

# CSN-341 Computer Networks

27.11.2022

---

Akshat Agarwal

20113018

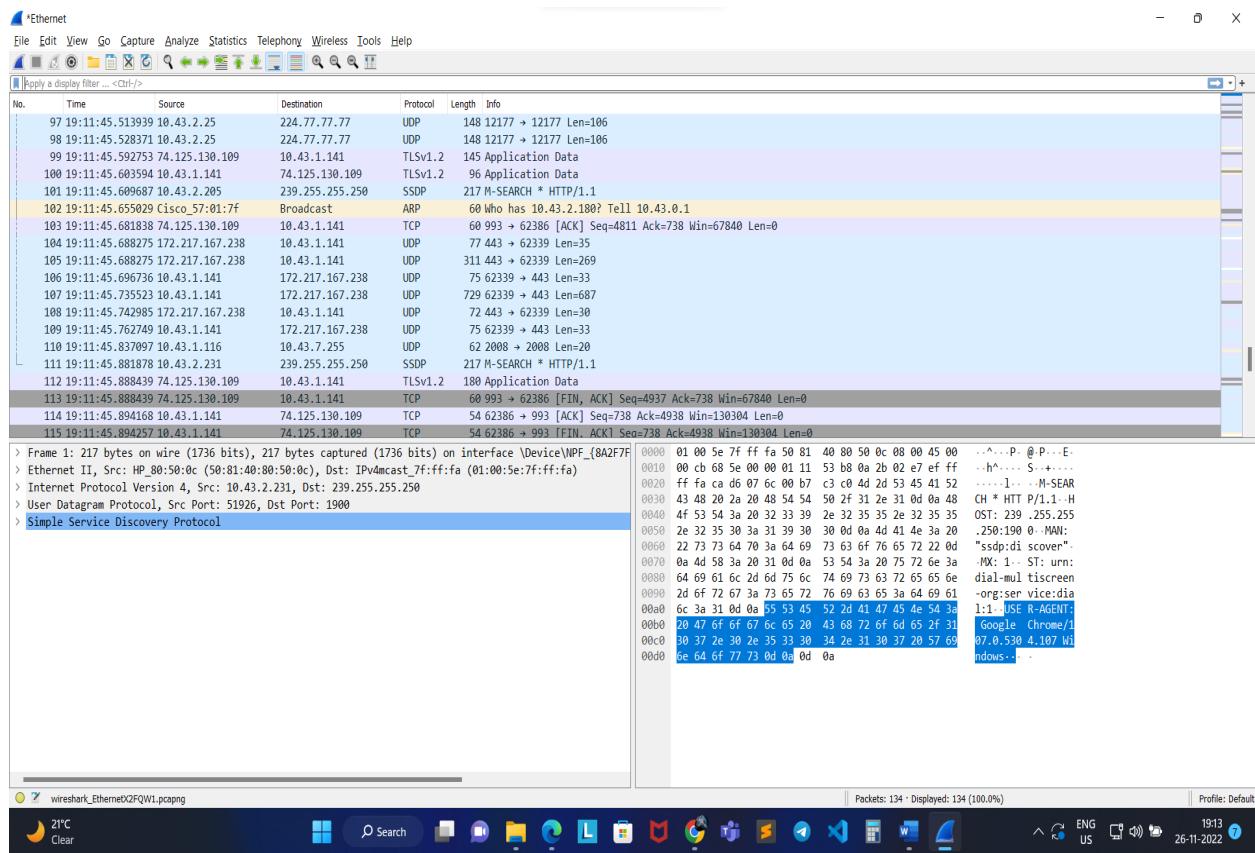
Computer Science and Engineering

Indian Institute of Technology, Roorkee

# Task 1 : Wireshark

# LAB : Getting Started

### Ques1.)



The protocols appearing in the packets are:

- 1.) UDP
  - 2.) TCP
  - 3.) SSDP
  - 4.) ARP
  - 5.) TLSv1.2

## Ques2.)

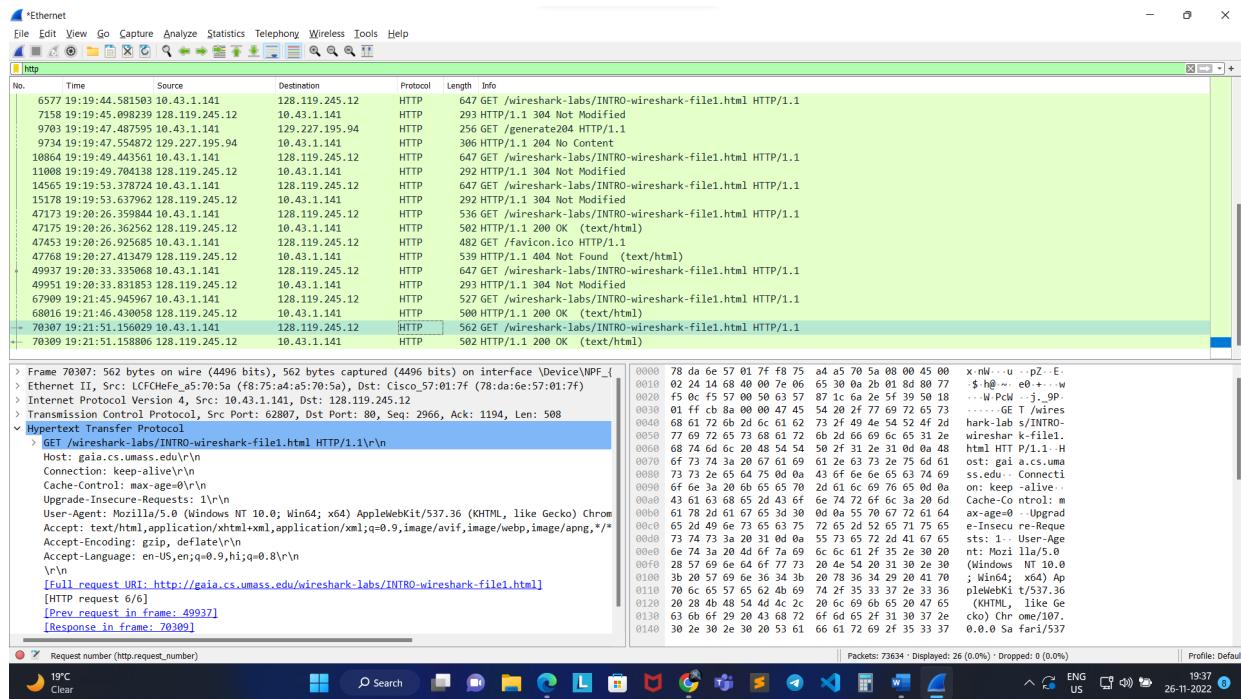
The get request was sent at time t = 19:21:51.156029 and the HTTP OK message was received at t = 19:21:51:158806.

Therefore the difference is : 0.002777 seconds

## Ques3.)

The Internet address of the gaia.cs.umass.edu is 128.119.245.12 and the Internet address of my computer is 10.43.1.141.

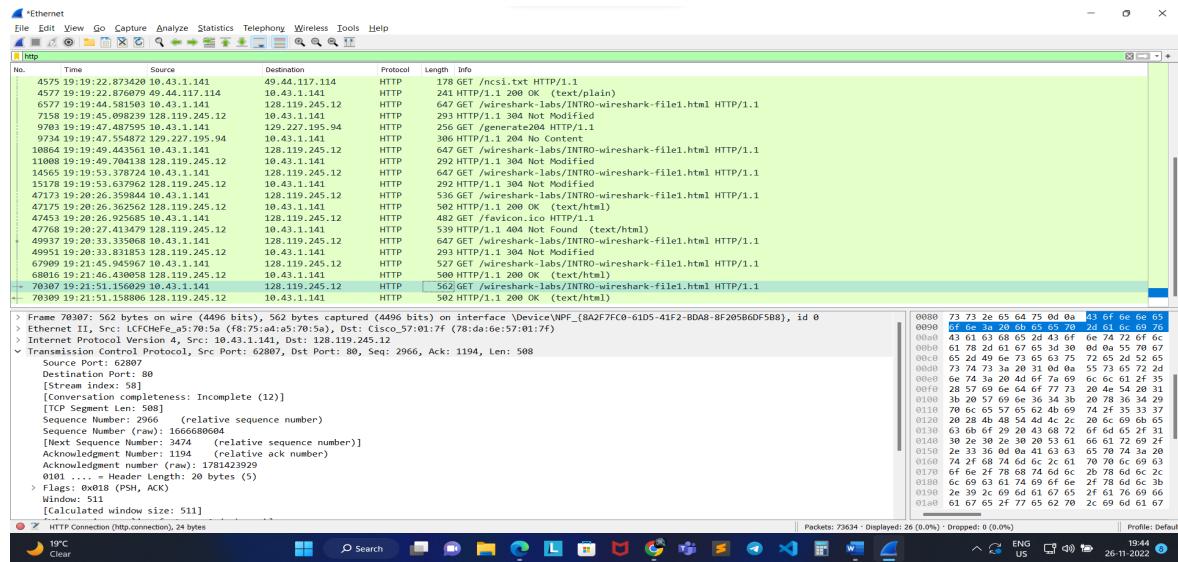
## Ques4.)



