Computer Network Lab – Week 1

PES1UG20CS806

Divyanshu Sharma

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

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

Command Used: ifconfig (**Displaying all the active network interface)**

```
student@PESSAT-182:~$ ifconfig
enp2s0    Link encap:Ethernet  HWaddr b8:ae:ed:a5:a6:ab
          inet addr:10.2.20.217  Bcast:10.2.20.255  Mask:255.255.255.0
          inet6 addr: fe80::9ad5:a2d4:44b4:8a5f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4839 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1462188 (1.4 MB)  TX bytes:463019 (463.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:358 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28212 (28.2 KB)  TX bytes:28212 (28.2 KB)
```

Analyse and fill the following table:

Interface Name	IPv4/IPv6	MAC Address
Enp2s0	10.2.20.217 / fe80::9ad5:a2d4:44b4:8a5f	b8:ae:ed:a5:a6:ab
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.

Command Used: sudo ifconfig interface_name 10.0.your_section.your_sno netmask 255.255.255.0

: sudo ifconfig enp2s0 10.0.8.06 netmask 255.255.255.0

```
student@PESSAT-182:~$ sudo ifconfig enp2s0 10.0.8.06 netmask 255.255.255.0
sudo: unable to resolve host PESSAT-182: Connection timed out
student@PESSAT-182:~$ ifconfig
enp2s0    Link encap:Ethernet  HWaddr b8:ae:ed:a5:a6:ab
          inet addr:10.0.8.6  Bcast:10.0.8.255  Mask:255.255.255.0
          inet6 addr: fe80::9ad5:a2d4:44b4:8a5f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

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

Command Used: sudo ifconfig interface_name down

sudo ifconfig enp2s0 down **(Deactivating enp2s0)**

```
student@PESSAT-182:~$ sudo ifconfig enp2s0 down
sudo: unable to resolve host PESSAT-182: Connection timed out
```

Command Used: sudo ifconfig interface_name up

sudo ifconfig enp2s0 up **(Activating enp2s0)**

```
student@CSELAB:~$ sudo enp2s0 up
```

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

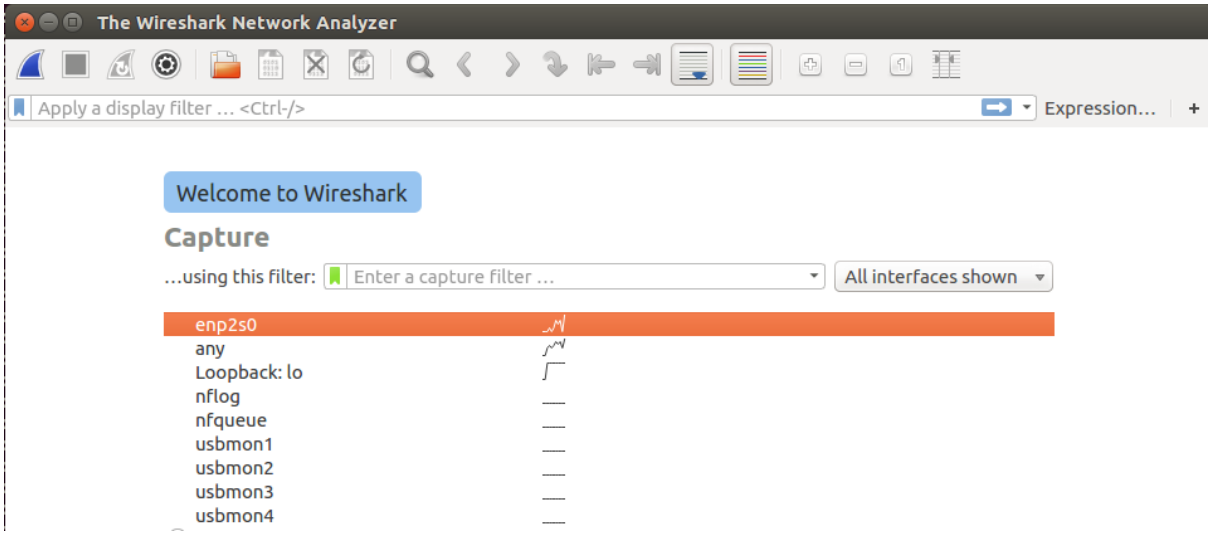
Command Used: ip neigh

```
student@student-H81H3-I:~$ ipneigh
ipneigh: command not found
student@student-H81H3-I:~$ ip neigh
192.168.3.5 dev enp2s0  FAILED
202.148.200.3 dev enp2s0  FAILED
202.148.202.4 dev enp2s0  FAILED
4.2.2.2 dev enp2s0  FAILED
student@student-H81H3-I:~$
```

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

Command Used – ping 10.0.8.06

