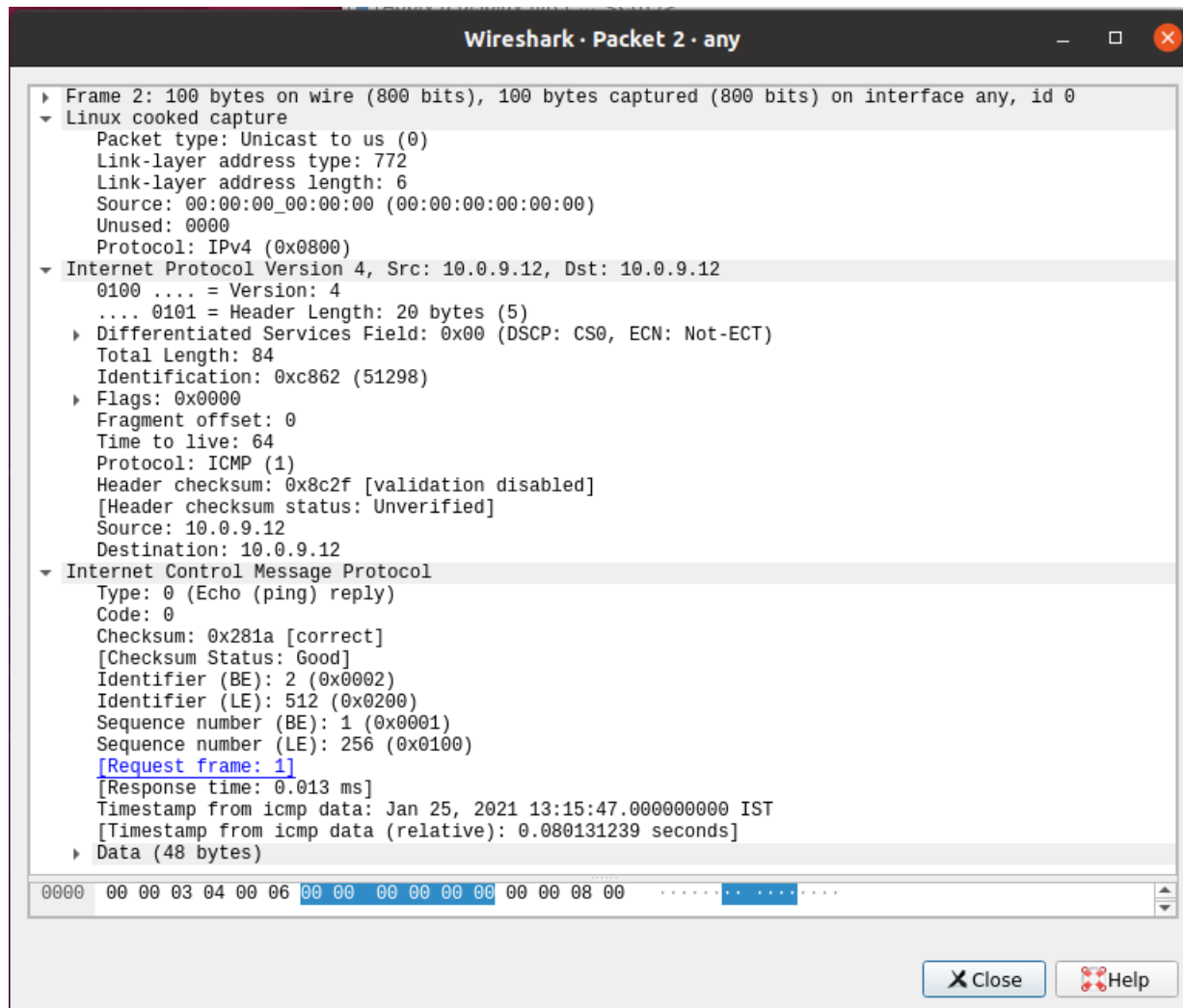
▶ Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0

- 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: 10.0.9.12, Dst: 10.0.9.12
  - 0100 .... = Version: 4
  - .... 0101 = Header Length: 20 bytes (5)
  - ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  - Total Length: 84
  - Identification: 0xc861 (51297)
  - ▶ Flags: 0x4000, Don't fragment
  - Fragment offset: 0
  - Time to live: 64
  - Protocol: ICMP (1)
  - Header checksum: 0x4c30 [validation disabled]
  - [Header checksum status: Unverified]
  - Source: 10.0.9.12
  - Destination: 10.0.9.12
- Internet Control Message Protocol
  - Type: 8 (Echo (ping) request)
  - Code: 0
  - Checksum: 0x201a [correct]
  - [Checksum Status: Good]
  - Identifier (BE): 2 (0x0002)
  - Identifier (LE): 512 (0x0200)
  - Sequence number (BE): 1 (0x0001)
  - Sequence number (LE): 256 (0x0100)
  - [\[Response frame: 2\]](#)
  - Timestamp from icmp data: Jan 25, 2021 13:15:47.000000000 IST
  - [Timestamp from icmp data (relative): 0.080118574 seconds]
  - ▶ Data (48 bytes)

0000	00 00 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....
0010	45 00 00 54 c8 61 40 00	40 01 4c 30 0a 00 09 0c	E..T.a@. @.L0...

Close Help

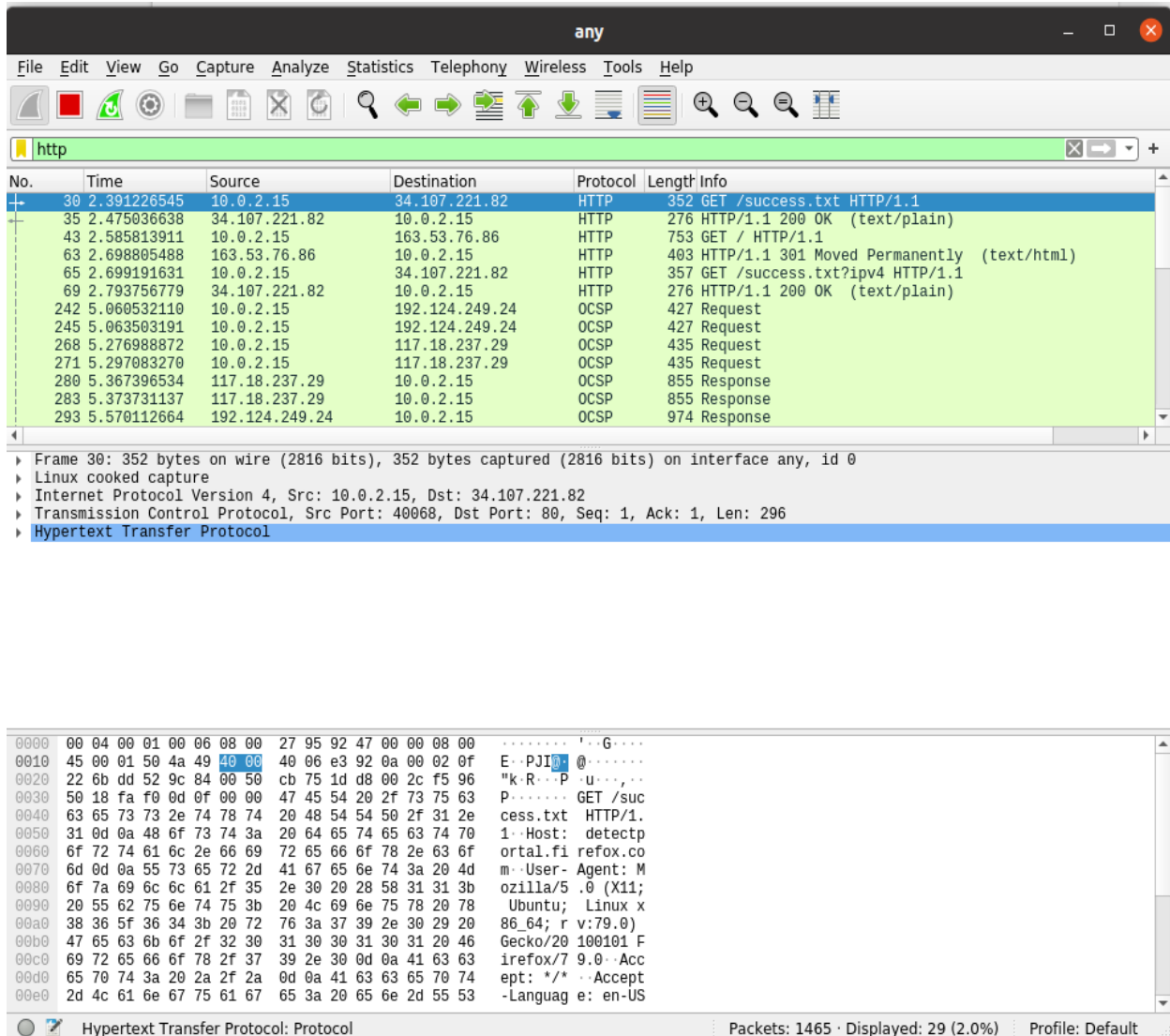
## Echo Response Packet:



Details	First Echo Request	First Echo Reply
Frame Number	1	2
Source IP address	10.0.9.12	10.0.9.12
Destination IP address	10.0.9.12	10.0.9.12
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

3.1 Upon browsing [www.flipkart.com](http://www.flipkart.com), and selecting 'http' in wireshark's filter toolbar:



The screenshot shows the Wireshark interface with the 'http' filter applied to the packet list. The packet list displays the following data:

No.	Time	Source	Destination	Protocol	Length	Info
30	2.391226545	10.0.2.15	34.107.221.82	HTTP	352	GET /success.txt HTTP/1.1
35	2.475036638	34.107.221.82	10.0.2.15	HTTP	276	HTTP/1.1 200 OK (text/plain)
43	2.585813911	10.0.2.15	163.53.76.86	HTTP	753	GET / HTTP/1.1
63	2.698805488	163.53.76.86	10.0.2.15	HTTP	403	HTTP/1.1 301 Moved Permanently (text/html)
65	2.699191631	10.0.2.15	34.107.221.82	HTTP	357	GET /success.txt?ipv4 HTTP/1.1
69	2.793756779	34.107.221.82	10.0.2.15	HTTP	276	HTTP/1.1 200 OK (text/plain)
242	5.060532110	10.0.2.15	192.124.249.24	OCSP	427	Request
245	5.063503191	10.0.2.15	192.124.249.24	OCSP	427	Request
268	5.276988872	10.0.2.15	117.18.237.29	OCSP	435	Request
271	5.297083270	10.0.2.15	117.18.237.29	OCSP	435	Request
280	5.367396534	117.18.237.29	10.0.2.15	OCSP	855	Response
283	5.373731137	117.18.237.29	10.0.2.15	OCSP	855	Response
293	5.570112664	192.124.249.24	10.0.2.15	OCSP	974	Response

The details pane shows the selected packet (Frame 30) with the following information:

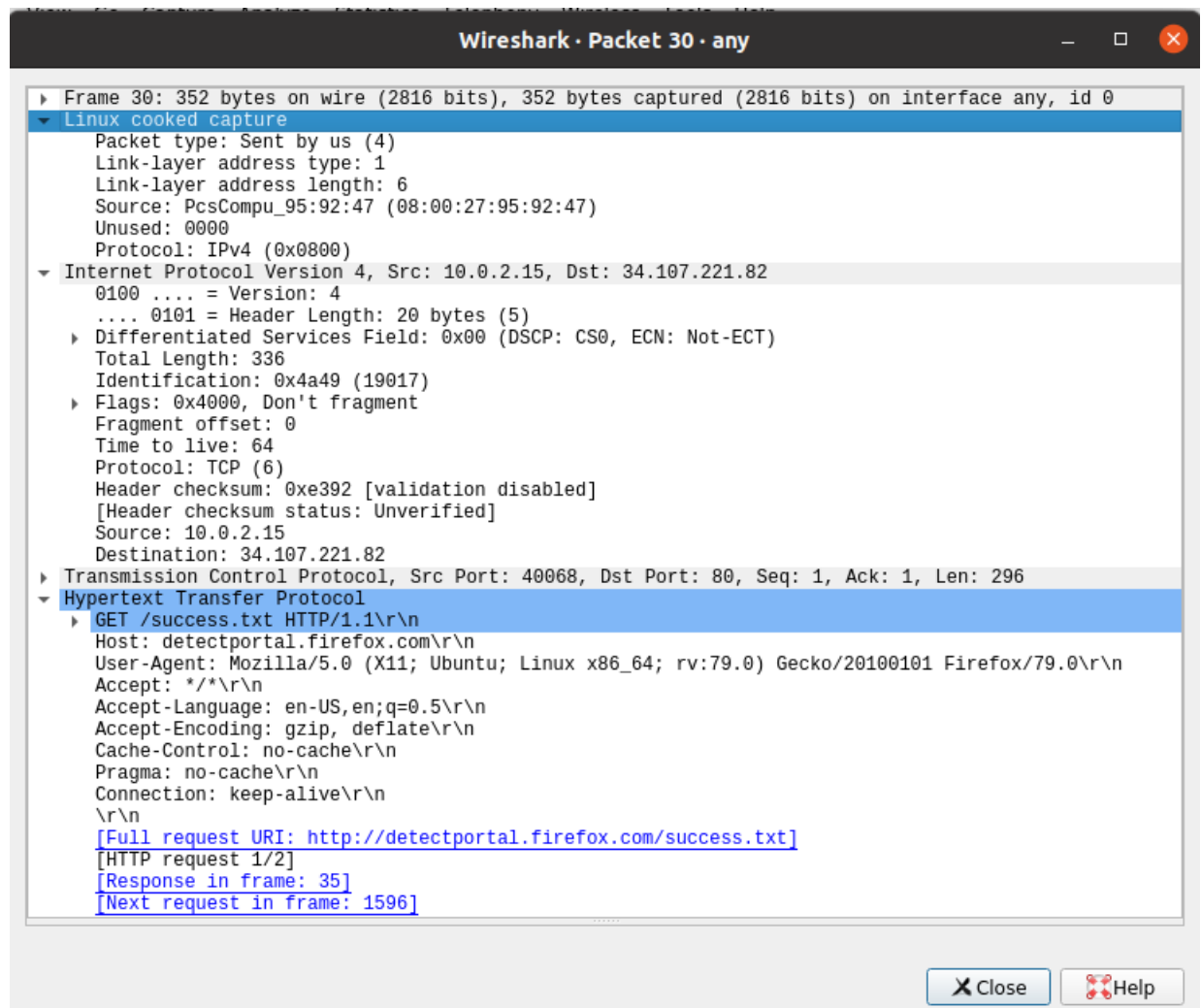
- Frame 30: 352 bytes on wire (2816 bits), 352 bytes captured (2816 bits) on interface any, id 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.2.15, Dst: 34.107.221.82
- Transmission Control Protocol, Src Port: 40068, Dst Port: 80, Seq: 1, Ack: 1, Len: 296
- Hypertext Transfer Protocol

The packet bytes pane shows the raw data of the selected packet, including the HTTP request line: GET /success.txt HTTP/1.1.

At the bottom, the status bar indicates: Packets: 1465 · Displayed: 29 (2.0%) Profile: Default