The User-Agent Field has the value :

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36\r\n
```

## Ques5.) The destination port is 80.



## Ques6.) HTTP GET and REPLY request on print command.

```
No. Time Source Destination Protocol Length Info
70307 19:21:51.156029 10.43.1.141 128.119.245.12 HTTP 562 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
Frame 70307: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits) on interface \Device\NPF_{8A2F7FC0-61D5-41F2-BDA8-8F205B6DF5B8}, id 0
Ethernet II, Src: LCFCHeFe_a5:70:5a (f8:75:a4:a5:70:5a), Dst: Cisco_57:01:7f (78:da:6e:57:01:7f)
Internet Protocol Version 4, Src: 10.43.1.141, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 62807, Dst Port: 80, Seq: 2966, Ack: 1194, Len: 508
    Source Port: 62807
    Destination Port: 80
    [Stream index: 58]
    [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 508]
    Sequence Number: 2966 (relative sequence number)
    Sequence Number (raw): 1666680604
    [Next Sequence Number: 3474 (relative sequence number)]
    Acknowledgment Number: 1194 (relative ack number)
    Acknowledgment number (raw): 1781423929
    0x01 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 511
    [Calculated window size: 511]
    [Window size scaling factor: -1 (unknown)]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    TCP payload (508 bytes)
Hypertext Transfer Protocol
No. Time Source Destination Protocol Length Info
70309 19:21:51.158806 128.119.245.12 10.43.1.141 HTTP 502 HTTP/1.1 200 OK (text/html)
Frame 70309: 502 bytes on wire (4016 bits), 502 bytes captured (4016 bits) on interface \Device\NPF_{8A2F7FC0-61D5-41F2-BDA8-8F205B6DF5B8}, id 0
Ethernet II, Src: Cisco_57:01:7f (78:da:6e:57:01:7f), Dst: LCFCHeFe_a5:70:5a (f8:75:a4:a5:70:5a)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.43.1.141
Transmission Control Protocol, Src Port: 80, Dst Port: 62807, Seq: 1194, Ack: 3474, Len: 448
    Source Port: 80
    Destination Port: 62807
    [Stream index: 58]
    [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 448]
    Sequence Number: 1194 (relative sequence number)
    Sequence Number (raw): 1781423929
    [Next Sequence Number: 1642 (relative sequence number)]
    Acknowledgment Number: 3474 (relative ack number)
    Acknowledgment number (raw): 1666681112
    0x01 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 955
    [Calculated window size: 955]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xa0d7 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    TCP payload (448 bytes)
Hypertext Transfer Protocol
Line-based text data: text/html (3 lines)
```

## LAB : HTTP

Ques1.)

My browser is running at HTTP 1.1 .

Ques2.)

**Accept-Language: en-US,en;q=0.9\r\n**

Ques3.)

The Internet address of the gaia.cs.umass.edu is 128.119.245.12 and the Internet address of my computer is 10.43.1.141.

Ques4.)

The status code returned from the browser is 200 OK .

Ques5.)

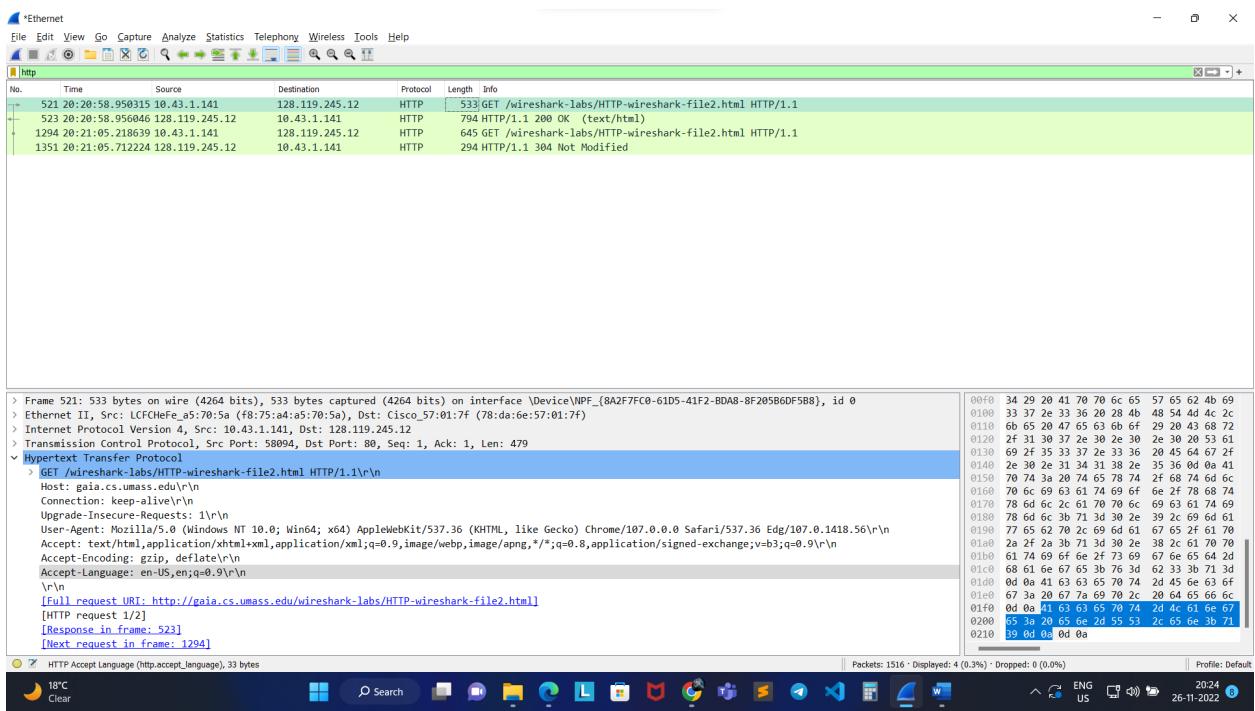
**Last-Modified: Sat, 26 Nov 2022 06:59:01 GMT\r\n**

Ques6.)

File data : 128 bytes.

Ques7.)

No, I don't see any in the HTTP Message below.



Ques8.)

There is no IF-MODIFIED-SINCE in the first GET.

Ques9.)

Yes, the server explicitly return the file from the server to the browser.

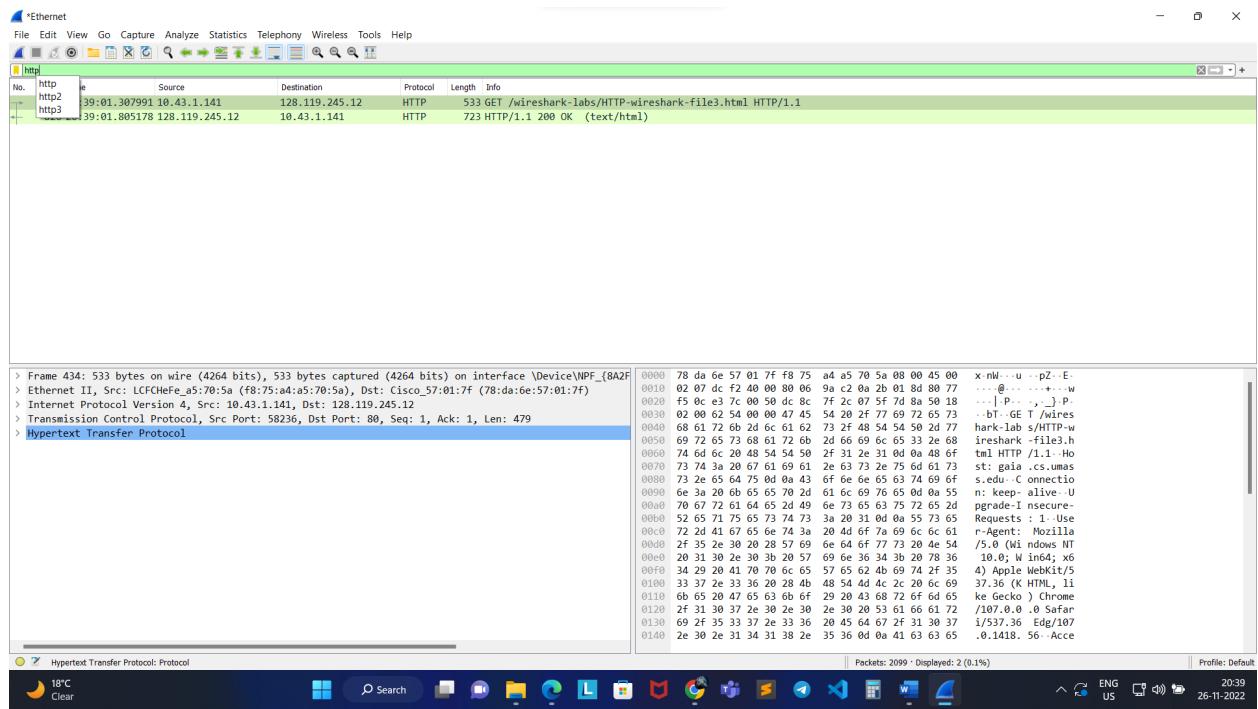
Ques10.)

Yes, there is an option of "IF-MODIFIED-SINCE" in the second HTTP GET request.

**If-Modified-Since: Sat, 26 Nov 2022 06:59:01 GMT\r\n**

Ques11.)

Status code received : 304 NOT MODIFIED. No, the server did not actually return the file.



## Ques12.)

Only 1 HTTP GET request is sent by the browser. The packet number 434 in trace of Wireshark contains the GET request.

## Ques13.)

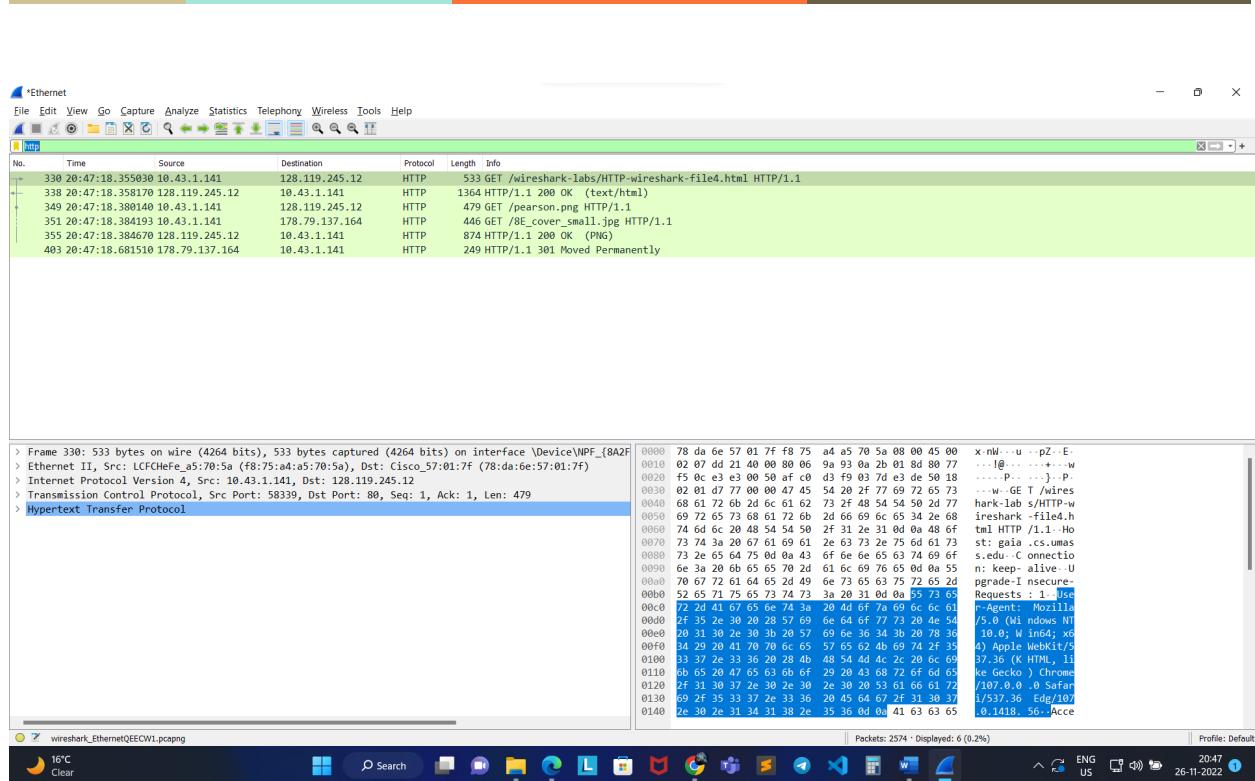
The packet number containing the status code and phrase associated is 132.

## Ques14.)

Status code : 200 Phrase code : OK

## Ques15.)

Number of data carrying packets : 4



Ques16.)

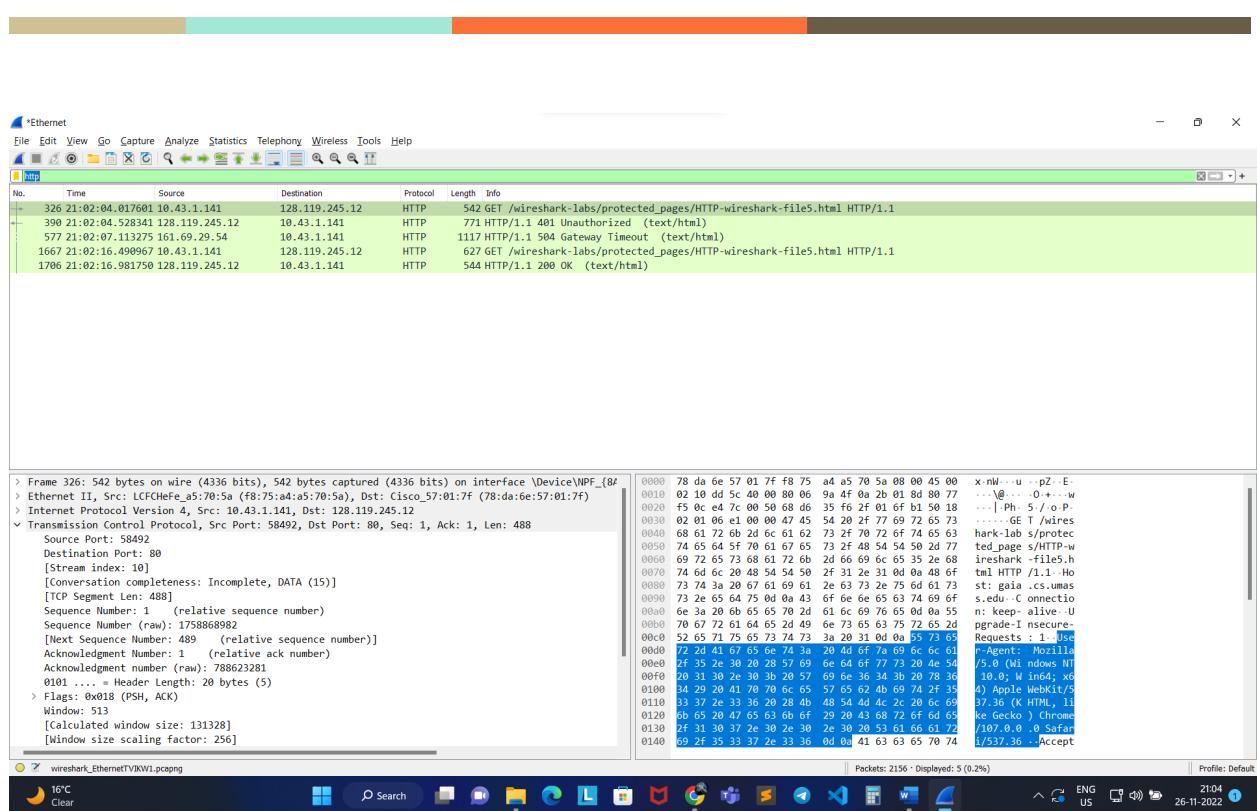
3 HTTP GET requests are sent by the browser. Two to the same destination IP address 128.119.245.12 and one to 178.79.137.164 .

### Ques17.)

The downloads have occurred in parallel.

Sequence numbers of GET request : 349 and 351.

Sequence numbers of RESPONSE message : 355 and 403.



Ques18.)

The message received by the first GET message is : 401 UNAUTHORIZED

Ques19.)

The HTTP GET message now contains an additional field

Authorization: Basic d2lyZXNoYXJrLXN0dWR1bnRzOm5ldHdvcms=\r\n

## LAB : DNS