```
student@PESSAT-182:~$ ping 10.0.8.06
PING 10.0.8.06 (10.0.8.6) 56(84) bytes of data.
64 bytes from 10.0.8.6: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 10.0.8.6: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.0.8.6: icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from 10.0.8.6: icmp_seq=4 ttl=64 time=0.036 ms
64 bytes from 10.0.8.6: icmp_seq=5 ttl=64 time=0.055 ms
64 bytes from 10.0.8.6: icmp_seq=6 ttl=64 time=0.054 ms
64 bytes from 10.0.8.6: icmp_seq=7 ttl=64 time=0.053 ms
64 bytes from 10.0.8.6: icmp_seq=8 ttl=64 time=0.042 ms
64 bytes from 10.0.8.6: icmp_seq=9 ttl=64 time=0.023 ms
64 bytes from 10.0.8.6: icmp_seq=10 ttl=64 time=0.054 ms
64 bytes from 10.0.8.6: icmp_seq=11 ttl=64 time=0.059 ms
64 bytes from 10.0.8.6: icmp_seq=12 ttl=64 time=0.053 ms
64 bytes from 10.0.8.6: icmp_seq=13 ttl=64 time=0.055 ms
64 bytes from 10.0.8.6: icmp_seq=14 ttl=64 time=0.018 ms
64 bytes from 10.0.8.6: icmp_seq=15 ttl=64 time=0.053 ms
64 bytes from 10.0.8.6: icmp_seq=16 ttl=64 time=0.050 ms
64 bytes from 10.0.8.6: icmp_seq=17 ttl=64 time=0.054 ms
64 bytes from 10.0.8.6: icmp_seq=18 ttl=64 time=0.050 ms
64 bytes from 10.0.8.6: icmp_seq=19 ttl=64 time=0.173 ms
64 bytes from 10.0.8.6: icmp_seq=20 ttl=64 time=0.046 ms
64 bytes from 10.0.8.6: icmp_seq=21 ttl=64 time=0.061 ms
64 bytes from 10.0.8.6: icmp_seq=22 ttl=64 time=0.054 ms
```

Observations to be made

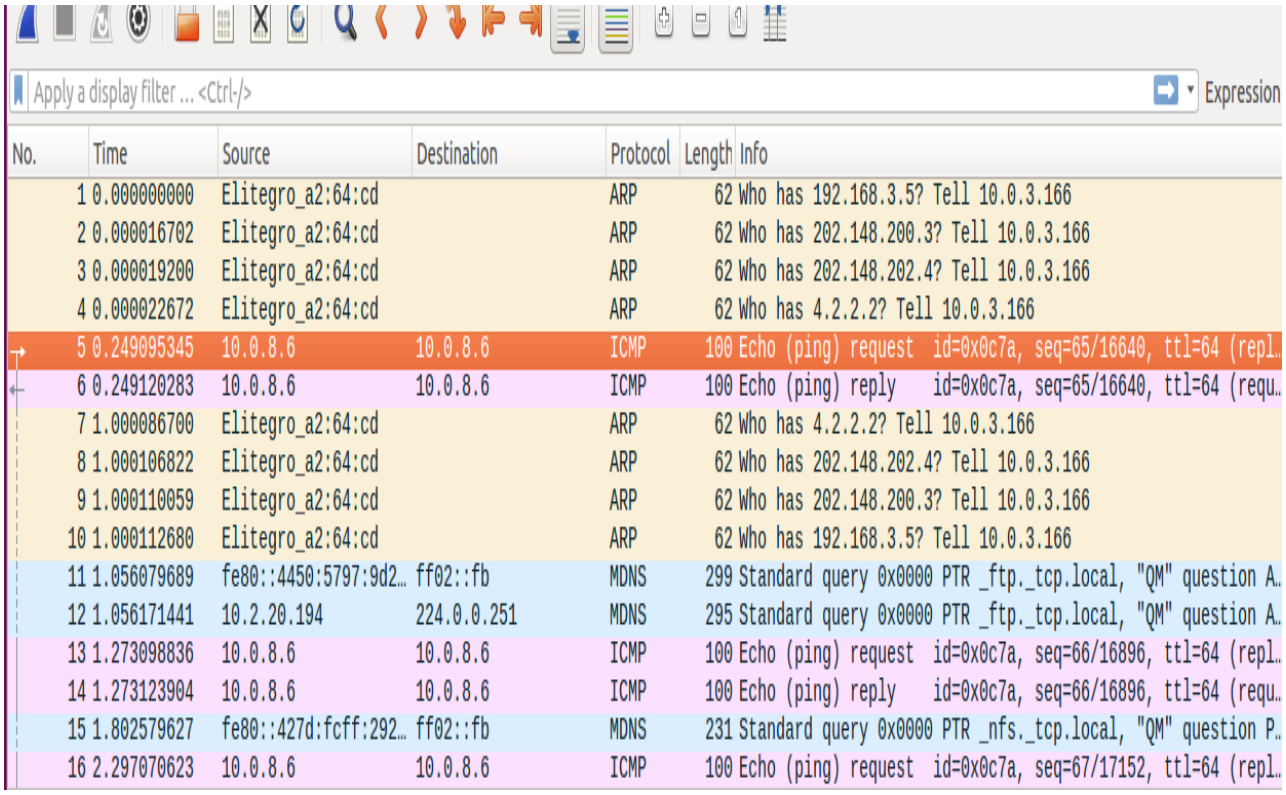
Step 4: Analyse the following in Terminal

- TTL
- Protocol used by ping
- Time

TTL	64
Protocol used by ping	ICMP
Time	Order of 10^{-2} 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.

Showing Request and Response Packet



Apply a display filter ... <Ctrl-/> Expression

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Elitegro_a2:64:cd		ARP	62	Who has 192.168.3.5? Tell 10.0.3.166
2	0.000016702	Elitegro_a2:64:cd		ARP	62	Who has 202.148.200.3? Tell 10.0.3.166
3	0.000019200	Elitegro_a2:64:cd		ARP	62	Who has 202.148.202.4? Tell 10.0.3.166
4	0.000022672	Elitegro_a2:64:cd		ARP	62	Who has 4.2.2.2? Tell 10.0.3.166
5	0.249095345	10.0.8.6	10.0.8.6	ICMP	100	Echo (ping) request id=0x0c7a, seq=65/16640, ttl=64 (repl…
6	0.249120283	10.0.8.6	10.0.8.6	ICMP	100	Echo (ping) reply id=0x0c7a, seq=65/16640, ttl=64 (requ…
7	1.000086700	Elitegro_a2:64:cd		ARP	62	Who has 4.2.2.2? Tell 10.0.3.166
8	1.000106822	Elitegro_a2:64:cd		ARP	62	Who has 202.148.202.4? Tell 10.0.3.166
9	1.000110059	Elitegro_a2:64:cd		ARP	62	Who has 202.148.200.3? Tell 10.0.3.166
10	1.000112680	Elitegro_a2:64:cd		ARP	62	Who has 192.168.3.5? Tell 10.0.3.166
11	1.056079689	fe80::4450:5797:9d2…	ff02::fb	MDNS	299	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question A…
12	1.056171441	10.2.20.194	224.0.0.251	MDNS	295	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question A…
13	1.273098836	10.0.8.6	10.0.8.6	ICMP	100	Echo (ping) request id=0x0c7a, seq=66/16896, ttl=64 (repl…
14	1.273123904	10.0.8.6	10.0.8.6	ICMP	100	Echo (ping) reply id=0x0c7a, seq=66/16896, ttl=64 (requ…
15	1.802579627	fe80::427d:fcff:292…	ff02::fb	MDNS	231	Standard query 0x0000 PTR _nfs._tcp.local, "QM" question P…
16	2.297070623	10.0.8.6	10.0.8.6	ICMP	100	Echo (ping) request id=0x0c7a, seq=67/17152, ttl=64 (repl…

Request Packet

Wireshark · Packet 5 · any