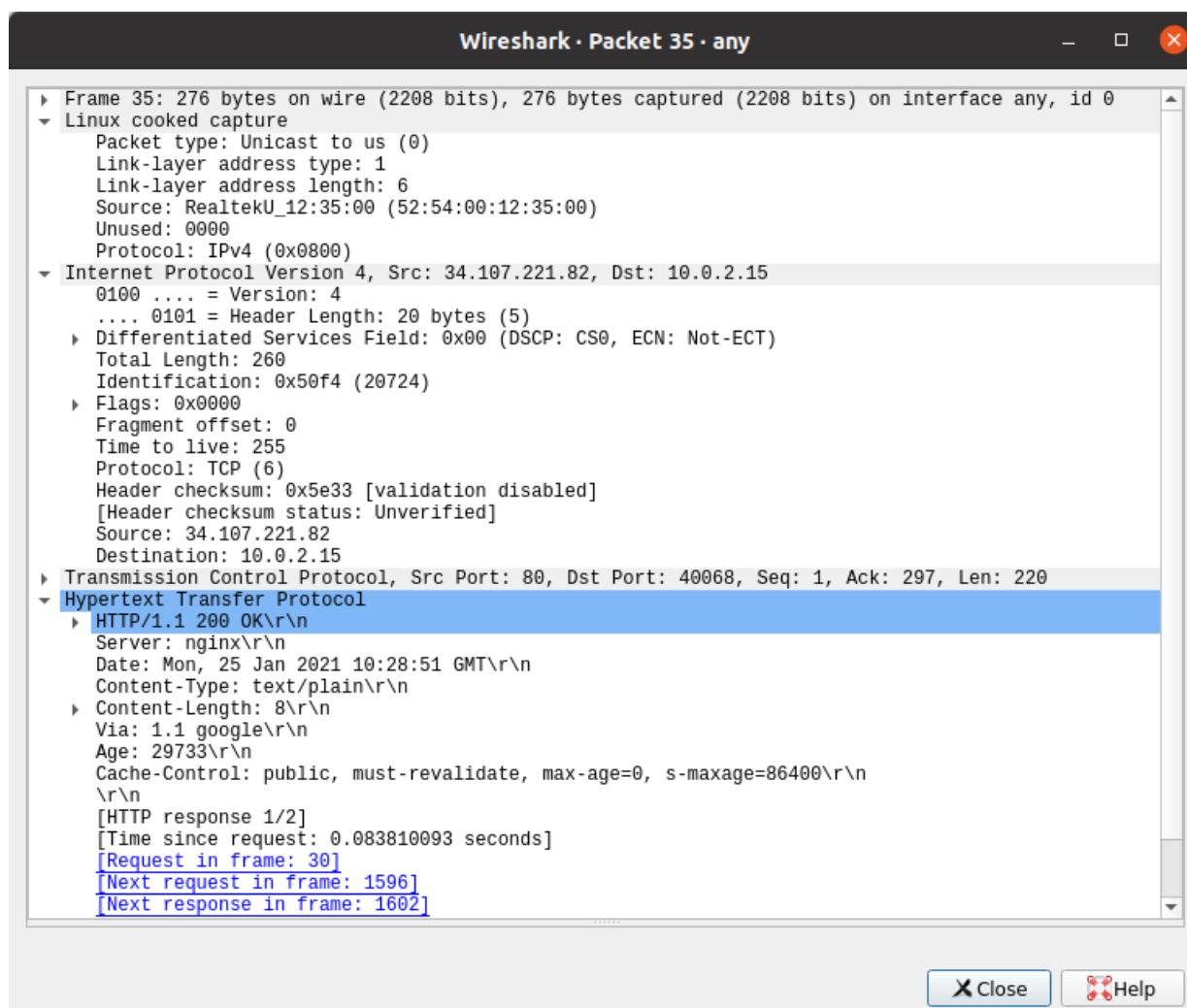
## 3.2 Echo Request and Reply:

### 3.2.1 Echo Request Packet:



### 3.2.2 Echo Response Packet:





### 3.2.3

Details	First Echo Request	First Echo Reply
Frame Number	30	35
Source Port	40068	80
Destination Port	80	40068
Source IP Address	10.0.2.15	34.107.221.82
Destination IP Address	34.107.221.82	10.0.2.15
Source Ethernet Address	08:00:27:95:92:47	52:54:00:12:35:00
Destination Ethernet Address	52:54:00:12:35:00	08:00:27:95:92:47

## Wireshark's follow TCP Stream:

The image shows the 'Follow TCP Stream' window in Wireshark, titled 'Wireshark · Follow TCP Stream (tcp.stream eq 1) · any'. The window displays the details of a selected TCP stream, which is an HTTP 301 response from www.flipkart.com. The status bar at the bottom indicates '1 client pkt, 1 server pkt, 1 turn'.

**GET / HTTP/1.1**  
Host: www.flipkart.com  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:79.0) Gecko/20100101 Firefox/79.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Cookie: SN=VIF93D90BC382D4AE3A32D771B0EBBC17B.TOKB45CC7F0CE764D3A8B3183B1B102763A.1611599351.L0;AMCV\_17EB401053DAF4840A490D4C%40AdobeOrg=-227196251%7CMCIDTS%7C18652%7CMCMID%7C70854714864010822691603881039682784308%7CMCAAMLH-1612117820%7C12%7CMCAAMB-1612117820%7C6G1ynYcLPuiQxYZrsz\_pkqfLG9yMXBpb2zX5dvJdYQJzPXImdj0y%7CMCOPTOUT-1611520222s%7CNONE%7MCAID%7CNONE  
Upgrade-Insecure-Requests: 1

**HTTP/1.1 301 Moved Permanently**  
Server: nginx  
Date: Mon, 25 Jan 2021 18:44:25 GMT  
Content-Type: text/html  
Content-Length: 178  
Location: https://www.flipkart.com/

**<html>**  
**<head><title>301 Moved Permanently</title></head>**  
**<body bgcolor="white">**  
**<center><h1>301 Moved Permanently</h1></center>**  
**<hr><center>nginx</center>**  
**</body>**  
**</html>**

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (1,044 bytes) Show and save data as ASCII Stream 1

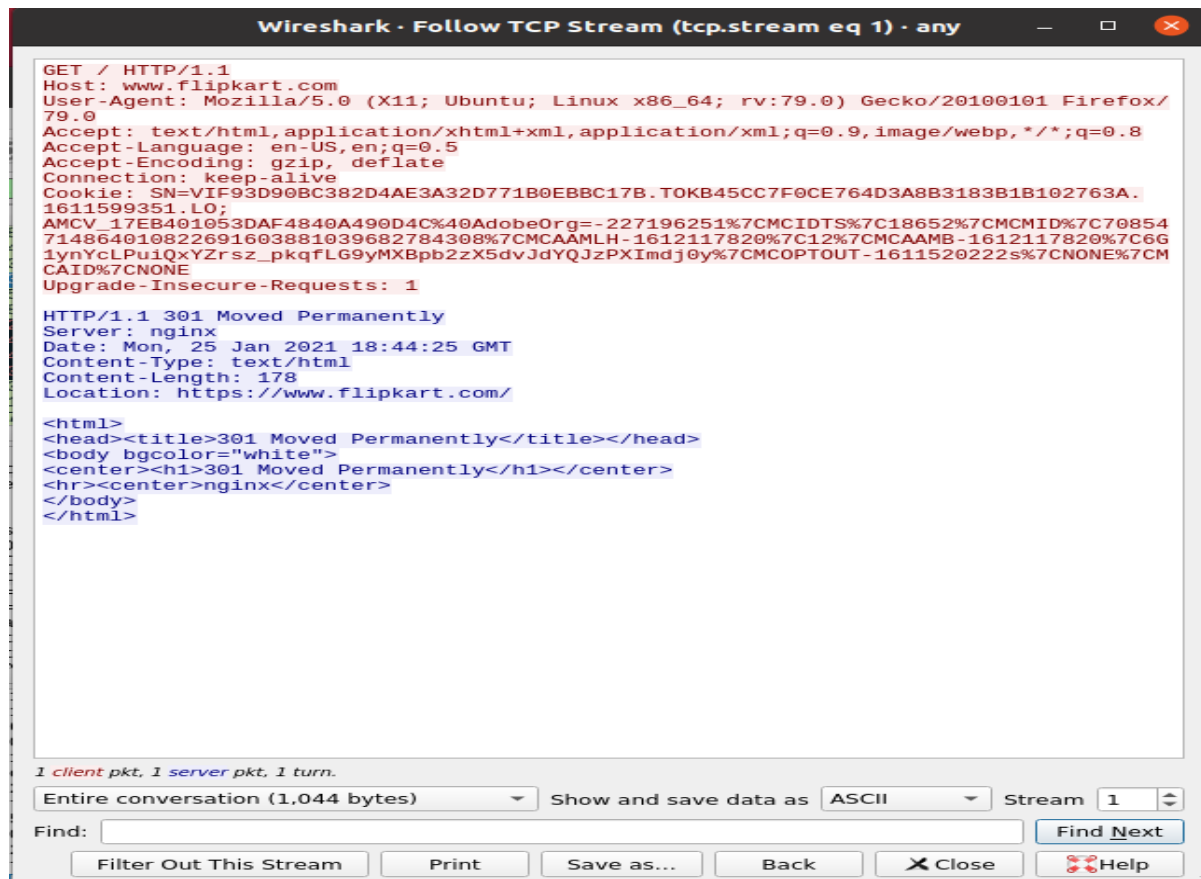
Find: Find Next

Filter Out This Stream Print Save as... Back X Close Help

### 3.3 HTTP Request and Response:

HTTP Request		HTTP Response	
Get	GET/HTTP/1.1\r\n	Server	nginx
Host	<a href="http://www.flipkart.com">www.flipkart.com</a>	Content-Type	Text/plain\r\n
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0\r\n	Date	Mon, 25 Jan 2021 10:28:51 GMT\r\n
Accept-Language	en-US,en;q=0.5\r\n	Location	<a href="https://www.flipkart.com/">https://www.flipkart.com/</a>
Accept-Encoding	gzip,deflate\r\n	Content-Length	178
Connection	keep-alive\r\n	Connection	Keep-alive