```
C:\ Command Prompt

C:\Users\Akshat Agarwal>nslookup www.iitb.ac.in
Server: umbva01.iitr.ac.in
Address: 192.168.108.121

Non-authoritative answer:
Name: www.iitb.ac.in
Address: 103.21.124.10

C:\Users\Akshat Agarwal>nslookup -type=NS iitb.ac.in
Server: umbva01.iitr.ac.in
Address: 192.168.108.121

iitr.ac.in
    primary name server = iitrdcad01.iitr.ac.in
    responsible mail addr = hostmaster.iitr.ac.in
    serial = 15522
    refresh = 900 (15 mins)
    retry = 600 (10 mins)
    expire = 86400 (1 day)
    default TTL = 3600 (1 hour)

C:\Users\Akshat Agarwal>nslookup iitrdcad01.iitr.ac.in
Server: umbva01.iitr.ac.in
Address: 192.168.108.121

Non-authoritative answer:
Name: iitrdcad01.iitr.ac.in
Address: 192.168.101.11

C:\Users\Akshat Agarwal>
```

Ques1.) The IP address of [www.iitb.ac.in](http://www.iitb.ac.in) : 103.21.124.10 .

Ques2.) The IP address of the DNS server : 192.168.108.121

Ques3.) The answer comes from a non-authoritative answer.

Ques4.) Authoritative name server : iitrdcad01.iitr.ac.in . To find the IP address of an authoritative name server, we would run nslookup authoritative\_servername.

