

## COMPUTER NETWORKS LAB

### WEEK 1

NAME: VISHWA MEHUL MEHTA

SRN: PES2UG20CS389

SECTION: F

DATE: 21/01/2022

Study and understand the basic networking tools -  
Wireshark, Tcpdump, Ping, Traceroute and Netcat.

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

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

ifconfig (or) ip addr show

```
vishwa@pop-os:~$ 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::c4c3:1e2:88b7:59c6 prefixlen 64 scopeid 0x10
    ether 08:00:27:4f:0b:16 txqueuelen 1000 (Ethernet)
    RX packets 212320 bytes 260305880 (260.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 98034 bytes 11674264 (11.6 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
```

```
vishwa@pop-os:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UP
    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 pfifo_fast
    link/ether 08:00:27:4f:0b:16 brd ff:ff:ff:ff:ff:ff
```

Analyze and fill the following table:

### ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address	
lo	127.0.0.1	00:00:00:00:00:00	Loop back device
enp0s3	10.0.2.15	08:00:27:4f:0b:16	Ethernet
WLAN	-	-	-

**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**

```
vishwa@pop-os:~$ sudo ifconfig enp0s3 10.0.6.56
vishwa@pop-os:~$ 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 08:00:27:4f:0b:16 brd ff:ff:ff:ff:ff:ff
    inet 10.0.6.56/8 brd 10.255.255.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::c4c3:1e2:88b7:59c6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
vishwa@pop-os:~$
```

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

**sudo ifconfig interface\_name down**

**sudo ifconfig interface\_name up**

```
vishwa@pop-os:~$ sudo ifconfig enp0s3 down
vishwa@pop-os:~$ 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> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:00:27:4f:0b:16 brd ff:ff:ff:ff:ff:ff
vishwa@pop-os:~$
```

```
vishwa@pop-os:~$ sudo ifconfig enp0s3 up
vishwa@pop-os:~$ 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 08:00:27:4f:0b:16 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 86398sec preferred_lft 86398sec
    inet6 fe80::c4c3:1e2:88b7:59c6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
vishwa@pop-os:~$
```

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

```
vishwa@pop-os:~$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 STALE
vishwa@pop-os:~$
```

## 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.**

```
vishwa@pop-os:~$ sudo ifconfig enp0s3 10.0.6.56
vishwa@pop-os:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.6.56 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::c4c3:1e2:88b7:59c6 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:4f:0b:16 txqueuelen 1000 (Ethernet)
    RX packets 229785 bytes 272227538 (272.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 110926 bytes 15223124 (15.2 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 22715 bytes 2295158 (2.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22715 bytes 2295158 (2.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Step 2:** Launch Wireshark and select 'any' interface

```
vishwa@pop-os:~$ sudo wireshark
18:45:24.062 Main Warn QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```