```
▼ Frame 5: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interfa
  ▼ Interface id: 0 (any)
      Interface name: any
    Encapsulation type: Linux cooked-mode capture (25)
    Arrival Time: Jan 22, 2021 13:32:04.627143414 IST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1611302524.627143414 seconds
    [Time delta from previous captured frame: 0.249072673 seconds]
    [Time delta from previous displayed frame: 0.249072673 seconds]
    [Time since reference or first frame: 0.249095345 seconds]
    Frame Number: 5
    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: 10.0.8.6, Dst: 10.0.8.6
```

Wireshark · Packet 5 · any

```
    Protocol: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 10.0.8.6, Dst: 10.0.8.6
    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: 0xa0a0 (41120)
  ▼ Flags: 0x4000, Don't fragment
      0... .... .... .... = Reserved bit: Not set
      .1.. .... .... .... = Don't fragment: Set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x75fd [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.8.6
    Destination: 10.0.8.6
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x01fa [correct]
    [Checksum Status: Good]
    Identifier (BE): 3194 (0x0c7a)
```

Response Packet

Wireshark · Packet 6 · any

```
    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: 10.0.8.6, Dst: 10.0.8.6
    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: 0xa0a1 (41121)
  ▼ Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0xb5fc [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.8.6
    Destination: 10.0.8.6
▼ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x09fa [correct]
    [Checksum Status: Good]
    Identifier (BE): 3194 (0x0c7a)
    Identifier (LE): 31244 (0x7a0c)
    Sequence number (BE): 65 (0x0041)
    Sequence number (LE): 16640 (0x4100)
    [Request frame: 5]
    [Response time: 0.025 ms]
    Timestamp from icmp data: Jan 22, 2021 13:32:04.000000000 IST
    [Timestamp from icmp data (relative): 0.627168352 seconds]
```

Wireshark · Packet 6 · any

```
▼ Frame 6: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
  ▼ Interface id: 0 (any)
      Interface name: any
    Encapsulation type: Linux cooked-mode capture (25)
    Arrival Time: Jan 22, 2021 13:32:04.627168352 IST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1611302524.627168352 seconds
    [Time delta from previous captured frame: 0.000024938 seconds]
    [Time delta from previous displayed frame: 0.000024938 seconds]
    [Time since reference or first frame: 0.249120283 seconds]
    Frame Number: 6
    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: 10.0.8.6, Dst: 10.0.8.6
    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: 0xa0a1 (41121)
  ▼ Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
```

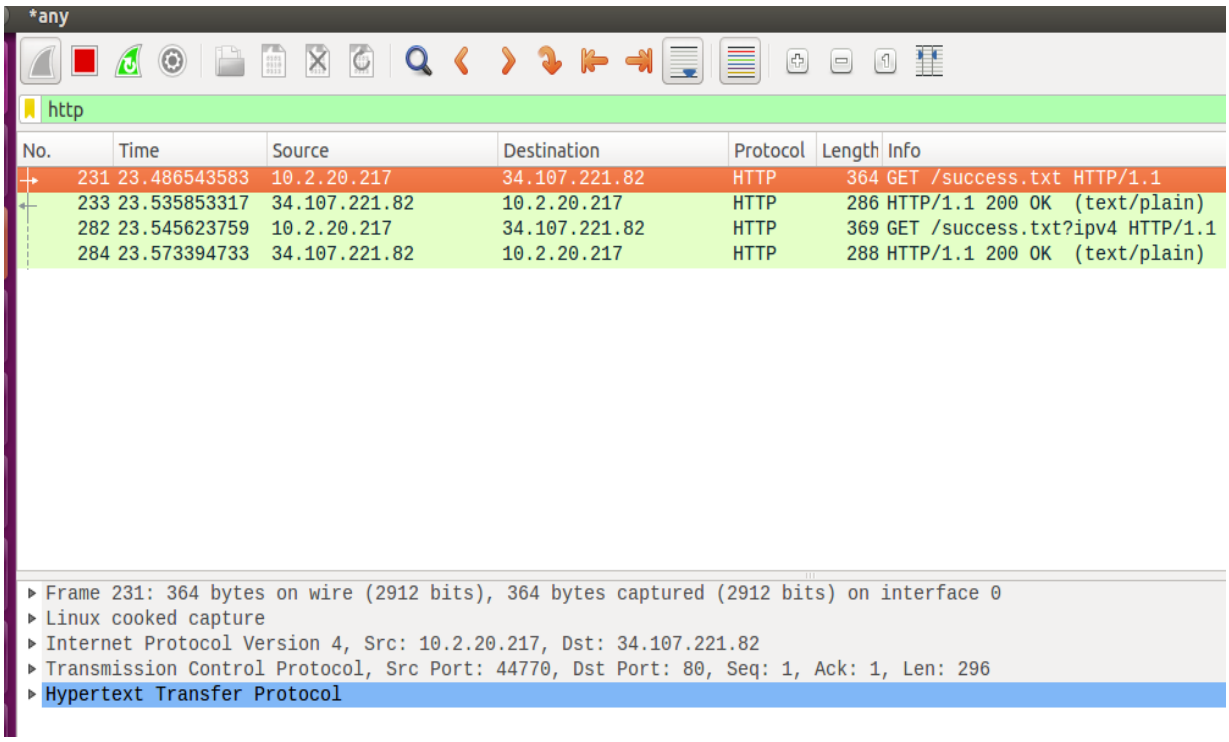
Details	First Echo Request	First Echo Reply
Frame Number	5	6
Source IP address	10.0.8.6	10.0.8.6
Destination IP address	10.0.8.6	10.0.8.6
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	00:00:00:00:00:00	00:00:00:00:00:00
Destination Ethernet Address	00:00:00:00:00:00	00:00:00:00:00:00
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



Request Packet

Wireshark · Packet 1914 · any