```
C:\ Command Prompt

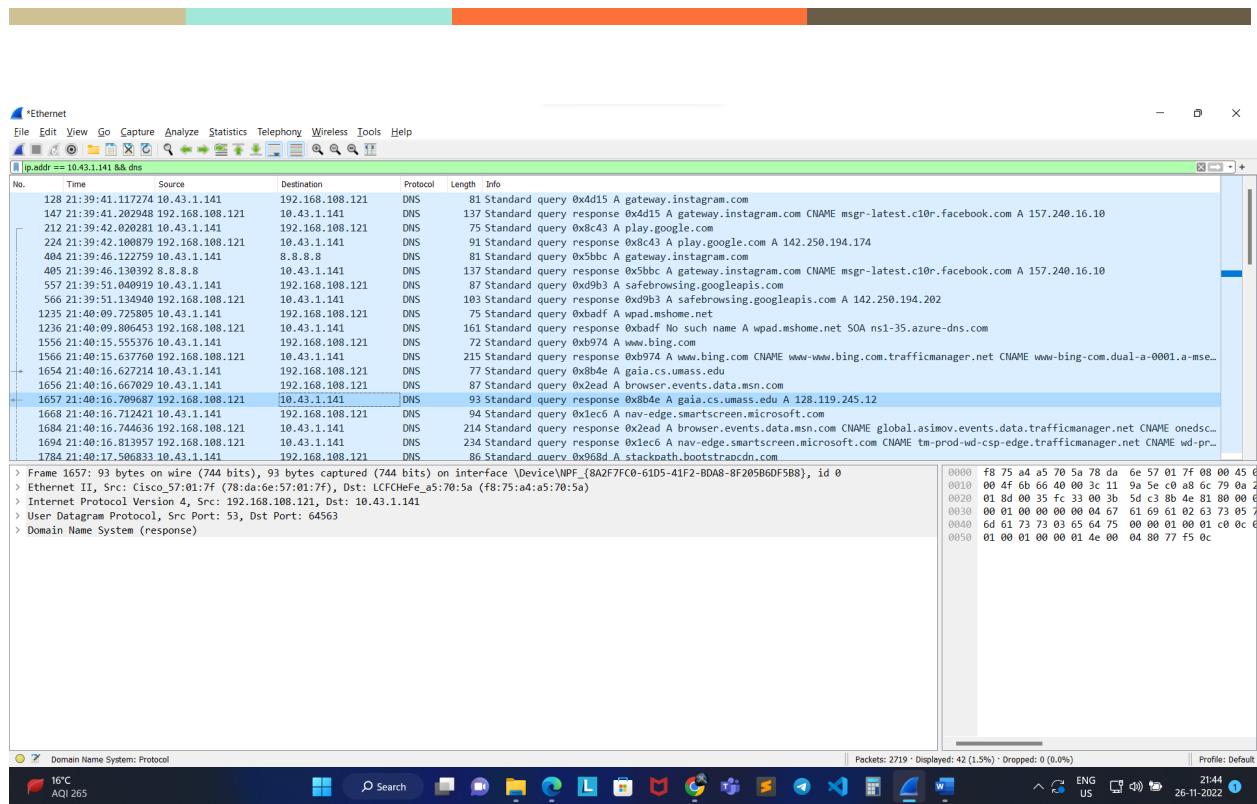
C:\Users\Akshat Agarwal>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\Akshat Agarwal>
```

Clearing the DNS cache.



Ques5.) The packet number for the first DNS to gaia.cs.umass.edu is 1654. The query is sent over UDP.

Ques6.) The packet number for the receiver packet is 1657. The response is received using the UDP protocol.

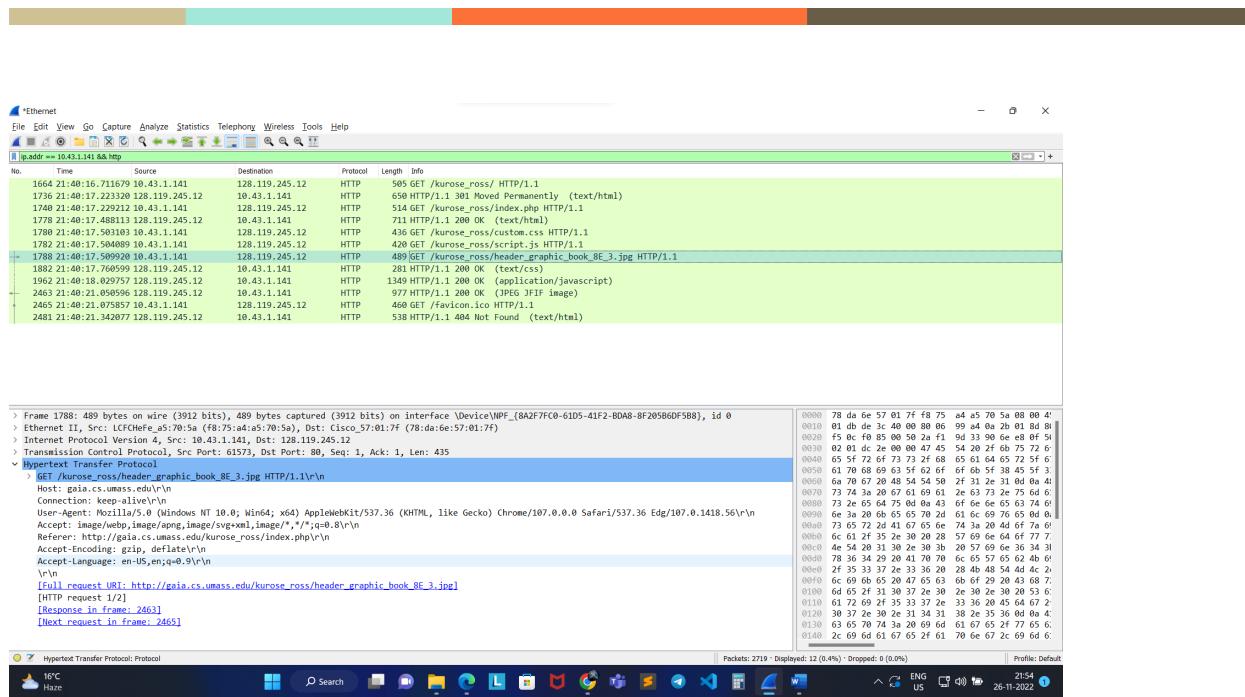
Ques7.) DNS query message: UDP Src port : 64563 Dest port : 53

DNS response message: UDP Src port : 53 Dest port : 64563

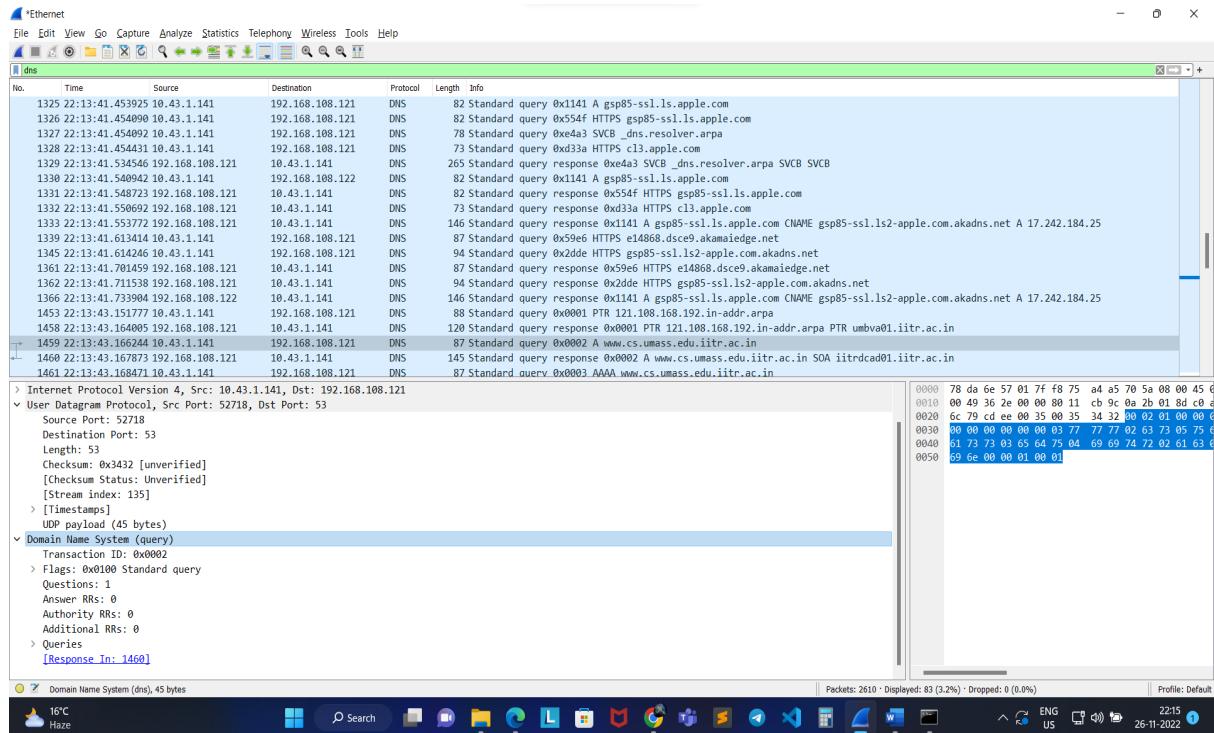
Ques8.) The DNS query is sent to 192.168.108.121 .

Ques9.) The DNS query message contains 1 question and 0 answers.

Ques10.) The DNS response message contains 1 question and 1 answer.



**Ques11.)** The DNS packet number in the trace of DNS query made to resolve gaia.cs.umass.edu so that this initial HTTP request can be sent to the gaia.cs.umass.edu IP address is 1658. The packet number of the DNS query is 1667. For the image file, the packet number is 1788. No additional query is sent to the DNS server for getting the IP address. Instead the DNS cache is used. The DNS cache tells the IP address of the address. No additional DNS query is required.

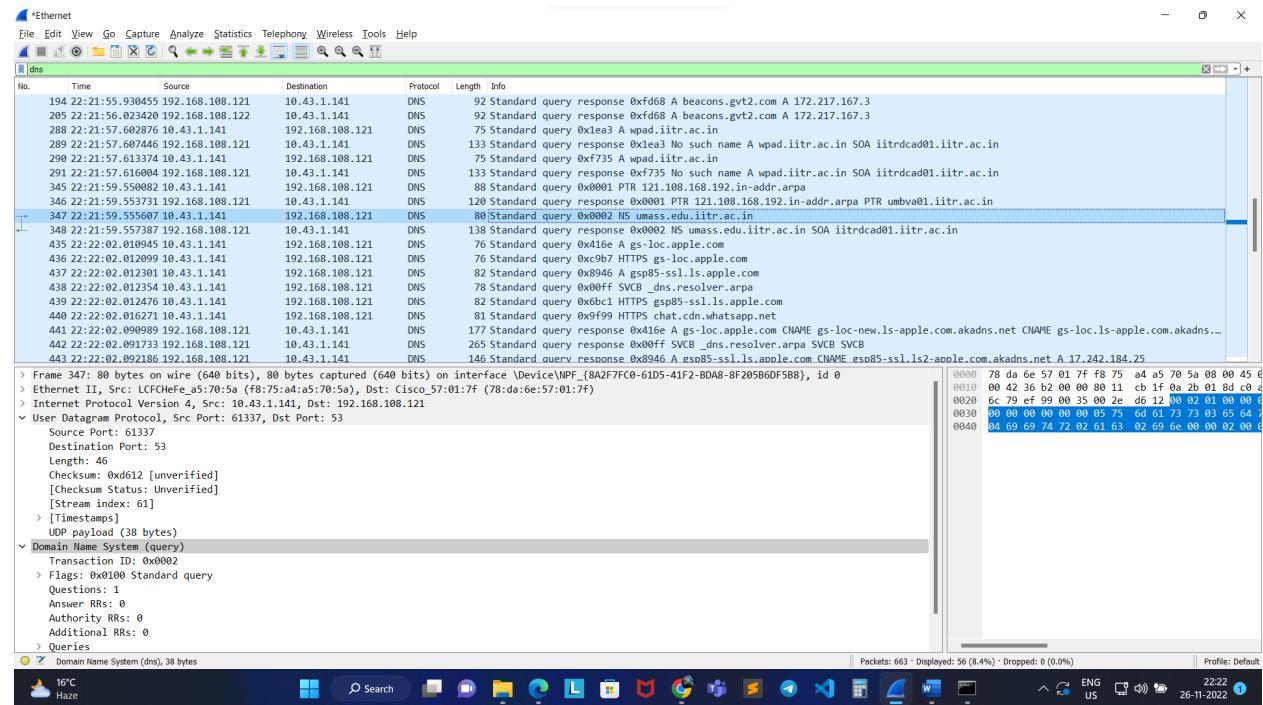


Ques12.) Destination port of the DNS query message is 53. The source port is : 52720 .

Ques13.) The DNS query message is sent to 192.168.108.121. This is the IP address of my default local DNS server.

Ques14.) Type of DNS query : A . No, this query does not contain any answers.

Ques15.) The DNS response message to the query message contains 1 question and 1 answer.

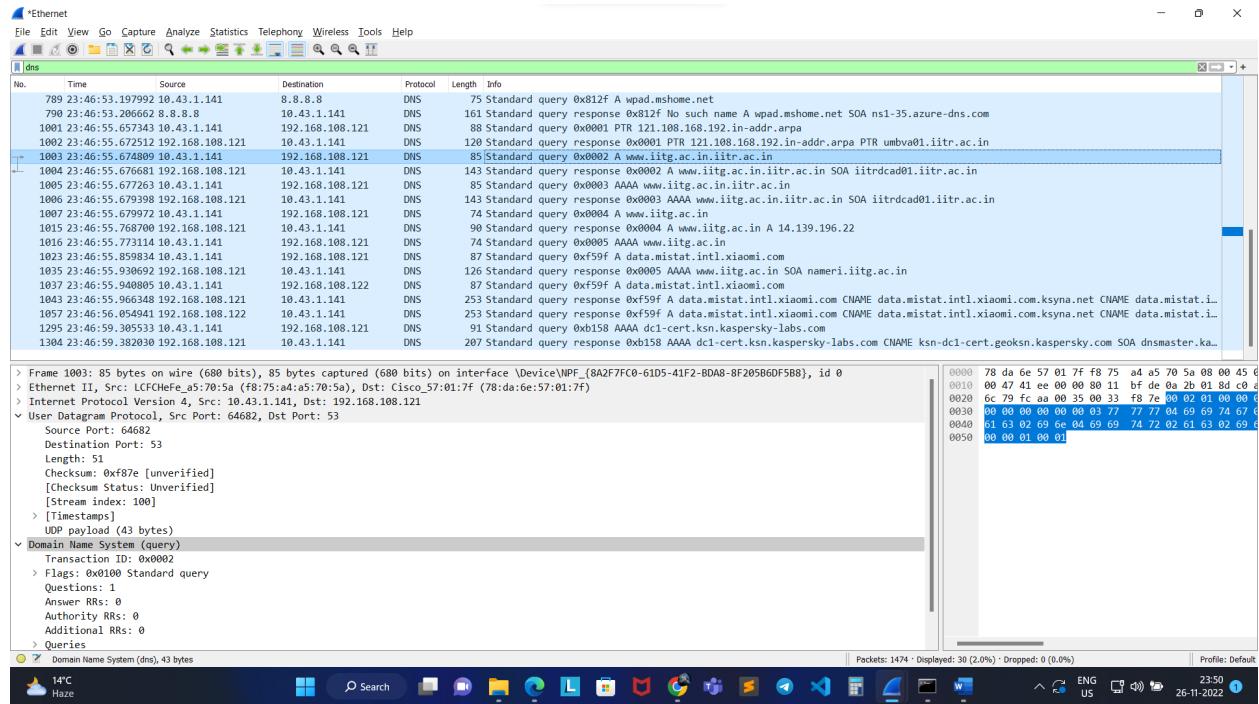


Ques16.) The DNS query is sent to 192.168.108.121 and this is the IP address of my local DNS server.

Ques17.) The DNS query has 1 question and 0 answers.

Ques18.) The DNS response in this case has 0 answers.

## LAB : UDP



Ques1.) For the first UDP segment we have :

- 1.) The packet number in the first UDP segment is 1003.
- 2.) The UDP packet is carrying DNS application layer protocol.
- 3.) There are 4 fields in this UDP header.

**User Datagram Protocol, Src Port: 64682, Dst Port: 53**

**Source Port: 64682**

**Destination Port: 53**

**Length: 51**

**Checksum: 0xf87e [unverified]**

**[Checksum Status: Unverified]**

**[Stream index: 100]**

**> [Timestamps]**

**UDP payload (43 bytes)**

Ques2.)