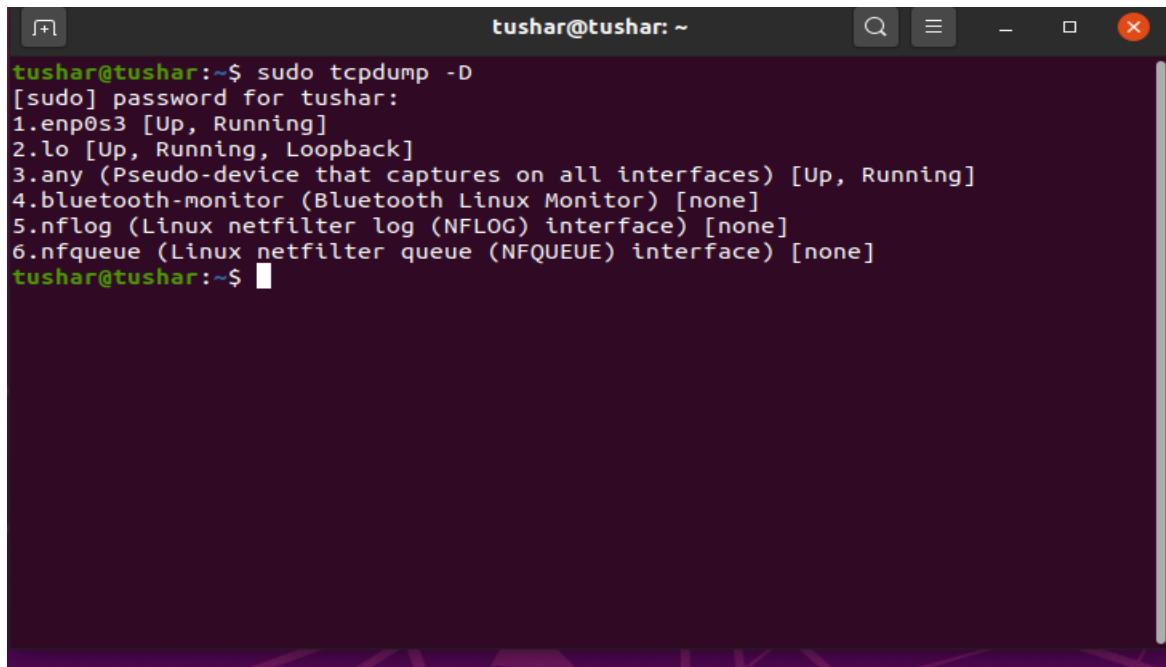
### 3.4 Wireshark's follow TCP Stream:



## Task 4- Capturing packets with tcpdump

### 4.1 Interfaces available for Capture:

Command- `sudo tcpdump -D`



```
tushar@tushar:~$ sudo tcpdump -D
[sudo] password for tushar:
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]
tushar@tushar:~$
```

### 4.2 Capturing all packets in any interface:

Command- `sudo tcpdump -i any`

```
18:26:00.588801 IP localhost.domain > localhost.52626: 60173 1/0/1 PTR maa05s09-in-f4.1e100.net. (95)
18:26:00.591613 IP maa05s14-in-f2.1e100.net.https > tushar.51044: Flags [P.], seq 1432:1514, ack 197, win 65535, length 82
18:26:00.591648 IP tushar.51044 > maa05s14-in-f2.1e100.net.https: Flags [.], ack 1514, win 63020, length 0
18:26:00.594155 IP maa05s14-in-f2.1e100.net.https > tushar.51044: Flags [P.], seq 1514:1584, ack 197, win 65535, length 70
18:26:00.594190 IP tushar.51044 > maa05s14-in-f2.1e100.net.https: Flags [.], ack 1584, win 63020, length 0
18:26:00.594534 IP tushar.51044 > maa05s14-in-f2.1e100.net.https: Flags [P.], seq 197:236, ack 1584, win 63020, length 39
18:26:00.594851 IP maa05s14-in-f2.1e100.net.https > tushar.51044: Flags [.], ack 236, win 65535, length 0
18:26:00.680488 IP maa05s09-in-f4.1e100.net > tushar: ICMP echo reply, id 2, seq 24, length 64
18:26:01.589970 IP tushar > maa05s09-in-f4.1e100.net: ICMP echo request, id 2, seq 25, length 64
18:26:01.691816 IP maa05s09-in-f4.1e100.net > tushar: ICMP echo reply, id 2, seq 25, length 64
18:26:02.591494 IP tushar > maa05s09-in-f4.1e100.net: ICMP echo request, id 2, seq 26, length 64
18:26:02.682121 IP maa05s09-in-f4.1e100.net > tushar: ICMP echo reply, id 2, seq 26, length 64
18:26:03.613432 IP tushar > maa05s09-in-f4.1e100.net: ICMP echo request, id 2, seq 27, length 64
18:26:03.692053 IP maa05s09-in-f4.1e100.net > tushar: ICMP echo reply, id 2, seq 27, length 64
18:26:04.251394 IP tushar.43902 > 82.221.107.34.bc.googleusercontent.com.http: Flags [.], ack 58496221, win 64021, length 0
18:26:04.252367 IP localhost.47508 > localhost.domain: 39797+ [1au] PTR? 82.221.107.34.in-addr.arpa. (55)
18:26:04.252561 IP 82.221.107.34.bc.googleusercontent.com.http > tushar.43902: Flags [.], ack 1, win 65535, length 0
18:26:04.253580 IP tushar.44807 > dns.google.domain: 52624+ [1au] PTR? 82.221.107.34.in-addr.arpa. (55)
18:26:04.445566 IP tushar.43904 > 82.221.107.34.bc.googleusercontent.com.http: Flags [.], ack 58560221, win 64021, length 0
18:26:04.446617 IP 82.221.107.34.bc.googleusercontent.com.http > tushar.43904: Flags [.], ack 1, win 65535, length 0
18:26:04.476587 IP dns.google.domain > tushar.44807: 52624 1/0/1 PTR 82.221.107.34.bc.googleusercontent.com. (107)
18:26:04.615356 IP tushar > maa05s09-in-f4.1e100.net: ICMP echo request, id 2, seq 28, length 64
18:26:04.709626 IP maa05s09-in-f4.1e100.net > tushar: ICMP echo reply, id 2, seq 28, length 64
18:26:04.952615 IP tushar.37344 > 117.18.237.29.http: Flags [.], ack 59649600, win 63920, length 0
18:26:04.953320 IP localhost.43213 > localhost.domain: 48412+ [1au] PTR? 29.237.18.117.in-addr.arpa. (55)
18:26:04.953673 IP 117.18.237.29.http > tushar.37344: Flags [.], ack 1, win 65535, length 0
```

### 4.3 To filter packets based on protocol(ex:icmp):

Command- `sudo tcpdump -i any -c5 icmp`

```
tushar@tushar:~$ 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
18:29:32.691488 IP tushar > maa05s09-in-f4.1e100.net: ICMP echo request, id 2, seq 234, length 64
18:29:32.763592 IP maa05s09-in-f4.1e100.net > tushar: ICMP echo reply, id 2, seq 234, length 64
18:29:33.692368 IP tushar > maa05s09-in-f4.1e100.net: ICMP echo request, id 2, seq 235, length 64
18:29:33.769779 IP maa05s09-in-f4.1e100.net > tushar: ICMP echo reply, id 2, seq 235, length 64
18:29:34.701080 IP tushar > maa05s09-in-f4.1e100.net: ICMP echo request, id 2, seq 236, length 64
5 packets captured
5 packets received by filter
0 packets dropped by kernel
tushar@tushar:~$
```