```
▶ Frame 1914: 411 bytes on wire (3288 bits), 411 bytes captured (3288 bits) on interface 0
▼ Linux cooked capture
    Packet type: Sent by us (4)
    Link-layer address type: 1
    Link-layer address length: 6
    Source: Elitegro_a5:a5:e4 (b8:ae:ed:a5:a5:e4)
    Unused: bb3f
    Protocol: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 10.2.20.199, Dst: 163.53.78.110
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 395
    Identification: 0x0a92 (2706)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x1e6f [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.2.20.199
    Destination: 163.53.78.110
▶ Transmission Control Protocol, Src Port: 53792, Dst Port: 80, Seq: 1, Ack: 1, Len: 343
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: www.flipkart.com\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.flipkart.com/]
    [HTTP request 1/1]
    [Response in frame: 1916]
```

Response Packet

Wireshark · Packet 1916 · any

```
▶ Frame 1916: 415 bytes on wire (3320 bits), 415 bytes captured (3320 bits) on interface 0
▼ Linux cooked capture
    Packet type: Unicast to us (0)
    Link-layer address type: 1
    Link-layer address length: 6
    Source: HewlettP_d4:86:00 (14:58:d0:d4:86:00)
    Unused: 0106
    Protocol: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 163.53.78.110, Dst: 10.2.20.199
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 399
    Identification: 0xf3de (62430)
  ▶ Flags: 0x0000
    Time to live: 58
    Protocol: TCP (6)
    Header checksum: 0x7b1e [validation disabled]
    [Header checksum status: Unverified]
    Source: 163.53.78.110
    Destination: 10.2.20.199
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 53792, Seq: 1, Ack: 344, Len: 347
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 301 Moved Permanently\r\n
    Server: nginx\r\n
    Date: Wed, 27 Jan 2021 07:50:41 GMT\r\n
    Content-Type: text/html\r\n
  ▶ Content-Length: 178\r\n
    Location: https://www.flipkart.com/\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.012462130 seconds]
    [Request in frame: 1914]
    [Request URI: http://www.flipkart.com/]
    File Data: 178 bytes
▶ Line-based text data: text/html (7 lines)
```

Observations to be made

Step 3

Details	**First Echo Request**	**First Echo Reply**
Frame Number	1914	1916
Source Port	53792	80
Destination Port	80	53792
Source IP address	10.2.20.199	163.53.78.110
Destination IP address	163.53.78.110	10.2.20.199
Source Ethernet Address	b8:ae:ed:a5:a5:e4	14:58:d0:d4:86:00
Destination Ethernet Address	14:58:d0:d4:86:00	b8:ae:ed:a5:a5:e4

(Connection Details)

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

HTTP Request		**HTTP Response**	
Get	GET / HTTP1.1\r\n	**Server**	nginx
Host	www.flipkart.com	**Content-Type**	Text/html
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0\r\n	**Date**	Wed, 27 Jan 2021 07:50:41 GMT
Accept-Language	En-US, en;q=0.5\r\n	**Location**	https://www.flipkart.com/
Accept-Encoding	Gzip, deflate\r\n	**Content-Length**	178\r\n
Connection	Keep-alive	**Connection**	Keep-alive

Task 4: Capturing packets with tcpdump

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

Command Used - sudo tcpdump -D

```
student@student-H81H3-I:~$ sudo tcpdump -D
1.enp2s0 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.nflog (Linux netfilter log (NFLOG) interface)
5.nfqueue (Linux netfilter queue (NFQUEUE) interface)
6.usbmon1 (USB bus number 1)
7.usbmon2 (USB bus number 2)
8.usbmon3 (USB bus number 3)
9.usbmon4 (USB bus number 4)
```

(Viewing Interfaces available for Capture)

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

Command Used - sudo tcpdump -i any