- 1.) Source port length = 2 bytes
- 2.) Destination port length = 2 bytes
- 3.) Length field = 2 bytes
- 4.) Checksum length = 2 bytes

Ques3.) Length is the length of the UDP packet including both header and length.

Ques4.) There are 16 bits available for length. So the maximum length can be  $2^{16} - 1$ . In these 8 bytes will be for UDP header and network layer adds a minimum of 20 bytes of header length. So the maximum payload is 65507 bytes.

Ques5.) Largest possible port number :  $2^{16} - 1$  ;

Ques6.) The protocol number of UDP is 17.

Ques7.)

Packet number for query : 1003

Source port number : 64682

Destination port number : 53

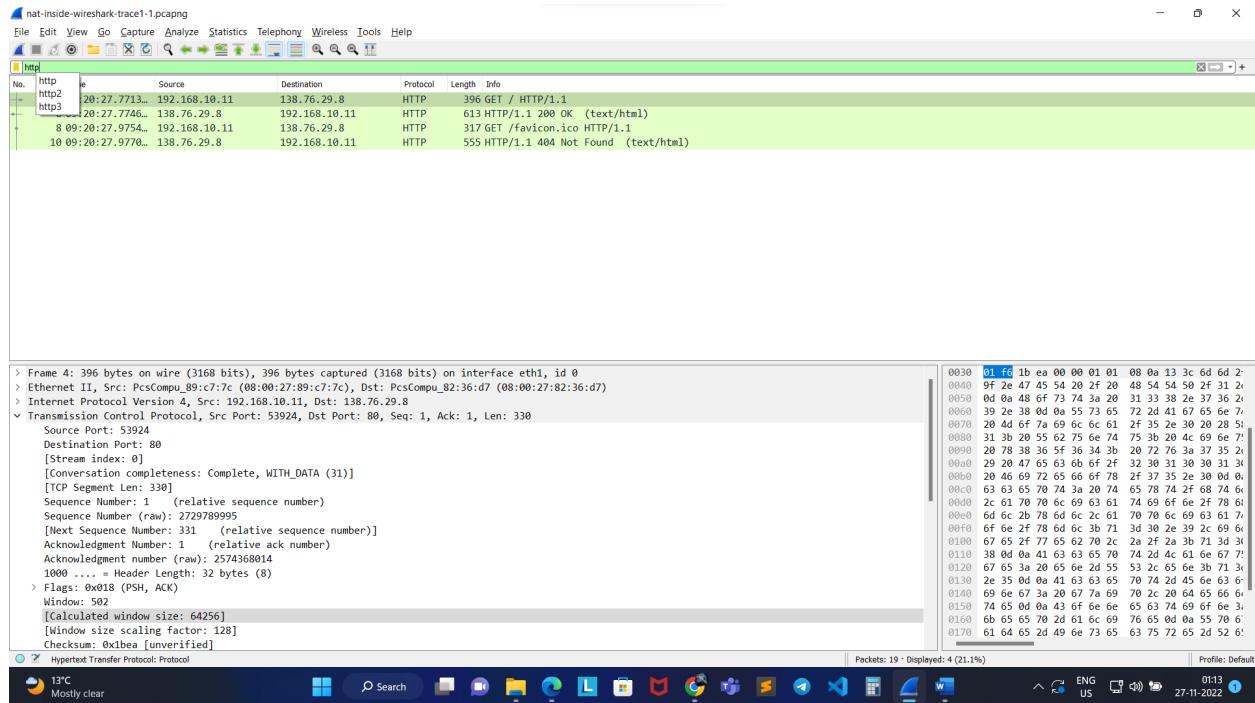
Packet number for response : 1004

Source port number : 53

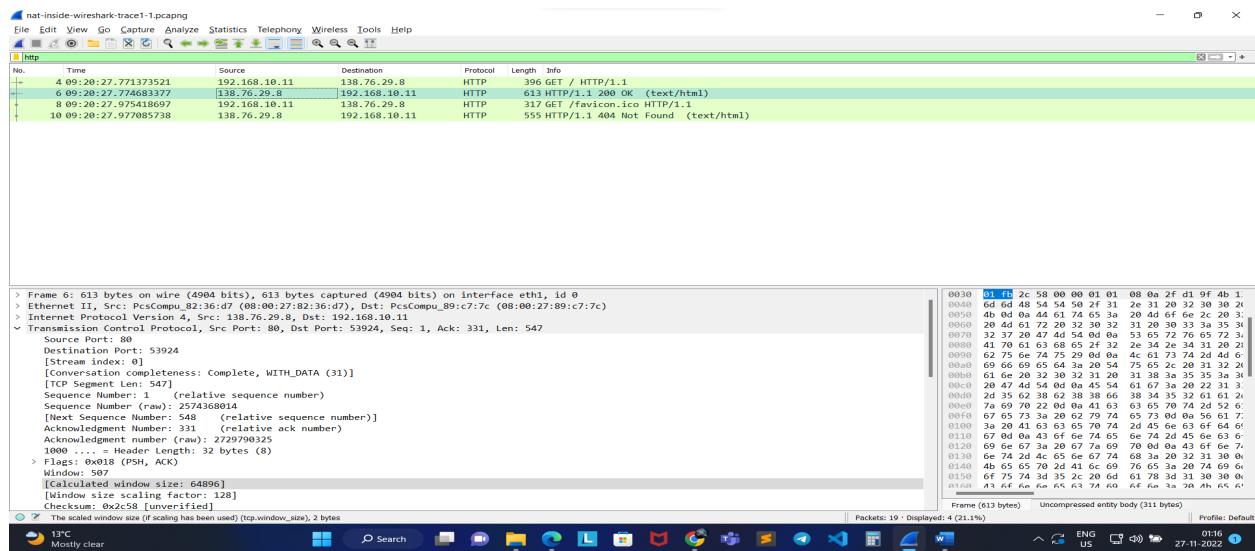
Destination port number : 64682

## LAB : NAT

Ques1.) The IP address of the client that sends the HTTP GET request is 192.168.10.11 . The source port in the TCP segment is 53924. Destination IP address is : 138.76.29.8. Destination port number is 80.



Ques2.) The difference is : 0.030672101



### Ques3.)

- 1.) Source IP Address : 138.76.29.8 .
- 2.) Destination IP Address : 192.168.10.11
- 3.) Source Port : 80
- 4.) Destination Port number : 53924

**Screenshot 1:** Wireshark capture of network traffic showing the initial HTTP request and response. The request (Frame 4) is a GET to port 53924. The response (Frame 6) is a 200 OK. The request (Frame 8) is a GET for the favicon.ico file.

No.	Time	Source	Destination	Protocol	Length	Info
4	09:20:27.771391145	10.0.1.254	138.76.29.8	HTTP	396	GET / HTTP/1.1
6	09:20:27.774660820	138.76.29.8	10.0.1.254	HTTP	613	HTTP/1.1 200 OK (text/html)
8	09:20:27.975435044	10.0.1.254	138.76.29.8	HTTP	317	GET /favicon.ico HTTP/1.1
10	09:20:27.977078167	138.76.29.8	10.0.1.254	HTTP	555	HTTP/1.1 404 Not Found (text/html)

**Screenshot 2:** Detailed view of the captured frames. Frame 4 shows the raw bytes and hex dump. Frame 6 shows the decompressed entity body (311 bytes).

**Screenshot 3:** Another Wireshark capture of the same session, showing the same frames and details.

Ques4.) At t = 0.027356291s, the HTTP GET message appears in NAT package capture.

Ques5.)

- 1.) Source IP Address : 10.0.1.254
- 2.) Destination IP Address : 138.76.29.8
- 3.) Source Port Number : 53924
- 4.) Destination Port Number : 80