**Step 3:** In terminal, type **ping 10.0.your\_section.your\_sno**

```
vishwa@pop-os:~$ ping 10.0.6.56
PING 10.0.6.56 (10.0.6.56) 56(84) bytes of data.
64 bytes from 10.0.6.56: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 10.0.6.56: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 10.0.6.56: icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from 10.0.6.56: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.0.6.56: icmp_seq=5 ttl=64 time=0.060 ms
64 bytes from 10.0.6.56: icmp_seq=6 ttl=64 time=0.042 ms
64 bytes from 10.0.6.56: icmp_seq=7 ttl=64 time=0.061 ms
64 bytes from 10.0.6.56: icmp_seq=8 ttl=64 time=0.048 ms
64 bytes from 10.0.6.56: icmp_seq=9 ttl=64 time=0.047 ms
64 bytes from 10.0.6.56: icmp_seq=10 ttl=64 time=0.062 ms
64 bytes from 10.0.6.56: icmp_seq=11 ttl=64 time=0.038 ms
64 bytes from 10.0.6.56: icmp_seq=12 ttl=64 time=0.061 ms
64 bytes from 10.0.6.56: icmp_seq=13 ttl=64 time=0.045 ms
64 bytes from 10.0.6.56: icmp_seq=14 ttl=64 time=0.044 ms
64 bytes from 10.0.6.56: icmp_seq=15 ttl=64 time=0.048 ms
64 bytes from 10.0.6.56: icmp_seq=16 ttl=64 time=0.057 ms
64 bytes from 10.0.6.56: icmp_seq=17 ttl=64 time=0.062 ms
64 bytes from 10.0.6.56: icmp_seq=18 ttl=64 time=0.074 ms
64 bytes from 10.0.6.56: icmp_seq=19 ttl=64 time=0.053 ms
^C
--- 10.0.6.56 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18418ms
rtt min/avg/max/mdev = 0.037/0.051/0.074/0.009 ms
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=5/1280, ttl=64 (reply in 2)
2	0.000013873	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=5/1280, ttl=64 (request in 1)
3	1.024351825	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=6/1536, ttl=64 (reply in 4)
4	1.024360633	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=6/1536, ttl=64 (request in 3)
5	2.051608366	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=7/1792, ttl=64 (reply in 6)
6	2.051621825	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=7/1792, ttl=64 (request in 5)
7	3.071837837	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=8/2048, ttl=64 (reply in 8)
8	3.071851419	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=8/2048, ttl=64 (request in 7)
9	4.096004801	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=9/2304, ttl=64 (reply in 10)
10	4.096018233	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=9/2304, ttl=64 (request in 9)
11	5.120668755	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=10/2560, ttl=64 (reply in 12)
12	5.120683140	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=10/2560, ttl=64 (request in 11)
13	6.147541528	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=11/2816, ttl=64 (reply in 14)

## Observations to be made

### Step 4: Analyze the following in Terminal

- TTL : 64
- Protocol used by ping : ICMP (Internet Control Message Protocol)
- Time : 18418ms

### 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.

**First echo request:**



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=5/1280, ttl=64 (reply in 2)
2	0.000013873	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=5/1280, ttl=64 (request in 1)
3	1.024351825	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=6/1536, ttl=64 (reply in 4)
4	1.024360633	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=6/1536, ttl=64 (request in 3)
5	2.051608366	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=7/1792, ttl=64 (reply in 6)
6	2.051621825	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=7/1792, ttl=64 (request in 5)
7	3.071837837	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=8/2048, ttl=64 (reply in 8)
8	3.071851419	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=8/2048, ttl=64 (request in 7)
9	4.096004801	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=9/2304, ttl=64 (reply in 10)
10	4.096018233	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=9/2304, ttl=64 (request in 9)
11	5.120668755	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=10/2560, ttl=64 (reply in 12)
12	5.120683140	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=10/2560, ttl=64 (request in 11)
13	6.147541528	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=11/2816, ttl=64 (reply in 14)

▶ Frame 2: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0  