```
student@student-H81H3-I:~$ 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), capture size 262144 bytes
12:25:26.608732 IP6 fe80::e22d:e1ae:99f8:118f.mdns > ff02::fb.mdns: 0 [4a] [11q] PTR (QM)? _ftp._tcp.local. AAAA (QM)? linux-2.local. A (QM)? l
inux-2.local. TXT (QM)? UBUNTU18.04._smb._tcp.local. SRV (QM)? UBUNTU18.04._smb._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _afpovertcp._t
cp.local. PTR (QM)? _smb._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local. (237)
12:25:26.608787 IP 10.2.20.172.mdns > 224.0.0.251.mdns: 0 [5a] [11q] PTR (QM)? _ftp._tcp.local. AAAA (QM)? linux-2.local. A (QM)? linux-2.local
. TXT (QM)? UBUNTU18.04._smb._tcp.local. SRV (QM)? UBUNTU18.04._smb._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _afpovertcp._tcp.local. PT
R (QM)? _smb._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local. (253)
12:25:26.609900 IP localhost.54410 > student-H81H3-I.domain: 3394+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arp
a. (90)
12:25:26.610013 IP 10.2.20.199.44210 > 192.168.3.5.domain: 27920+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa
. (90)
12:25:26.610531 IP 192.168.3.5.domain > 10.2.20.199.44210: 27920 NXDomain 0/1/0 (154)
12:25:26.869465 IP localhost.45037 > student-H81H3-I.domain: 60264+ PTR? 5.3.168.192.in-addr.arpa. (42)
12:25:26.869614 IP 10.2.20.199.44210 > 192.168.3.5.domain: 18320+ PTR? 5.3.168.192.in-addr.arpa. (42)
12:25:27.109538 IP 10.0.8.22.mdns > 224.0.0.251.mdns: 0 [3q] [4n] ANY (QM)? 7.e.0.2.1.3.3.1.3.e.3.7.3.6.3.b.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6
.arpa. ANY (QM)? PESSAT-192.local. ANY (QM)? 22.8.0.10.in-addr.arpa. (208)
12:25:27.110021 IP localhost.59054 > student-H81H3-I.domain: 14025+ PTR? 22.8.0.10.in-addr.arpa. (40)
12:25:27.110149 IP 10.2.20.199.44210 > 192.168.3.5.domain: 9032+ PTR? 22.8.0.10.in-addr.arpa. (40)
12:25:27.110168 IP 10.2.20.199.44210 > 202.148.200.3.domain: 9032+ PTR? 22.8.0.10.in-addr.arpa. (40)
12:25:27.110180 IP 10.2.20.199.44210 > dns2.dwl.co.in.domain: 9032+ PTR? 22.8.0.10.in-addr.arpa. (40)
12:25:27.110189 IP 10.2.20.199.44210 > b.resolvers.level3.net.domain: 9032+ PTR? 22.8.0.10.in-addr.arpa. (40)
12:25:27.111244 IP localhost.35162 > student-H81H3-I.domain: 59743+ PTR? 3.200.148.202.in-addr.arpa. (44)
12:25:27.111346 IP 10.2.20.199.44210 > 192.168.3.5.domain: 11397+ PTR? 3.200.148.202.in-addr.arpa. (44)
12:25:27.326683 IP b.resolvers.level3.net.domain > 10.2.20.199.44210: 9032 NXDomain* 0/1/0 (99)
12:25:29.833158 IP 10.2.20.199.38948 > fra15s46-in-f14.1e100.net.http: Flags [FP.], seq 8192851, ack 2971778173, win 502, options [nop,nop,TS v
al 383949145 ecr 580600259], length 0
12:25:29.833487 IP localhost.57642 > student-H81H3-I.domain: 35312+ PTR? 142.16.217.172.in-addr.arpa. (45)
```

(Capturing all Packets in any Interface)

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:

Command Used - sudo tcpdump -i any -c5 icmp

```
student@CSELAB:~$ 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), capture size 262144 bytes
^Z
[2]+  Stopped                 sudo tcpdump -i any -c5 icmp
```

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

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

```
student@CSELAB:~$ 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), capture size 262144 bytes
13:40:28.855906 IP 10.2.20.195.40132 > 34.107.221.82.80: Flags [.], ack 456533556, win 321, options [nop,nop,TS val 453416 ecr 3514518024], len
gth 0
E..4.'@.@...
..."k.R...P9....6&4...A.......
...(.{].
13:40:28.884541 IP 34.107.221.82.80 > 10.2.20.195.40132: Flags [.], ack 1, win 303, options [nop,nop,TS val 3514528178 ecr 450876], length 0
E..44...9..k"k.R
....P...6&49....../s......
```

```
....P...6&49....../.......
.|.....<
10 packets captured
10 packets received by filter
0 packets dropped by kernel
student@CSELAB:~$
```

Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.

Command Used - sudo traceroute www.google.com