### 4.4 To check the packet content:

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

```
tushar@tushar:~$ sudo tcpdump -i any -c10 -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
18:34:31.932594 IP 10.0.2.15.59848 > 34.122.121.32.80: Flags [S], seq 3568749603, win 64240, options [mss 1460,sackOK,TS val 660127072 ecr 0,nop,wscale 7], length 0
E..<t.@.0./
...zy ...P...#.....
'X'.....
18:34:32.326403 IP 34.122.121.32.80 > 10.0.2.15.59848: Flags [S.], seq 114688001, ack 3568749604, win 65535, options [mss 1460], length 0
E.../.@...zy
....P.....$`...^.....
18:34:32.326521 IP 10.0.2.15.59848 > 34.122.121.32.80: Flags [.], ack 1, win 64240, length 0
E..(t.@.0.B
...zy ...P...$....P.....
18:34:32.326932 IP 10.0.2.15.59848 > 34.122.121.32.80: Flags [P.], seq 1:88, ack 1, win 64240, length 87: HTTP: GET / HTTP/1.1
E...t.@.0...
...zy ...P...$....P.....GET / HTTP/1.1
Host: connectivity-check.ubuntu.com
Accept: */*
Connection: close

18:34:32.327650 IP 34.122.121.32.80 > 10.0.2.15.59848: Flags [.], ack 88, win 65535, length 0
E..(.0.@...zy
....P.....{P...v.....
18:34:32.966365 IP 34.122.121.32.80 > 10.0.2.15.59848: Flags [P.], seq 1:149, ack 88, win 65535, length 148: HTTP: HTTP/1.1 204 No Content
E...1.@.0.b"zy
....P.....{P....Z..HTTP/1.1 204 No Content
Date: Mon, 25 Jan 2021 13:04:32 GMT
Server: Apache/2.4.18 (Ubuntu)
X-NetworkManager-Status: online
Connection: close
```

```

X-NetworkManager-Status: online
Connection: close

18:34:32.966437 IP 10.0.2.15.59848 > 34.122.121.32.80: Flags [.], ack 149, win 64092, length 0
E..(t.@.@.
...zy ...P...{....P..}....
18:34:32.966511 IP 34.122.121.32.80 > 10.0.2.15.59848: Flags [F.], seq 149, ack 88, win 65535, length 0
E..(.2.@...zy
....P.....{P...US.....
18:34:32.966874 IP 10.0.2.15.59848 > 34.122.121.32.80: Flags [F.], seq 88, ack 150, win 64091, length 0
E..(t.@.@.?
...zy ...P...{....P..}....
18:34:32.967762 IP 34.122.121.32.80 > 10.0.2.15.59848: Flags [.], ack 89, win 65535, length 0
E..(.3.@...zy
....P.....|P...ur.....
10 packets captured
10 packets received by filter
0 packets dropped by kernel
tushar@tushar:~$

```

## 4.5 To save packets to a file:

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

```

tushar@tushar:~$ 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
10 packets received by filter
0 packets dropped by kernel
tushar@tushar:~$

```

webserver.pcap file:

The screenshot displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for file operations, capture control, and analysis. The main window is divided into three panes:

- Packet List:** Shows 10 captured packets. The first packet (No. 1) is a SYN packet from 10.0.2.15 to 35.224.170.84 on port 80. Subsequent packets include an ACK, an HTTP GET request, and several HTTP responses (204 No Content, 200 OK).
- Packet Details:** For the selected packet (Frame 1), it shows the Ethernet II header, Internet Protocol Version 4 details (Source: 10.0.2.15, Destination: 35.224.170.84), and the Transmission Control Protocol details (Source Port: 41836, Destination Port: 80, Seq: 0, Len: 0).
- Packet Bytes:** Shows the raw packet data in hexadecimal and ASCII format.

## Task 5- Perform Traceroute Checks

### 5.1 Running the traceroute:

Command- sudo traceroute [www.google.com](http://www.google.com)

```
tushar@tushar:~$ sudo traceroute www.google.com
traceroute to www.google.com (142.250.76.68), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.1) 2.561 ms 2.410 ms 2.351 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 * * *
tushar@tushar:~$
```

Destination address of google.com – 142.250.76.68

No. of hops – 30 max hops

### 5.2 Disabling the mapping of ip addresses with host names:

Command- sudo traceroute -n [www.google.com](http://www.google.com)



```
tushar@tushar:~$ sudo traceroute -n www.google.com
traceroute to www.google.com (216.58.200.132), 30 hops max, 60 byte packets
 1  10.0.2.1  0.579 ms  0.377 ms  0.415 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  * * *
```

### 5.3 Using -I option so that traceroute uses ICMP protocol:

Command- sudo traceroute -I [www.google.com](http://www.google.com)

```
tushar@tushar:~$ sudo traceroute -I www.google.com
traceroute to www.google.com (216.58.200.132), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.1)  0.243 ms  0.208 ms  0.196 ms
 2  192.168.43.1 (192.168.43.1)  11.282 ms  11.286 ms  11.276 ms
 3  * * *
 4  10.72.203.242 (10.72.203.242)  106.745 ms  10.72.203.226 (10.72.203.226)  106.731 ms  10.72.203.242 (10.72.203.242)  106.917 ms
 5  192.168.65.252 (192.168.65.252)  112.624 ms  118.451 ms  192.168.65.250 (192.168.65.250)  118.460 ms
 6  192.168.65.253 (192.168.65.253)  118.817 ms  192.168.65.249 (192.168.65.249)  60.557 ms  192.168.65.253 (192.168.65.253)  72.154 ms
 7  172.26.74.20 (172.26.74.20)  71.945 ms  43.215 ms  55.515 ms
 8  172.26.77.242 (172.26.77.242)  70.373 ms  76.165 ms  76.188 ms
 9  192.168.65.138 (192.168.65.138)  76.802 ms  119.807 ms  126.821 ms
10  192.168.65.141 (192.168.65.141)  126.730 ms  126.746 ms  126.680 ms
11  172.26.29.107 (172.26.29.107)  128.071 ms  128.067 ms  85.781 ms
12  172.26.29.107 (172.26.29.107)  113.010 ms  98.150 ms  97.123 ms
13  10.70.80.197 (10.70.80.197)  97.743 ms  102.411 ms  59.570 ms
14  10.70.80.225 (10.70.80.225)  66.014 ms  71.672 ms  72.269 ms
15  74.125.48.26 (74.125.48.26)  70.993 ms  71.173 ms  75.467 ms
16  74.125.242.129 (74.125.242.129)  75.427 ms  64.580 ms  69.603 ms
17  216.239.54.197 (216.239.54.197)  64.801 ms  70.845 ms  65.624 ms
18  maa05s10-in-f4.1e100.net (216.58.200.132)  70.659 ms  57.151 ms  52.331 ms
tushar@tushar:~$
```

### 5.4 To test a TCP connection to gather data more relevant to web server:

Command- sudo traceroute -T [www.google.com](http://www.google.com)

```
tushar@tushar:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (216.58.200.132), 30 hops max, 60 byte packets
 1  maa05s10-in-f4.1e100.net (216.58.200.132)  78.687 ms  91.008 ms  91.103 ms
tushar@tushar:~$
```



## **Task 6- Explore an entire network for information (Nmap)**

6.1 To scan a host using its hostname:

Command- nmap [www.pes.edu](http://www.pes.edu)

```
tushar@tushar:~$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-25 23:33 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.33s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 125.62 seconds
tushar@tushar:~$
```

6.2 To scan a host using its IP address:

Command- nmap 163.53.78.128

```
tushar@tushar:~$ nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-25 23:38 IST
Nmap scan report for 163.53.78.128
Host is up (0.32s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 192.30 seconds
tushar@tushar:~$
```

6.3 Scanning multiple IP address or subnet (IPv4):

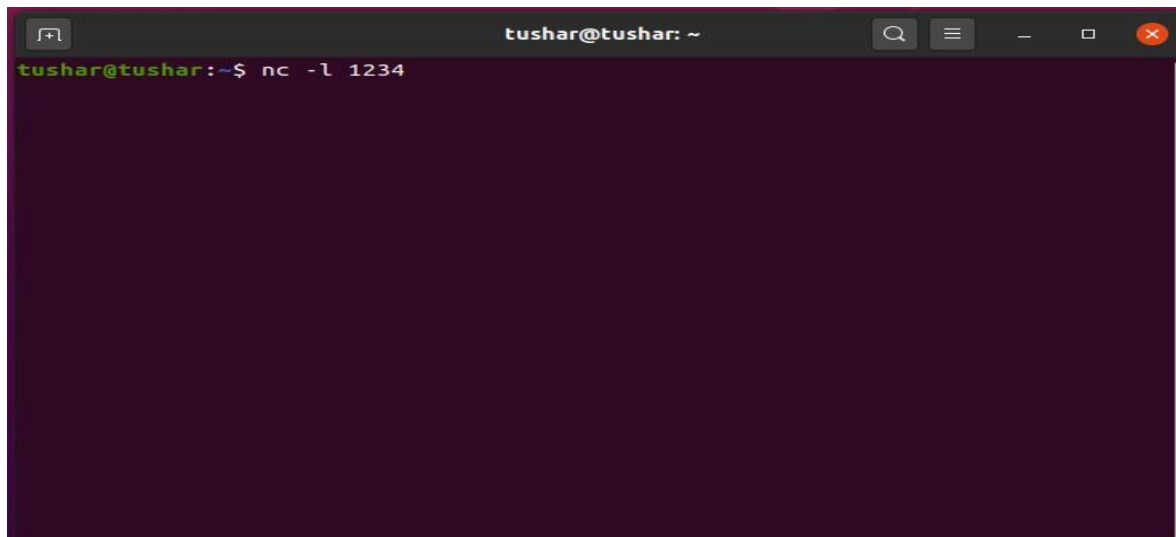
Command- nmap 192.168.1.1 192.168.1.2 192.168.1.3

```
tushar@tushar:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-25 23:44 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.14 seconds
tushar@tushar:~$
```