 ▼ Linux cooked capture v1  
 Packet type: Unicast to us (0)  
 Link-layer address type: Loopback (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: 10.0.6.56, Dst: 10.0.6.56

▶ Frame 2: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0  
 ▼ Linux cooked capture v1  
 ▼ Internet Protocol Version 4, Src: 10.0.6.56, Dst: 10.0.6.56  
 0100 .... = Version: 4  
 .... 0101 = Header Length: 20 bytes (5)  
 ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
 Total Length: 84  
 Identification: 0x72f4 (29428)  
 ▶ Flags: 0x00  
 Fragment Offset: 0  
 Time to Live: 64  
 Protocol: ICMP (1)  
 Header Checksum: 0xe745 [validation disabled]  
 [Header checksum status: Unverified]

First echo reply:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=5/1280, ttl=64 (reply in 2)
2	0.000013873	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=5/1280, ttl=64 (request in 1)
3	1.024351825	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=6/1536, ttl=64 (reply in 4)
4	1.024360633	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=6/1536, ttl=64 (request in 3)
5	2.051608366	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=7/1792, ttl=64 (reply in 6)
6	2.051621825	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=7/1792, ttl=64 (request in 5)
7	3.071837837	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=8/2048, ttl=64 (reply in 8)
8	3.071851419	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=8/2048, ttl=64 (request in 7)
9	4.096004801	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=9/2304, ttl=64 (reply in 10)
10	4.096018233	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=9/2304, ttl=64 (request in 9)
11	5.120668755	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=10/2560, ttl=64 (reply in 12)
12	5.120683140	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) reply id=0x000d, seq=10/2560, ttl=64 (request in 11)
13	6.147541528	10.0.6.56	10.0.6.56	ICMP	100	Echo (ping) request id=0x000d, seq=11/2816, ttl=64 (reply in 14)

▶ Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0  
 ▼ Linux cooked capture v1  
 Packet type: Unicast to us (0)  
 Link-layer address type: Loopback (772)  
 Link-layer address length: 6  
 Source: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
 Unused: 0000  
 Protocol: IPv4 (0x0800)  
 ▶ Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0  
 ▼ Linux cooked capture v1  
 ▼ Internet Protocol Version 4, Src: 10.0.6.56, Dst: 10.0.6.56  
 0100 .... = Version: 4  
 .... 0101 = Header Length: 20 bytes (5)  
 ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
 Total Length: 84  
 Identification: 0x72f3 (29427)  
 ▶ Flags: 0x40, Don't fragment  
 Fragment Offset: 0  
 Time to Live: 64  
 Protocol: ICMP (1)  
 Header Checksum: 0xa746 [validation disabled]  
 [Header checksum status: Unverified]

Details	First Echo Request	First Echo Reply
Frame Number	1	2
Source IP address	10.0.6.56	10.0.6.56
Destination IP address	10.0.6.56	10.0.6.56
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

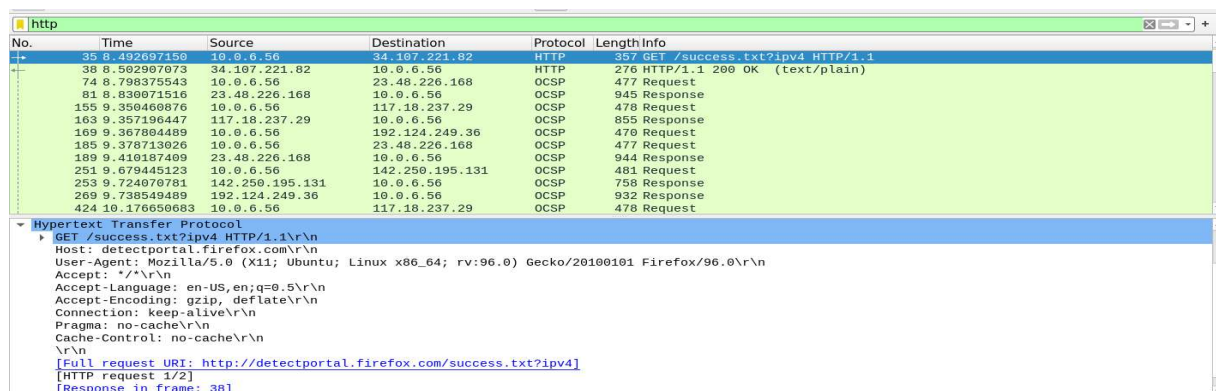
**Step 1:** Launch Wireshark and select 'any' interface. On the Filter toolbar, type-in 'http' and press enter

**Step 2:** Open Firefox browser, and browse [www.flipkart.com](http://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:

#### HTTP First Request:



No.	Time	Source	Destination	Protocol	Length	Info
35	8.492697158	10.0.6.56	34.107.221.82	HTTP	357	GET /success.txt?ip=4 HTTP/1.1
38	8.562907073	34.107.221.82	10.0.6.56	HTTP	276	HTTP/1.1 200 OK (text/plain)
74	8.798375543	10.0.6.56	23.48.226.168	OCSP	477	Request
81	8.838871516	23.48.226.168	10.0.6.56	OCSP	945	Response
155	9.350460876	10.0.6.56	117.18.237.29	OCSP	478	Request
163	9.357196447	117.18.237.29	10.0.6.56	OCSP	855	Response
169	9.367804489	10.0.6.56	192.124.249.36	OCSP	470	Request
185	9.378713026	10.0.6.56	23.48.226.168	OCSP	477	Request
189	9.410187469	23.48.226.168	10.0.6.56	OCSP	944	Response
251	9.679445123	10.0.6.56	142.250.195.131	OCSP	481	Request
253	9.724070781	142.250.195.131	10.0.6.56	OCSP	758	Response
269	9.738549489	192.124.249.36	10.0.6.56	OCSP	932	Response
424	10.176650683	10.0.6.56	117.18.237.29	OCSP	478	Request

**Hypertext Transfer Protocol**