```
student@student-H81H3-I:~$ sudo traceroute www.google.com
traceroute to www.google.com (142.250.76.68), 30 hops max, 60 byte packets
 1  10.2.20.1 (10.2.20.1)  1.412 ms  1.406 ms  1.398 ms
 2  192.168.4.1 (192.168.4.1)  0.894 ms  0.899 ms  0.889 ms
 3  192.168.254.1 (192.168.254.1)  0.160 ms  0.163 ms  0.166 ms
 4  1.6.180.188 (1.6.180.188)  2.148 ms 14.143.35.157.static-bangalore.vsnl.net.in (14.143.35.157)  1.944 ms 1.6.180.188 (1.6.180.188)  2.174 m
s
 5  100.66.8.23 (100.66.8.23)  16.376 ms * *
 6  100.66.8.23 (100.66.8.23)  16.316 ms 121.240.1.46 (121.240.1.46)  7.729 ms 100.67.56.103 (100.67.56.103)  15.909 ms
 7  72.14.210.200 (72.14.210.200)  16.344 ms 74.125.242.145 (74.125.242.145)  9.333 ms 72.14.210.200 (72.14.210.200)  16.541 ms
 8  108.170.248.162 (108.170.248.162)  16.600 ms 142.250.228.187 (142.250.228.187)  10.295 ms 108.170.248.194 (108.170.248.194)  17.239 ms
 9  209.85.251.15 (209.85.251.15)  25.802 ms maa05s14-in-f4.1e100.net (142.250.76.68)  8.017 ms 209.85.251.15 (209.85.251.15)  25.711 ms
student@student-H81H3-I:~$
```

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

The destination address is **142.250.76.68** and there were **30 hops**.

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

Command used - sudo traceroute -n www.google.com

```
student@student-H81H3-I:~$ sudo traceroute -n www.google.com
traceroute to www.google.com (142.250.76.68), 30 hops max, 60 byte packets
 1  10.2.20.1  1.185 ms  1.174 ms  1.170 ms
 2  192.168.4.1  0.927 ms  0.920 ms  0.918 ms
 3  192.168.254.1  0.109 ms  0.137 ms  0.144 ms
 4  1.6.180.188  1.946 ms  2.044 ms 14.143.35.157  2.023 ms
 5  * * 172.29.209.114  7.411 ms
 6  100.67.56.103  15.999 ms 100.70.66.174  16.061 ms 121.240.1.46  7.741 ms
 7  72.14.210.200  16.208 ms  16.037 ms 74.125.242.129  10.326 ms
 8  108.170.248.178  17.412 ms 108.170.248.179  16.909 ms 142.250.228.245  10.256 ms
 9  209.85.251.15  25.443 ms 209.85.255.161  24.503 ms 142.250.76.68  7.533 ms
student@student-H81H3-I:~$
```

(Disabling mapping of IP addresses with hostnames)

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

Command Used - sudo traceroute -I www.google.com

```
student@student-H81H3-I:~$ sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.76.68), 30 hops max, 60 byte packets
 1  10.2.20.1 (10.2.20.1)  1.220 ms  1.210 ms  1.483 ms
 2  192.168.4.1 (192.168.4.1)  0.916 ms  0.918 ms  0.916 ms
 3  192.168.254.1 (192.168.254.1)  0.142 ms  0.158 ms  0.170 ms
 4  1.6.180.188 (1.6.180.188)  22.398 ms * *
 5  * * *
 6  100.67.56.103 (100.67.56.103)  40.589 ms  40.816 ms  40.992 ms
 7  72.14.210.200 (72.14.210.200)  40.959 ms  40.717 ms  40.796 ms
 8  108.170.248.211 (108.170.248.211)  40.936 ms  41.630 ms  42.466 ms
 9  216.239.50.22 (216.239.50.22)  32.939 ms  32.217 ms  32.183 ms
10  142.250.212.7 (142.250.212.7)  38.523 ms  38.551 ms  38.543 ms
11  74.125.242.129 (74.125.242.129)  39.290 ms  39.325 ms  39.330 ms
12  142.250.228.245 (142.250.228.245)  39.308 ms  39.310 ms  25.355 ms
13  maa05s14-in-f4.1e100.net (142.250.76.68)  24.907 ms  24.915 ms  24.601 ms
student@student-H81H3-I:~$
```

(traceroute with ICMP Protocol)

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.

Command Used - sudo traceroute -T www.google.com

```
student@student-H81H3-I:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (142.250.76.68), 30 hops max, 60 byte packets
 1  10.2.20.1 (10.2.20.1)  1.224 ms  1.208 ms  1.208 ms
 2  192.168.4.1 (192.168.4.1)  0.869 ms  0.867 ms  0.862 ms
 3  192.168.254.1 (192.168.254.1)  0.149 ms  0.163 ms  0.192 ms
 4  14.143.35.157.static-bangalore.vsnl.net.in (14.143.35.157)  2.163 ms 1.6.180.188 (1.6.180.188)  4.583 ms  4.449 ms
 5  172.31.167.46 (172.31.167.46)  7.654 ms * *
 6  121.240.1.46 (121.240.1.46)  7.454 ms 100.70.66.174 (100.70.66.174)  15.874 ms  15.918 ms
 7  74.125.242.129 (74.125.242.129)  8.821 ms 72.14.210.200 (72.14.210.200)  17.075 ms 74.125.242.145 (74.125.242.145)  8.120 ms
 8  108.170.248.195 (108.170.248.195)  16.561 ms 108.170.248.179 (108.170.248.179)  27.056 ms 108.170.248.211 (108.170.248.211)  17.613 ms
 9  209.85.251.243 (209.85.251.243)  26.974 ms 142.250.212.7 (142.250.212.7)  25.367 ms 209.85.251.243 (209.85.251.243)  26.423 ms
10  maa05s14-in-f4.1e100.net (142.250.76.68)  8.151 ms  9.414 ms 74.125.242.145 (74.125.242.145)  22.403 ms
student@student-H81H3-I:~$
```

(Testing TCP Connection with traceroute)

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.

Command Used - nmap www.pes.edu