Ques6.) The source address is different as compared to question 1.

Ques7.) No, HTTP GET message remains unchanged.

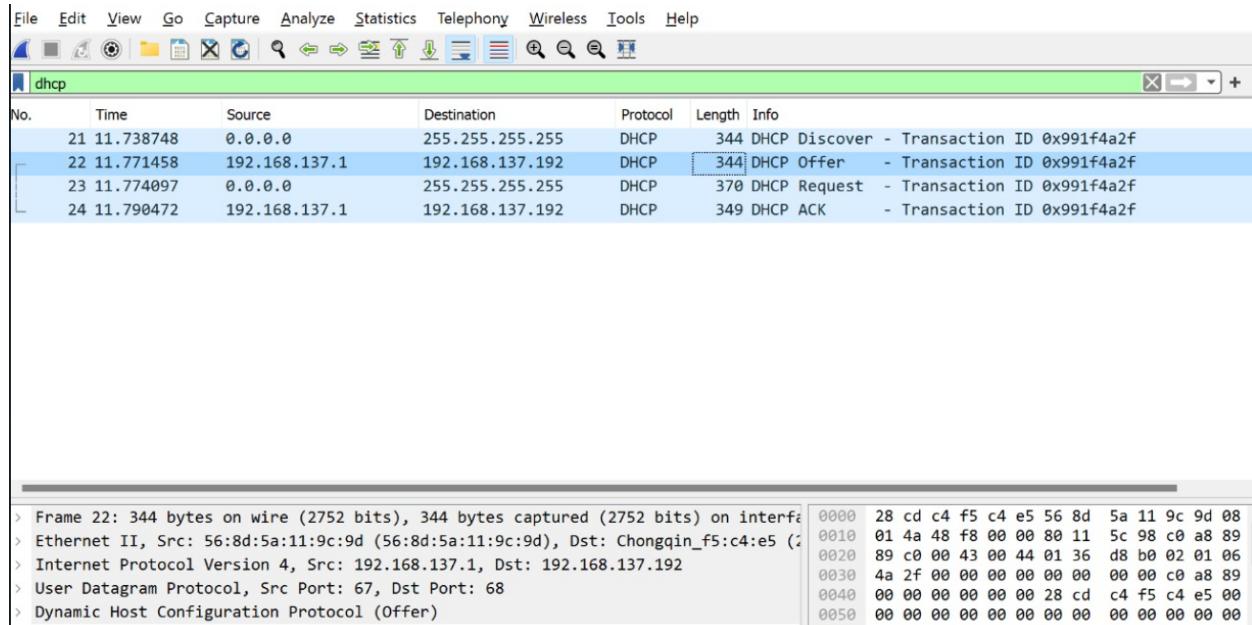
Ques8.) Version and Checksum are changed and Header length and Flags remain unchanged.

Ques9.) At t = 0.030625966, the received ACK message is received.

Ques10.)

- 1.) Source IP Address : 138.76.29.8
- 2.) Destination Address: 10.0.1.254
- 3.) Source Port : 80
- 4.) Destination Port : 53924

## LAB : DHCP



- Ques1.) The DHCP DISCOVER message is sent using the UDP protocol.
- Ques2.) Source IP = 0.0.0.0 . This address is used to designate an invalid or unknown host.
- Ques3.) Destination IP = 255.255.255.255 . This is the broadcast address.
- Ques4.) Transaction ID = 0x2ce974f3
- Ques5.) The client is requesting for following options:

Option: (53) DHCP Message Type (Discover)  
 Option: (61) Client identifier  
 Option: (50) Requested IP Address (10.43.1.141)  
 Option: (12) Host Name  
 Option: (60) Vendor class identifier  
 Option: (55) Parameter Request List  
 Option: (255) End

Ques6.) We can say that this OFFER message was in response to DHCP DISCOVER message because they have the same transaction ID.

Ques7.) The source IP of DHCP OFFER message : 192.168.137.192 .

Ques8.) The destination IP address is 255.255.255.255 .

Ques9.) It is providing following information :

- > Option: (53) DHCP Message Type (Offer)
- > Option: (1) Subnet Mask (255.255.248.0)
- > Option: (58) Renewal Time Value
- > Option: (59) Rebinding Time Value
- > Option: (51) IP Address Lease Time
- > Option: (54) DHCP Server Identifier (192.168.101.11)
- > Option: (3) Router
- > Option: (6) Domain Name Server
- > Option: (15) Domain Name
- > Option: (255) End

Ques10.) Source Port : 68 Destination Port : 67

Ques11.) Source IP = 0.0.0.0 .

Ques12.) Destination IP = 255.255.255.255 . This is the broadcast message.

Ques13.) Transaction ID : 0x2ce974f3. Yes, it will match.

Ques14.) There is no difference between the OPTIONS field between DHCP DISCOVER message and DHCP OFFER message .

Ques15.) The source IP address in the IP datagram containing the ACK message is 192.168.137.1 . This is the address of the DHCP server that sends acknowledgement messages.

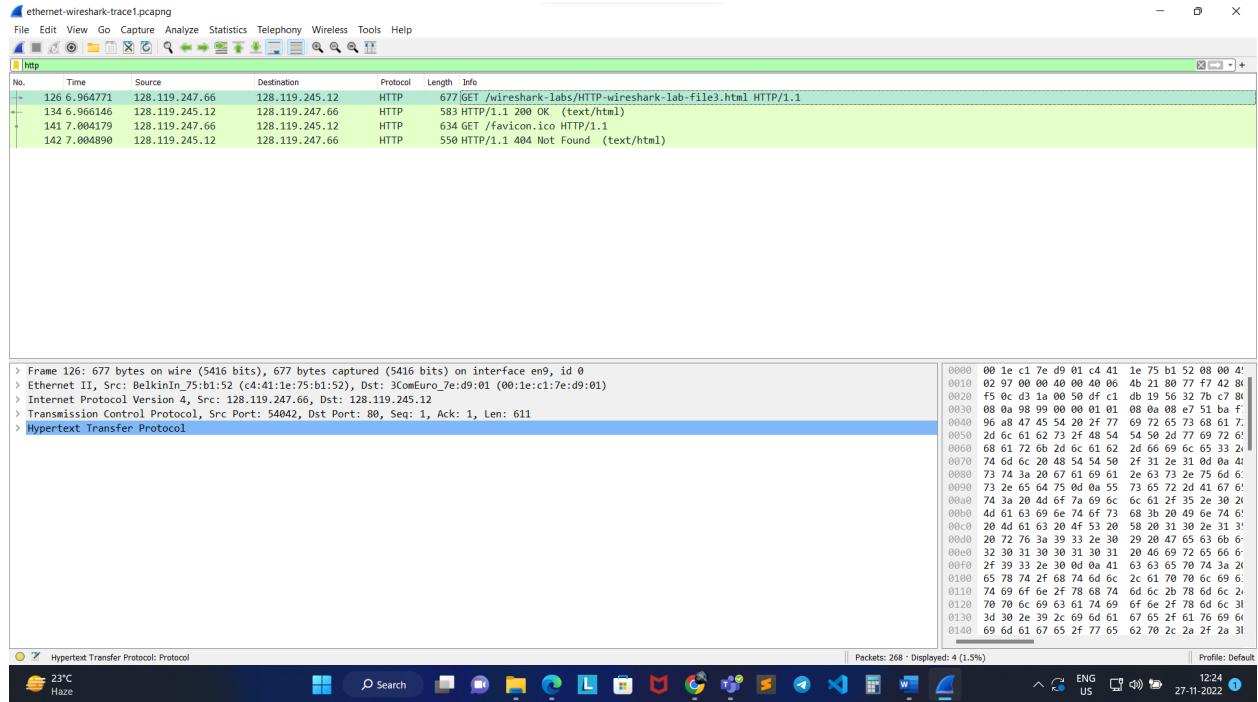
Ques16.) The destination IP : 255.255.255.255, as the client is still not assigned an IP address.

Ques17.) Is is : Your (Client) IP Address

Ques18.) The lease time is 30 minutes.