```

GET /success.txt?ip=4 HTTP/1.1\r\n
Host: detectportal.firefox.com\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0\r\n
Accept: */*\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache\r\n
\r\n
[Full request URI: http://detectportal.firefox.com/success.txt?ip=4]
[HTTP request 1/2]
[Response in frame: 38]

```

#### HTTP First Response:

No.	Time	Source	Destination	Protocol	Length	Info
35	8.492697150	10.0.6.56	34.107.221.82	HTTP	357	GET /success.txt?ipv4 HTTP/1.1
38	8.502907073	34.107.221.82	10.0.6.56	HTTP	276	HTTP/1.1 200 OK (text/plain)
74	8.798375543	10.0.6.56	23.48.226.168	OCSP	477	Request
81	8.830971516	23.48.226.168	10.0.6.56	OCSP	945	Response
155	9.350460876	10.0.6.56	117.18.237.29	OCSP	478	Request
163	9.357196447	117.18.237.29	10.0.6.56	OCSP	855	Response
169	9.367804489	10.0.6.56	192.124.249.36	OCSP	470	Request
185	9.378713926	10.0.6.56	23.48.226.168	OCSP	477	Request
189	9.410187409	23.48.226.168	10.0.6.56	OCSP	944	Response
251	9.679445123	10.0.6.56	142.250.195.131	OCSP	481	Request
253	9.724070781	142.250.195.131	10.0.6.56	OCSP	750	Response
269	9.738549489	192.124.249.36	10.0.6.56	OCSP	932	Response
424	10.176650683	10.0.6.56	117.18.237.29	OCSP	478	Request

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Server: nginx\r\n

Content-Length: 0\r\n

Via: 1.1 google\r\n

Date: Thu, 20 Jan 2022 16:25:30 GMT\r\n

Cache-Control: public, must-revalidate, max-age=0, s-maxage=86400\r\n

Age: 77654\r\n

Content-Type: text/plain\r\n

\r\n

[HTTP response 1/2]

[Time since request: 0.010209923 seconds]

[Request in frame: 35]

[Next request in frame: 2895]

Details	First Echo Request	First Echo Reply
Frame Number	35	38
Source Port	38282	80
Destination Port	80	38282
Source IP address	10.0.6.56	34.107.221.82
Destination IP address	34.107.221.82	10.0.6.56
Source Ethernet Address	08:00:27:4f:0b:16	52:54:00:12:35:02
Destination Ethernet Address	52:54:00:12:35:02	08:00:27:4f:0b:16

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

HTTP Request		HTTP Response	
Get	GET /Success.txt?ipv4 HTTP/1.1	Server	nginx\r\n
Host	detectportal.firefox\r\n	Content-Type	text/plain\r\n

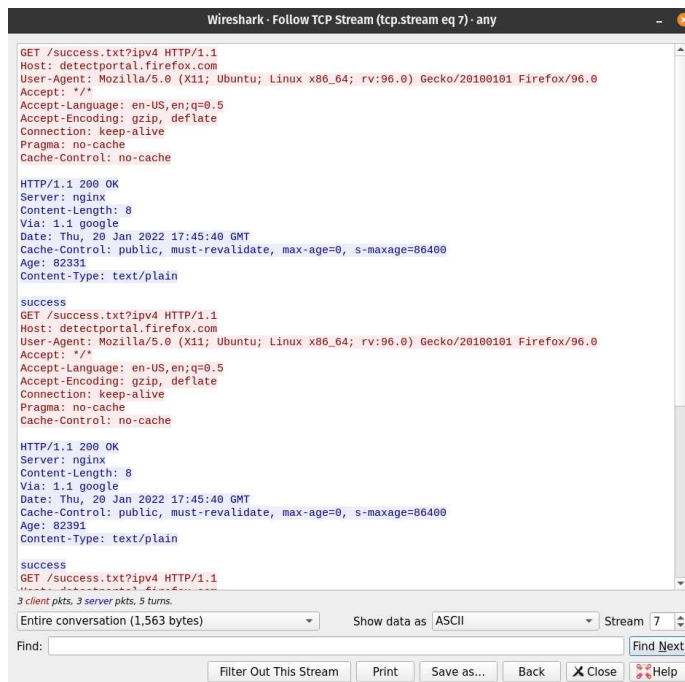
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0\r\n	Date	Thu, 20 Jan 2022 16:25:30 GMT\r\n
Accept-Language	en-US,en;q=0.5\r\n	Location	-
Accept-Encoding	gzip, deflate\r\n	Content-Length	8\r\n
Connection	keep-alive\r\n	Connection	close\r\n

### 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`

```
vishwa@pop-os:~$ sudo tcpdump -D
[sudo] password for vishwa:
1.enp0s3 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
vishwa@pop-os:~$
```

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

`sudo tcpdump -i any`

**Note:** Perform some pinging operation while giving above command. Also type [www.google.com](http://www.google.com) in browser.