```
student@student-H81H3-I:~$ sudo apt-get install nmap
Reading package lists... Done
Building dependency tree
Reading state information... Done
nmap is already the newest version (7.01-2ubuntu2).
The following packages were automatically installed and are no longer required:
  gyp libjs-inherits libjs-node-uuid libssl-dev libssl-doc libuv1 libuv1-dev node-abbrev node-ansi node-ansi-color-table node-archy
  node-async node-block-stream node-combined-stream node-cookie-jar node-delayed-stream node-forever-agent node-form-data node-fstream
  node-fstream-ignore node-github-url-from-git node-glob node-graceful-fs node-gyp node-inherits node-ini node-json-stringify-safe
  node-lockfile node-lru-cache node-mime node-minimatch node-mkdirp node-mute-stream node-node-uuid node-nopt node-normalize-package-data
  node-npmlog node-once node-osenv node-qs node-read node-read-package-json node-request node-retry node-rimraf node-semver node-sha
  node-sigmund node-slide node-tar node-tunnel-agent node-underscore node-which zlib1g-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 41 not upgraded.
student@student-H81H3-I:~$ nmap www.pes.edu

Starting Nmap 7.01 ( https://nmap.org ) at 2021-01-25 12:35 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.042s 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 8.76 seconds
```

(Scanning Host with Hostname)

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

Command Used - nmap 163.53.78.128

```
student@student-H81H3-I:~$ nmap 163.53.78.128

Starting Nmap 7.01 ( https://nmap.org ) at 2021-01-25 12:37 IST
Nmap scan report for 163.53.78.128
Host is up (0.0094s latency).
Not shown: 996 filtered ports
PORT      STATE  SERVICE
21/tcp    open   ftp
80/tcp    open   http
443/tcp   open   https
6346/tcp  closed gnutella

Nmap done: 1 IP address (1 host up) scanned in 4.63 seconds
```

(Scanning Host with IP Address)

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

Command Used - nmap 192.168.1.1 192.168.1.2 192.168.1.3

```
student@student-H81H3-I:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3

Starting Nmap 7.01 ( https://nmap.org ) at 2021-01-25 12:40 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 4.29 seconds
student@student-H81H3-I:~$
```

Task 7 a): Netcat as Chat tool

a) Intra system communication (Using 2 terminals in the same system)

Step 1: Open a terminal (Ctrl+Alt+T). This will act as a Server.

Step 2: Type nc -l any_portnum (For eg., nc -l 1234)

Command Used : nc –l 8090

Note: It will goto listening mode

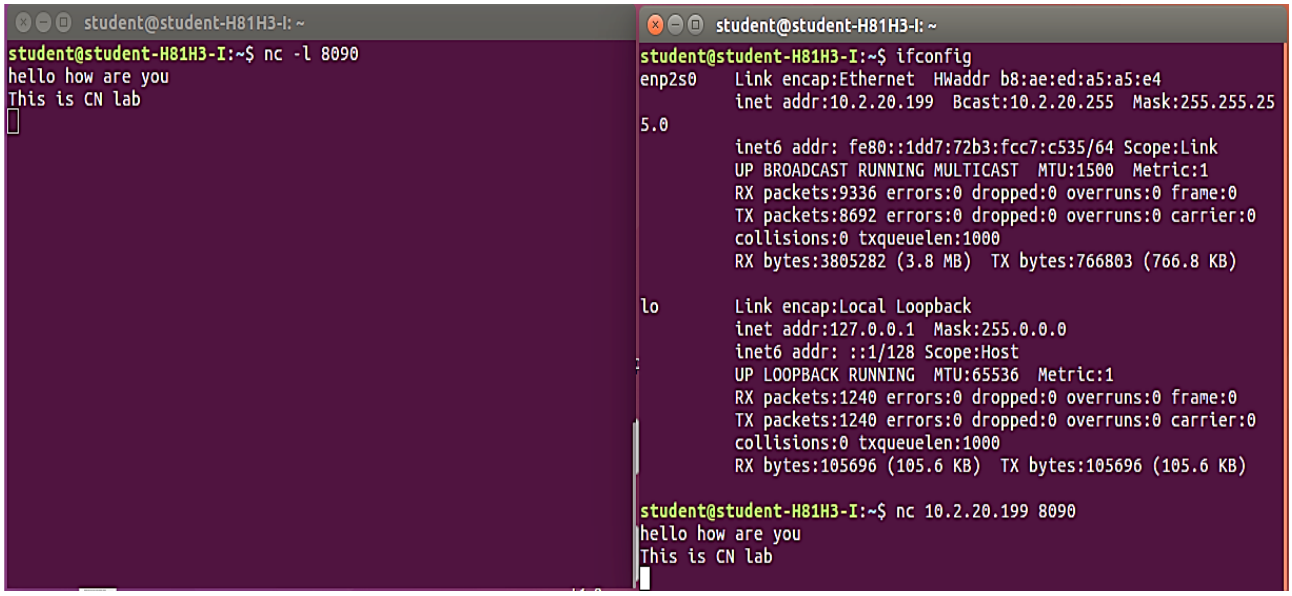
Step 3: Open another terminal and this will act as a client.

Step 4: Type nc <your-system-ip-address> portnum

Command Used : nc 10.2.20.199 8090

Note: portnum should be common in both the terminals (for eg., nc 10.0.2.8 1234)

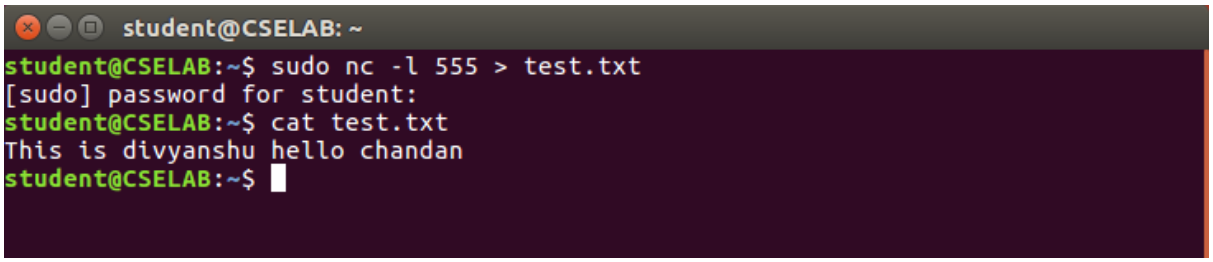
Step 5: Type anything in client will appear in server



Task 7 b): Use Netcat to Transfer Files The netcat utility can also be used to transfer files.

Step 1: At the server side, create an empty file named 'test.txt' sudo nc -l 555 > test.txt

Step 2: At the client side, we have a file 'testfile.txt'. Add some contents to it.



Step 3: Run the client as: sudo nc 10.0.2.8 555 < testfile.txt here, 10.0.2.8 is the IP address of server and 555 is the port number.