## **Task 7 a)- Netcat as Chat tool**

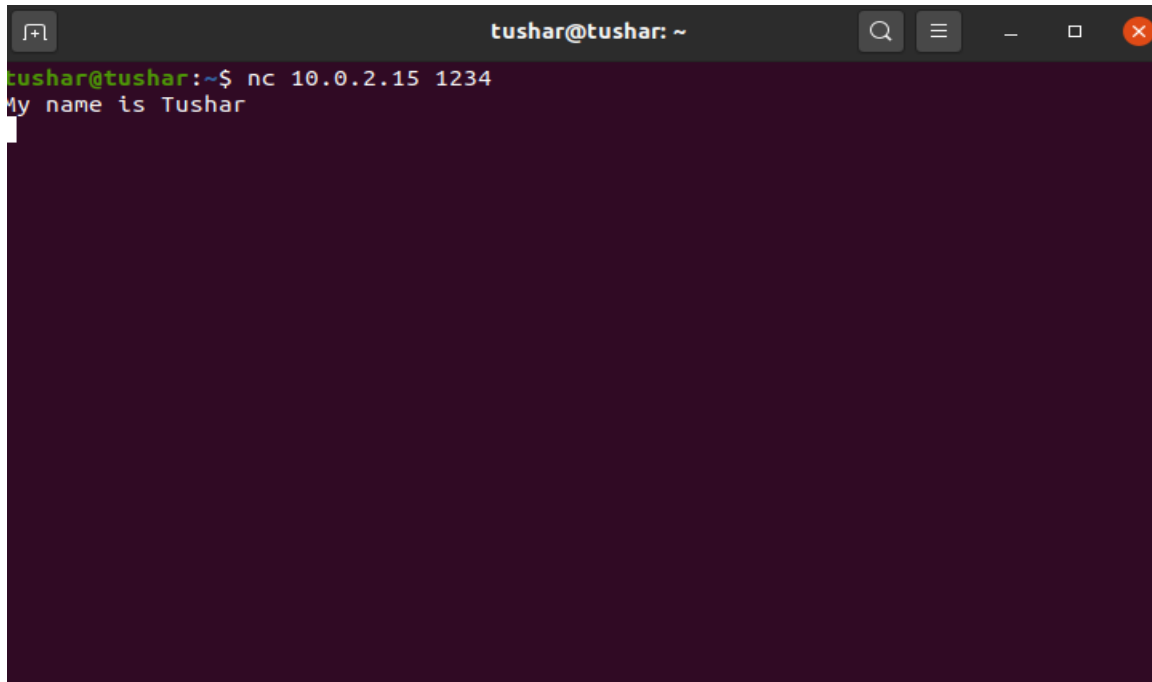
7a.1 Intra system communication:

Command on server's terminal: nc -l 1234

A terminal window titled 'tushar@tushar: ~' with a dark purple background. The command 'nc -l 1234' has been entered at the prompt 'tushar@tushar:~\$'.

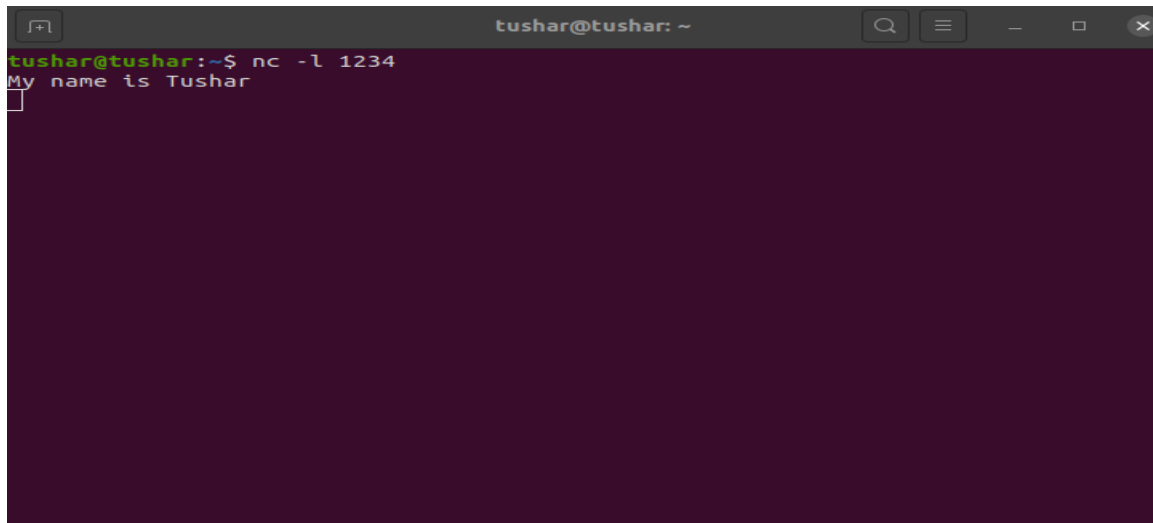
```
tushar@tushar:~$ nc -l 1234
```

Command on Client's terminal: nc 10.0.2.15 1234

A terminal window titled 'tushar@tushar: ~' with a dark purple background. The command 'nc 10.0.2.15 1234' has been entered at the prompt 'tushar@tushar:~\$'. The output 'My name is Tushar' is displayed on the next line.

```
tushar@tushar:~$ nc 10.0.2.15 1234
My name is Tushar
```

Whatever is typed on client's terminal is appearing on server side.

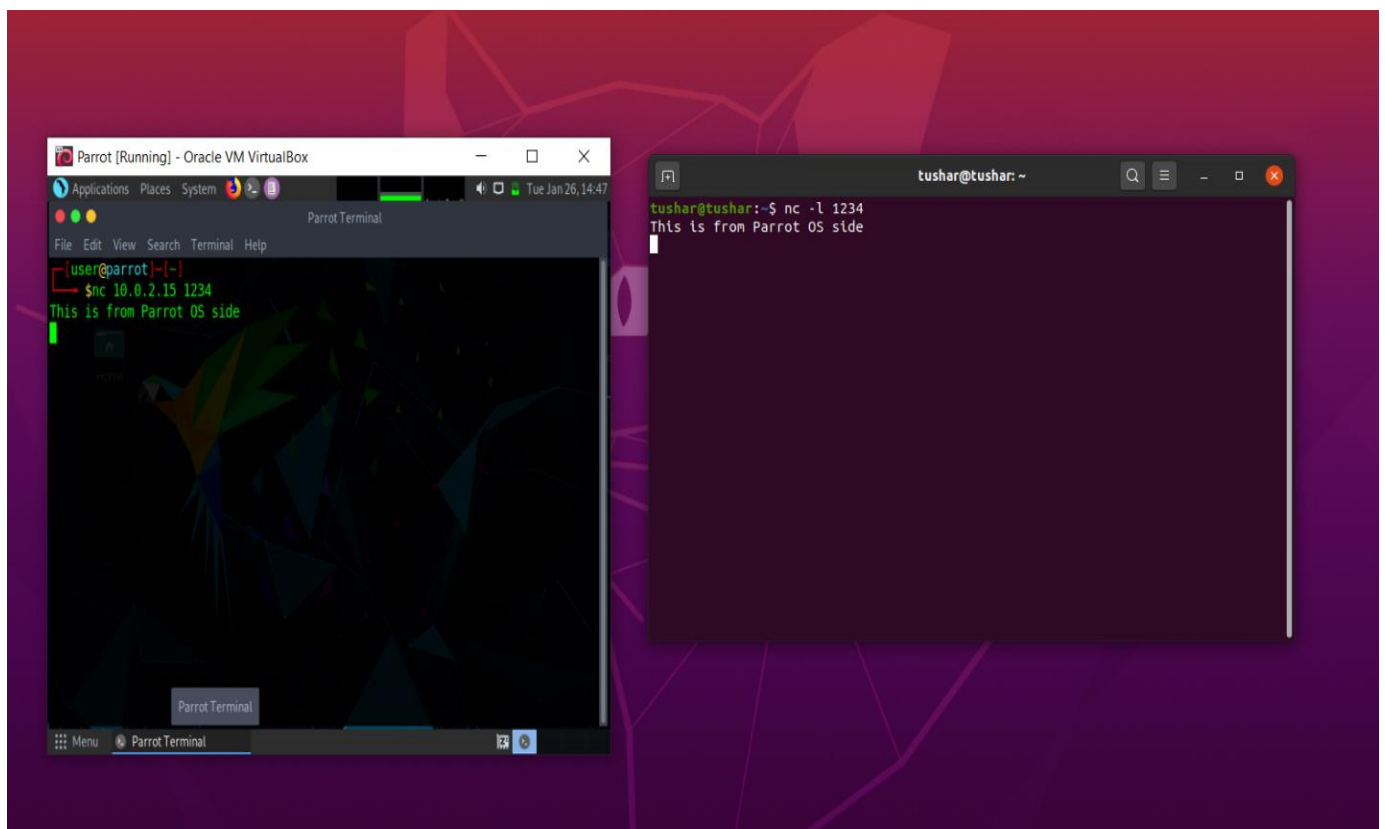


```
tushar@tushar: ~  
tushar@tushar:~$ nc -l 1234  
My name is Tushar  
█
```