```

vishwa@pop-os: ~
vishwa@pop-os: ~
vishwa@pop-os:~$ sudo tcpdump -i any
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
22:17:31.016789 lo      In  IP localhost.33932 > localhost.domain: 44882+ [1au] A? www.google.com. (43)
22:17:31.016870 lo      In  IP localhost.33932 > localhost.domain: 29535+ [1au] AAAA? www.google.com. (43)
22:17:31.017384 lo      In  IP localhost.domain > localhost.33932: 44882 1/0/1 A 142.250.196.36 (59)
22:17:31.017550 enp0s3 Out IP pop-os.54530 > dsldevice.lan.domain: 10942+ AAAA? www.google.com. (32)
22:17:31.026824 enp0s3 In  IP dsldevice.lan.domain > pop-os.54530: 10942 1/0/0 AAAA 2404:6800:4007:82a::2004 (60)
22:17:31.027126 lo      In  IP localhost.domain > localhost.33932: 29535 1/0/1 AAAA 2404:6800:4007:82a::2004 (71)
22:17:31.028299 enp0s3 Out IP pop-os > maa03s45-in-f4.1e100.net: ICMP echo request, id 15, seq 1, length 64
22:17:31.039882 enp0s3 In  IP maa03s45-in-f4.1e100.net > pop-os: ICMP echo reply, id 15, seq 1, length 64
22:17:31.040157 lo      In  IP localhost.40658 > localhost.domain: 60020+ [1au] PTR? 36.196.250.142.in-addr.arpa. (56)
22:17:31.040647 lo      In  IP localhost.domain > localhost.40658: 60020 1/0/1 PTR maa03s45-in-f4.1e100.net. (94)
22:17:31.064951 lo      In  IP localhost.37017 > localhost.domain: 6739+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
22:17:31.065214 lo      In  IP localhost.domain > localhost.37017: 6739$ 1/0/1 PTR localhost. (75)
22:17:31.065554 lo      In  IP localhost.58174 > localhost.domain: 61948+ [1au] PTR? 1.1.168.192.in-addr.arpa. (53)
22:17:31.065835 enp0s3 Out IP pop-os.54759 > dsldevice.lan.domain: 35552+ PTR? 1.1.168.192.in-addr.arpa. (42)
22:17:31.068198 enp0s3 In  IP dsldevice.lan.domain > pop-os.54759: 35552 1/0/0 PTR dsldevice.lan. (69)
22:17:31.068527 lo      In  IP localhost.domain > localhost.58174: 61948 1/0/1 PTR dsldevice.lan. (80)
22:17:31.068987 lo      In  IP localhost.47466 > localhost.domain: 17002+ [1au] PTR? 56.6.0.10.in-addr.arpa. (51)
22:17:31.069235 enp0s3 Out IP pop-os.43651 > dsldevice.lan.domain: 465+ PTR? 56.6.0.10.in-addr.arpa. (40)

```

```

vishwa@pop-os: ~
vishwa@pop-os: ~
vishwa@pop-os:~$ ping www.google.com
PING www.google.com (142.250.196.36) 56(84) bytes of data:
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=1 ttl=58 time=11.6 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=2 ttl=58 time=12.5 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=3 ttl=58 time=13.0 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=4 ttl=58 time=14.3 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=5 ttl=58 time=16.9 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=6 ttl=58 time=15.1 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=7 ttl=58 time=13.6 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=8 ttl=58 time=12.0 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=9 ttl=58 time=14.3 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=10 ttl=58 time=12.4 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=11 ttl=58 time=14.6 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=12 ttl=58 time=15.3 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=13 ttl=58 time=11.9 ms
64 bytes from maa03s45-in-f4.1e100.net (142.250.196.36): icmp_seq=14 ttl=58 time=10.1 ms
^C
--- www.google.com ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13032ms
rtt min/avg/max/mdev = 10.105/13.398/16.904/1.734 ms
vishwa@pop-os:~$

```

## 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**



**Step 1:** Run the traceroute using the following command.  
**sudo traceroute www.google.com**

```
vishwa@pop-os: ~  
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets  
1 _gateway (10.0.2.2) 0.549 ms 0.458 ms 0.409 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 * * *  
vishwa@pop-os:~$
```

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

**Destination: 142.250.196.36      No. of hops: 30**

**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**