```
student@CSELAB:~/Desktop$ cat testfile.txt
This is divyanshu hello chandan
student@CSELAB:~/Desktop$ sudo nc 10.0.8.80 555 < testfile.txt
student@CSELAB:~/Desktop$ ifconfig
enp2s0    Link encap:Ethernet  HWaddr b8:ae:ed:a5:a6:0f
          inet addr:10.0.8.90  Bcast:10.0.8.255  Mask:255.255.255.0
          inet6 addr: fe80::c96c:b21b:5a9a:ed38/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1215 errors:0 dropped:0 overruns:0 frame:0
          TX packets:492 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:134401 (134.4 KB)  TX bytes:64822 (64.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:14653 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14653 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:1057130 (1.0 MB)  TX bytes:1057130 (1.0 MB)
```

Step 4: At server side, verify the file transfer using the command cat test.txt

```
student@CSELAB:~$ cat test.txt
This is divyanshu hello chandan
student@CSELAB:~$
```

Task 7 c): Other Commands

1) To test if a particular TCP port of a remote host is open.

Command Used :- nc -vn 10.0.2.8 555

```
student@CSELAB:~$ nc -vn 10.0.8.80 80
Connection to 10.0.8.80 80 port [tcp/*] succeeded!
```

2) Run a web server with a static web page.

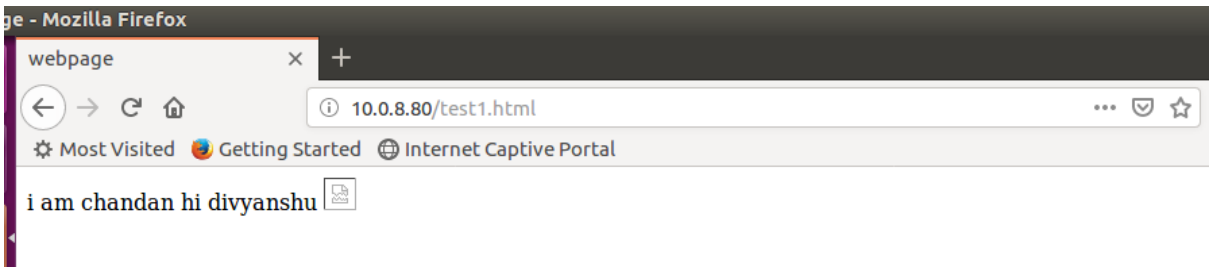
Step 1: Run the command below on local host (e.g. 10.0.2.8) to start a web server that serves test.html on port 80.

Command Used:- while true; do sudo nc -lp 80 < test1.html; done

```
@CSELAB: /var/www/html
student@CSELAB:/var/www/html$ gedit test1.html
^Z
[1]+  Stopped                 gedit test1.html
student@CSELAB:/var/www/html$ cat test1.html
<html>
<head>
<title>
webpage</title>
</head>
<body>
i am chandan hi divyanshu
<img src="/var/www/html/up.png">
</body>
</html>
student@CSELAB:/var/www/html$ while true;do sudo nc -lp 80 < test1.html; done
[sudo] password for student:
nc: listen: Address already in use
```

```
nc: listen: Address already in use
^Z
[2]+  Stopped                 sudo nc -lp 80 < test1.html
student@CSELAB:/var/www/html$
```

Step 2: Now open **http://10.0.8.80/test1.html** from another host to access it.



Step 3: Observe the details on the terminal

Questions

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

   Answer – The Firefox browser used is running HTTP v1.1, and this can be seen in the request header which contains the method (GET) followed by the HTTP version. Similarly, the HTTP version of the web server is v1.1 and can be seen in the header of the HTTP response sent back to the browser.

```
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: www.flipkart.com\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
```

Request

```
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 301 Moved Permanently\r\n
    Server: nginx\r\n
    Date: Wed, 27 Jan 2021 07:50:41 GMT\r\n
    Content-Type: text/html\r\n
  ▶ Content-Length: 178\r\n
    Location: https://www.flipkart.com/\r\n
    \r\n
    [HTTP response 1/1]
```

Response

2. **How to tell ping to exit after a specified number of ECHO_REQUEST packets?**

   Answer – Ping continues to send ICMP packages until it receives an interrupt signal. To specify the number of ECHO_REQUEST packages after which ping will exit, we can use the -c option followed by the number of packages. ping -c 10 www.pes.edu