7a.2 Inter system communication:

Command on server's terminal (ubuntu): nc -l 1234

Command on clients's terminal (parrot): nc 10.0.2.15 1234

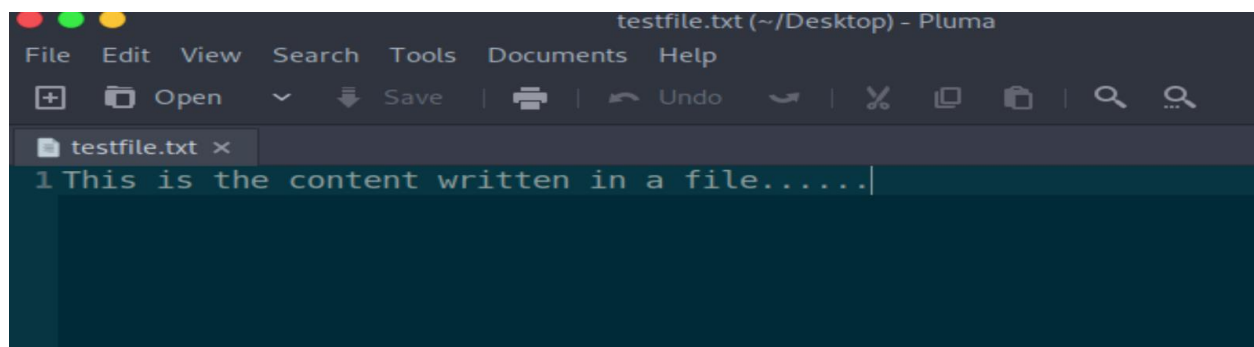
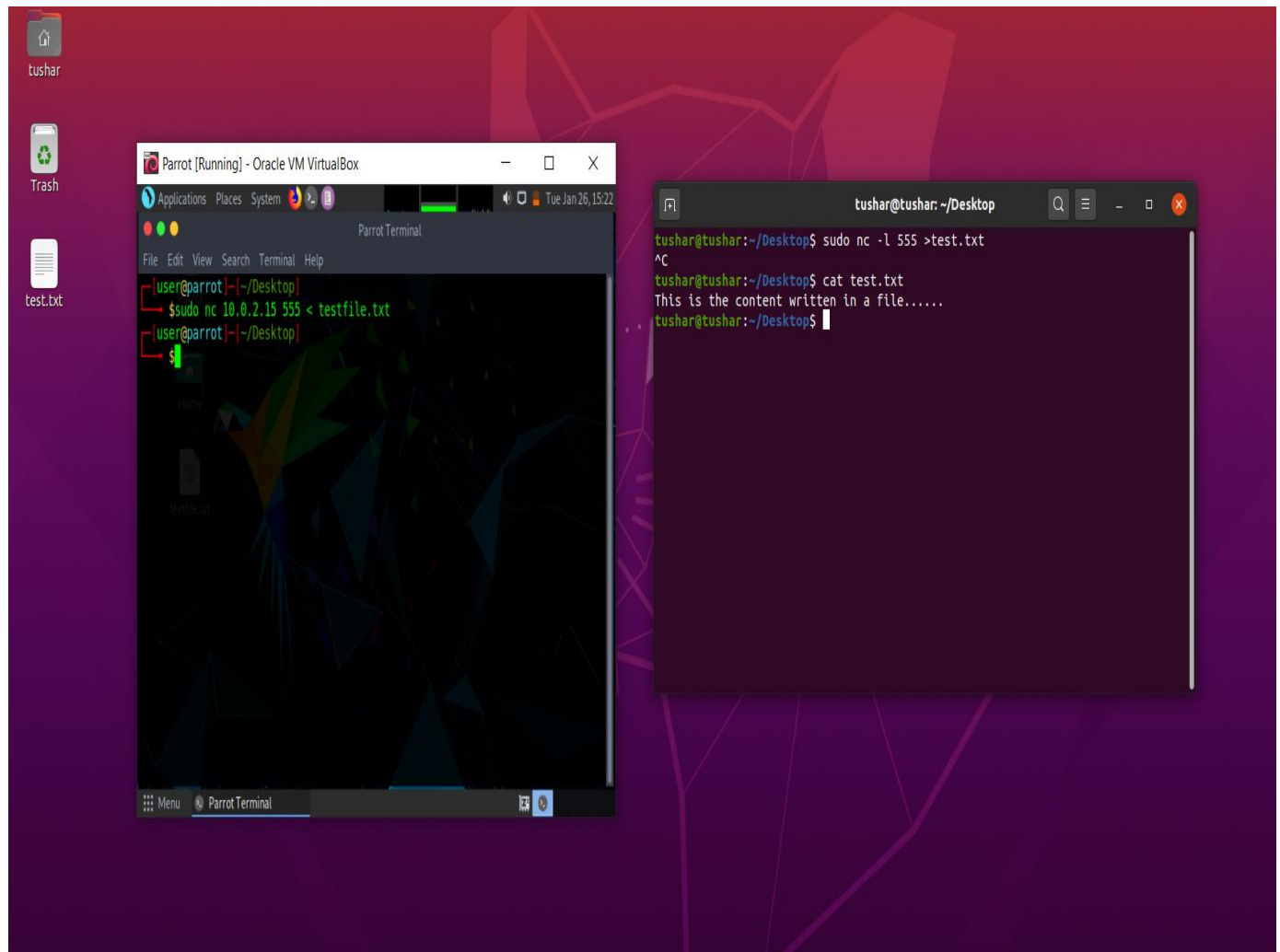