```
vishwa@pop-os: ~  
vishwa@pop-os:~$ sudo traceroute -n www.google.com  
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets  
1  10.0.2.2  0.455 ms  0.178 ms  0.370 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 * * *
```

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

**`sudo traceroute -I www.google.com`**

```
vishwa@pop-os:~$ sudo traceroute -I www.google.com  
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets  
1  _gateway (10.0.2.2)  0.475 ms  0.409 ms  0.366 ms  
2  * dsldevice.lan (192.168.1.1)  7.324 ms  7.283 ms  
3  223.178.56.1 (223.178.56.1)  8.212 ms  8.172 ms  8.133 ms  
4  nsg-corporate-101.95.187.122.airtel.in (122.187.95.101)  7.090 ms  7.051 ms  8.009 ms  
5  * * *  
6  72.14.216.192 (72.14.216.192)  26.974 ms  14.770 ms  14.638 ms  
7  216.239.43.131 (216.239.43.131)  12.081 ms  12.659 ms  12.518 ms  
8  142.251.55.29 (142.251.55.29)  11.497 ms  12.349 ms  12.253 ms  
9  maa03s45-in-f4.1e100.net (142.250.196.36)  12.919 ms  12.837 ms  11.994 ms  
vishwa@pop-os:~$
```

**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`**

```
vishwa@pop-os:~$ sudo traceroute -T www.google.com  
traceroute to www.google.com (142.250.196.36), 30 hops max, 60 byte packets  
1  _gateway (10.0.2.2)  1.271 ms  1.203 ms  1.157 ms  
2  maa03s45-in-f4.1e100.net (142.250.196.36)  10.360 ms  12.256 ms  18.408 ms  
vishwa@pop-os:~$
```



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

**Step 1:** You can scan a host using its host name or IP address, for instance. `nmap www.pes.edu`

```
vishwa@pop-os:~$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-21 22:53 IST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.032s 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 10.52 seconds
vishwa@pop-os:~$
```

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

`nmap 163.53.78.128`

```
vishwa@pop-os:~$ nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-21 22:54 IST
Nmap scan report for 163.53.78.128
Host is up (0.014s 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 6.32 seconds
vishwa@pop-os:~$
```

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

`nmap 192.168.1.1 192.168.1.2 192.168.1.3`

```
vishwa@pop-os:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-21 22:55 IST
Nmap scan report for dsldevice.lan (192.168.1.1)
Host is up (0.0037s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 3 IP addresses (1 host up) scanned in 150.48 seconds
vishwa@pop-os:~$
```

## 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`)

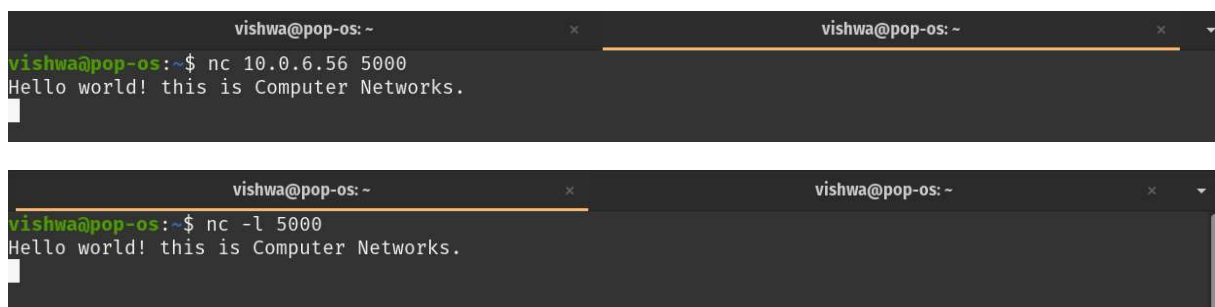
Note: It will goto listening mode

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

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

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



The image contains two terminal window screenshots. The top screenshot shows a terminal window titled 'vishwa@pop-os: ~' where the command `nc 10.0.6.56 5000` has been entered, and the output is 'Hello world! this is Computer Networks.'. The bottom screenshot shows another terminal window titled 'vishwa@pop-os: ~' where the command `nc -l 5000` has been entered, and the output is 'Hello world! this is Computer Networks.'.

### Questions on above observations:

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

**HTTP 1.1 for browser and server.**

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

**Thu, 20 Jan 2022 16:25:30 GMT\r\n**

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

**ping ip -v no\_of\_pings**

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

**nmap -O -v ip\_addr**