Ques19.) The first hop router has address is : 10.61.80.1

## LAB : ETHERNET



Ques1.) Ethernet Address : c4:41:1e:75:b1:52

Ques2.) Ethernet Destination Address : 00:1e:c1:7e:d9:01. This is the ethernet address of my next hop router. The ethernet frame is sent to the next router of mine, which then transports the packet off the link.

Ques3.) Type field : 0x0800 . This corresponds to the IPV4 protocol.

Ques4.) 54 bytes before the first letter.

0040	96	a8	47	45	54	20	2f	77	69	72	65	73	68	61	72	6b	- -	GET	/w ireshark
0050	2d	6c	61	62	73	2f	48	54	54	50	2d	77	69	72	65	73	- -	-labs/HT	TP-wires
0060	68	61	72	6b	2d	6c	61	62	2d	66	69	6c	65	33	2e	68	- -	hark-lab	-file3.h
0070	74	6d	6c	20	48	54	54	50	2f	31	2e	31	0d	0a	48	6f	- -	tml	HTTP /1.1.-Ho
0080	73	74	3a	20	67	61	69	61	2e	63	73	2e	75	6d	61	73	- -	st:	gaia .cs.umass.edu..U
0090	73	2e	65	64	75	0d	0a	55	73	65	72	2d	41	67	65	6e	- -	ser-Agen	t: Mozil la/5.0 (
00a0	74	3a	20	4d	6f	7a	69	6c	6c	61	2f	35	2e	30	20	28	- -	Macintosh	h; Intel Mac OS X 10.15;
00b0	4d	61	63	69	6e	74	6f	73	68	3b	20	49	6e	74	65	6c	- -	rv:93.0 )	Gecko/20100101 Firefox /93.0.-A ccept: text/html ,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.5.-A ccept-Encoding: gzip, deflate
00c0	20	4d	61	63	20	4f	53	20	58	20	31	30	2e	31	35	3b	- -		
00d0	20	72	76	3a	39	33	2e	30	29	20	47	65	63	6b	6f	2f	- -		
00e0	32	30	31	30	30	31	30	31	20	46	69	72	65	66	6f	78	- -		
00f0	2f	39	33	2e	30	0d	0a	41	63	63	65	70	74	3a	20	74	- -		
0100	65	78	74	2f	68	74	6d	6c	2c	61	70	70	6c	69	63	61	- -		
0110	74	69	6f	6e	2f	78	68	74	6d	6c	2b	78	6d	6c	2b	78	- -		
0120	70	70	6c	69	63	61	74	69	6f	6e	2f	78	6d	6c	3b	71	- -		
0130	3d	30	2e	39	2c	69	6d	61	67	65	2f	61	76	69	66	2c	- -		
0140	69	6d	61	67	65	2f	77	65	62	70	2c	2a	2f	2a	3b	71	- -		
0150	3d	30	2e	38	0d	0a	41	63	63	65	70	74	2d	4c	61	6e	- -		
0160	67	75	61	67	65	3a	20	65	6e	2d	55	53	2c	65	6e	3b	- -		
0170	71	3d	30	2e	35	0d	0a	41	63	63	65	70	74	2d	45	6e	- -		
0180	63	6f	64	69	6e	67	3a	20	67	7a	69	70	2c	20	64	65	- -		

Ques5.) Ethernet address : 00:1e:c1:7e:d9:01. This is the ethernet address of the next hop router of my computer.

Ques6.) Ethernet address of the destination : c4:41:1e:75:b1:52 . This is the ethernet address of my computer.

Ques7.) Type field : 0x0800. This corresponds to the IPv4 protocol of upper layers.

Ques8.) Here also 54 bytes from the beginning of the ethernet frame.

Ques9.) 4 ethernet frames were sent.

Ques10.)