## Task 7 b)- Using Netcat to transfer files

Command on server side (ubuntu): `sudo nc -l 555 > test.txt`

Command on client side (parrot): `sudo nc 10.0.2.15 555 < testfile.txt`

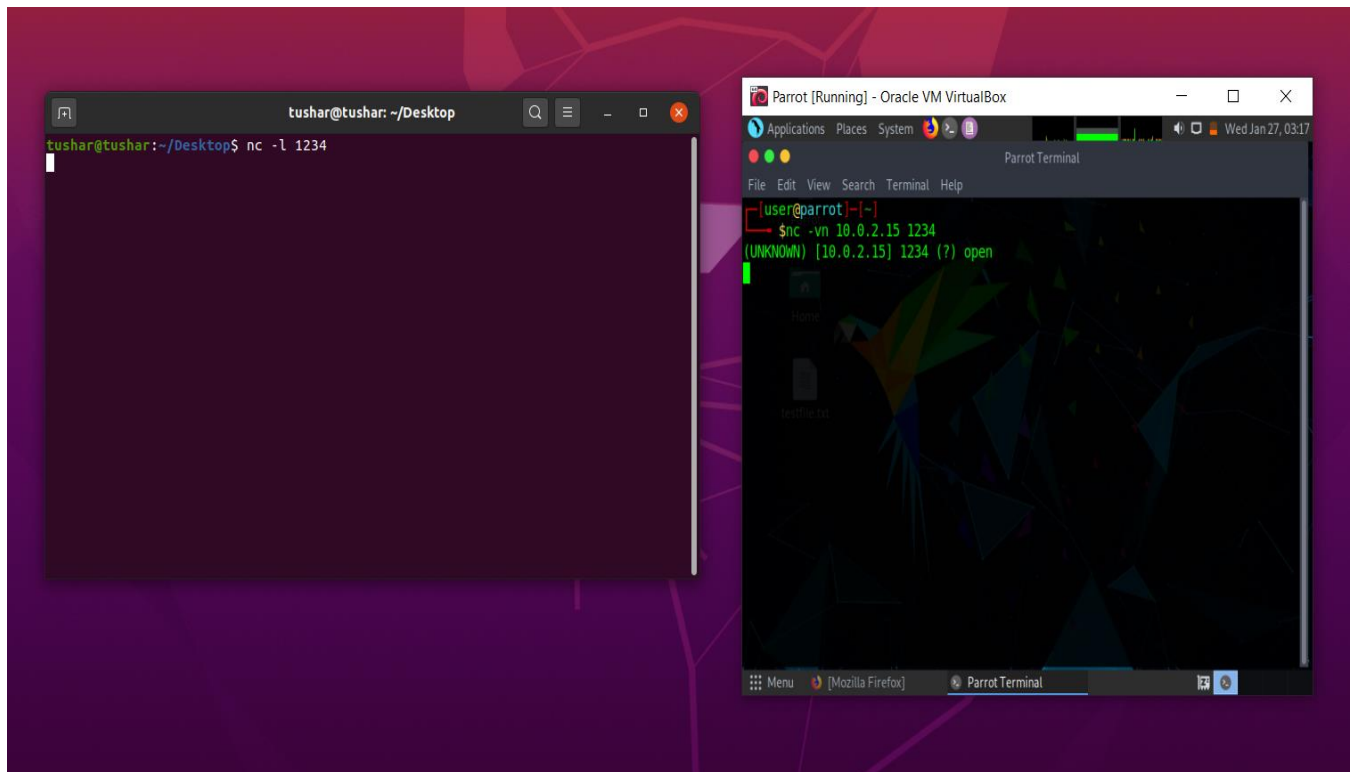
Command on server side to verify file transfer: `cat test.txt`



## Task 7 c)- Other commands

7.1 To check if a particular TCP port of a remote host is open:

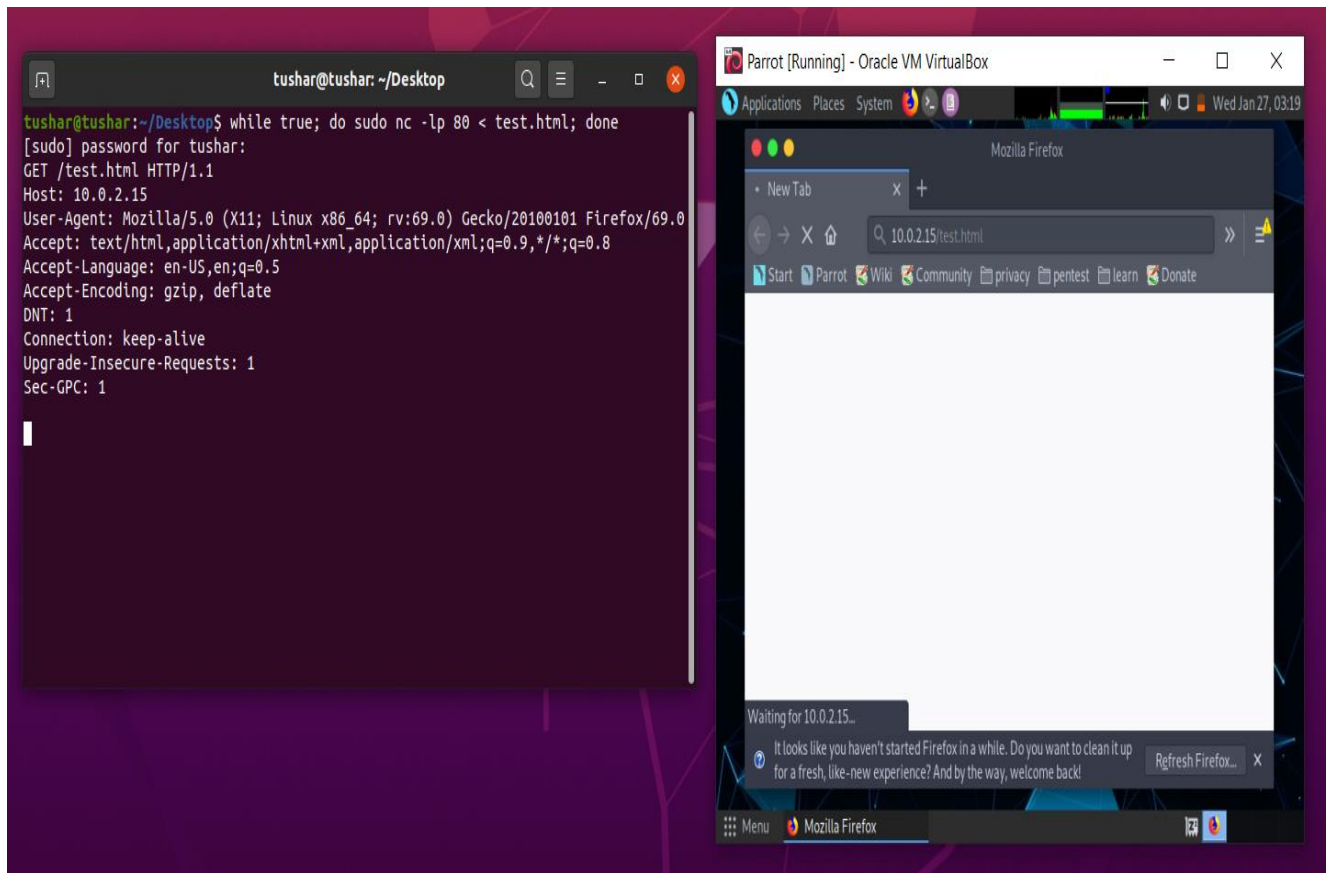
Command: `nc -vn 10.0.2.15 1234`



7.2 To start a web server that serves test.html on port 80:

Command on local host: `while true; do sudo nc -lp 80 < test.html; done`

After opening <http://10.0.2.15/test.html> from another host to access test.html:

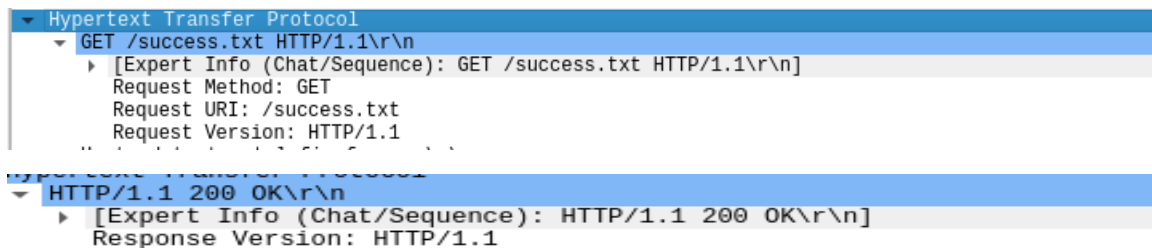


## Questions:

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

Ans: My browser(Firefox) is running HTTP version 1.1.The request header contains the information of this version.

The server is also of HTTP version 1.1 and can be seen in the header of HTTP response.



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

Ans: It can be seen in the response packet as shown below:

```
server: nginx  
Date: Mon, 25 Jan 2021 18:44:25 GMT
```

3) How to tell ping to exit after a specified number of ECHO\_REQUEST packets?

Ans: It can be done by using following command:

```
ping -c 15 www.google.com
```

Here no. of ECHO\_REQUEST packets is 15.

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

Ans: The server field in the HTTP response object stores the remote host app or server on which it is hosted.

Alternatively it can be found using the following command:

```
nmap -O -v www.flipkart.com
```