```
Windows Command Prompt
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Akshat Agarwal>arp -a

Interface: 192.168.137.1 --- 0x6
Internet Address      Physical Address      Type
192.168.137.2          f6-6e-47-a7-dd-7d  static
192.168.137.16         16-ce-8c-c2-c6-15  static
192.168.137.56         18-3e-ef-b9-6d-e8  static
192.168.137.58         e8-5a-8b-48-e9-8d  static
192.168.137.65         66-ab-e9-13-27-df  static
192.168.137.78         be-1a-7f-87-e8-a2  static
192.168.137.79         6c-6a-77-9b-d2-d2  static
192.168.137.121        bc-17-b8-c5-d8-70  static
192.168.137.142        90-e8-68-c7-5b-a9  static
192.168.137.183        f6-26-58-4e-aa-dc  static
192.168.137.192        28-cd-c4-f5-c4-e5  static
192.168.137.197        86-3a-72-c4-e8-d9  static
192.168.137.227        90-e8-68-c7-5b-a9  static
192.168.137.251        5c-ba-ef-41-db-33  static
192.168.137.255        ff-ff-ff-ff-ff-ff  static
224.0.0.5               01-00-5e-00-00-05  static
224.0.0.22              01-00-5e-00-00-16  static
224.0.0.251             01-00-5e-00-00-fb  static
224.0.0.252             01-00-5e-00-00-fc  static
224.0.1.60              01-00-5e-00-01-3c  static
224.77.77.77            01-00-5e-4d-4d-4d  static
230.0.0.1               01-00-5e-00-00-01  static
239.192.152.143         01-00-5e-40-98-8f  static
239.255.102.18          01-00-5e-7f-66-12  static
239.255.255.250         01-00-5e-7f-ff-fa  static
255.255.255.255         ff-ff-ff-ff-ff-ff  static

Interface: 192.168.56.1 --- 0x8
Internet Address      Physical Address      Type
192.168.56.255         ff-ff-ff-ff-ff-ff  static
224.0.0.5               01-00-5e-00-00-05  static
224.0.0.22              01-00-5e-00-00-16  static
224.0.0.251             01-00-5e-00-00-fb  static
224.0.0.252             01-00-5e-00-00-fc  static
224.77.77.77            01-00-5e-4d-4d-4d  static
230.0.0.1               01-00-5e-00-00-01  static
239.255.255.250         01-00-5e-7f-ff-fa  static
```

Ques11.) Three things displayed in the ARP cache :

- 1.) IPv4 address
- 2.) MAC address
- 3.) Type

```
C:\Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>arp -d

C:\WINDOWS\system32>arp -a

Interface: 192.168.137.1 --- 0x6
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16    static
  230.0.0.1              01-00-5e-00-00-01    static

Interface: 192.168.56.1 --- 0x8
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16    static

Interface: 10.43.1.141 --- 0xc
  Internet Address      Physical Address      Type
  10.43.0.1              78-da-6e-57-01-7f    dynamic
  224.0.0.22            01-00-5e-00-00-16    static
  230.0.0.1              01-00-5e-00-00-01    static

Interface: 172.27.224.1 --- 0x38
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16    static

C:\WINDOWS\system32>
```

Ques12.) Source Address : 00:1e:c1:7e:d9:01

Ques13.) Destination address : ff:ff:ff:ff:ff:ff. This corresponds to the universal broadcast message.

Ques14.) Type field : ARP(0x0806). This corresponds to the ARP protocol.

Ques15.) 12 bytes from beginning the ARP opcode begins.

Ques16.) The value is : 0x0806 .

Ques17.) Yes, the ARP request message contains the IP address off the sender. The IP address is : 128.119.247.1 .

Ques18.) Target IP Address : 128.119.247.77

Ques19.) The value is : 0x0806 .

Ques20.) The Ethernet address corresponding to the IP address that was specified in the ARP request message sent by my computer is : 00:00:00:00:00:00.

Ques21.) Yes, we can see ARP packets from many other devices in the same network. We can see them because they were broadcasted to all the devices in the network.

## LAB : ICMP

```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>ping -n 10 www.ust.hk

Pinging www.ust.hk [143.89.12.134] with 32 bytes of data:
Reply from 143.89.12.134: bytes=32 time=105ms TTL=50
Reply from 143.89.12.134: bytes=32 time=106ms TTL=50
Reply from 143.89.12.134: bytes=32 time=106ms TTL=50
Reply from 143.89.12.134: bytes=32 time=106ms TTL=50
Reply from 143.89.12.134: bytes=32 time=107ms TTL=50
Reply from 143.89.12.134: bytes=32 time=107ms TTL=50
Reply from 143.89.12.134: bytes=32 time=107ms TTL=50
Reply from 143.89.12.134: bytes=32 time=105ms TTL=50
Reply from 143.89.12.134: bytes=32 time=105ms TTL=50
Reply from 143.89.12.134: bytes=32 time=105ms TTL=50

Ping statistics for 143.89.12.134:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 105ms, Maximum = 107ms, Average = 105ms

C:\WINDOWS\system32>
```

Ques1.) IP Source Address : 10.43.1.141 IP Destination Address : 143.89.12.134

Ques2.) Because it was designed to communicate network layer information between hosts and routers, not between application layer processes.

### Ques3.)

Wireshark - Packet 468 - Ethernet

> Frame 468: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF\_{8A2F7FC0-61D5-41F2-BDA8-8F205B6DF5B8}, id 0

> Ethernet II, Src: LCFCHFe\_a5:70:5a (f8:75:a4:a5:70:5a), Dst: Cisco\_57:01:7f (78:da:6e:57:01:7f)

> Internet Protocol Version 4, Src: 10.43.1.141, Dst: 143.89.12.134

Internet Control Message Protocol

- Type: 8 (Echo (ping) request)
- Code: 0
- Checksum: 0x4c58 [correct]
- [Checksum Status: Good]
- Identifier (BE): 256 (0x0100)
- Identifier (LE): 1 (0x0001)
- Sequence Number (BE): 4 (0x0004)
- Sequence Number (LE): 1024 (0x0400)
- [Response frame: 471]
- Data (32 bytes)

0000	78 6a 57 01 7f f8 75 a4 a5 70 5a 08 00 45 00	x.nW...u :.pZ..E.
0010	00 3c 6e 0a 00 00 7e 01 27 20 0a 2b 01 8d 8f 59	:<n...~. : .+...Y
0020	0c 86 08 00 4c 58 01 00 00 04 61 62 63 64 65 66	....LX... .abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmn opqrstuv
0040	77 61 62 63 64 65 66 67 68 69	wabcfedg hi

No.: 468 · Time: 10.70525 · Source: 10.43.1.141 · Destination: 143.89.12.134 · Protocol: ICMP · Length: 74 · Info: Echo (ping) request id=0x0100, seq=4/1024, ttl=128 (reply in 471)

Show packet bytes

Close Help

### Ques4.)

Wireshark - Packet 471 - Ethernet

> Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 143.89.12.134, Dst: 10.43.1.141

Internet Control Message Protocol

- Type: 0 (Echo (ping) reply)
- Code: 0
- Checksum: 0x5458 [correct]
- [Checksum Status: Good]
- Identifier (BE): 256 (0x0100)
- Identifier (LE): 1 (0x0001)
- Sequence Number (BE): 4 (0x0004)
- Sequence Number (LE): 1024 (0x0400)
- [Request frame: 468]
- [Response time: 105.168 ms]
- Data (32 bytes)

0000	f8 75 a4 a5 70 5a 78 da 6e 57 01 7f 08 00 45 00	.u..pZx -nW .. E.
0010	00 3c ae 13 00 00 32 01 33 17 8f 59 0c 86 0a 2b	:<...2. 3...Y ..+
0020	01 8d 00 00 54 58 01 00 00 04 61 62 63 64 65 66	....TX... .abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmn opqrstuv
0040	77 61 62 63 64 65 66 67 68 69	wabcfedg hi

No.: 471 · Time: 10.855693 · Source: 143.89.12.134 · Destination: 10.43.1.141 · Protocol: ICMP · Length: 74 · Info: Echo (ping) reply id=0x0100, seq=4/1024, ttl=50 (request in 468)

Show packet bytes

Close Help

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>tracert www.inria.fr

Tracing route to inria.fr [128.93.162.83]
over a maximum of 30 hops:

 1  601 ms      2 ms      6 ms  10.43.0.1
 2  1 ms       1 ms      1 ms  172.16.1.129
 3  *          *          * Request timed out.
 4  *          *          * Request timed out.
 5  *          *          * Request timed out.
 6  *          *          * Request timed out.
 7  *          *          * Request timed out.
 8  *          *          * Request timed out.
 9  *          *          * Request timed out.
10  *          *          * Request timed out.
11  *          *          * Request timed out.
12  *          *          * Request timed out.
13  *          *          * Request timed out.
14  *          *          * Request timed out.
15  *          *          * Request timed out.
16  *          *          * Request timed out.
17  *      236 ms     234 ms  prod-inriafr-cms.inria.fr [128.93.162.83]

Trace complete.

C:\WINDOWS\system32>

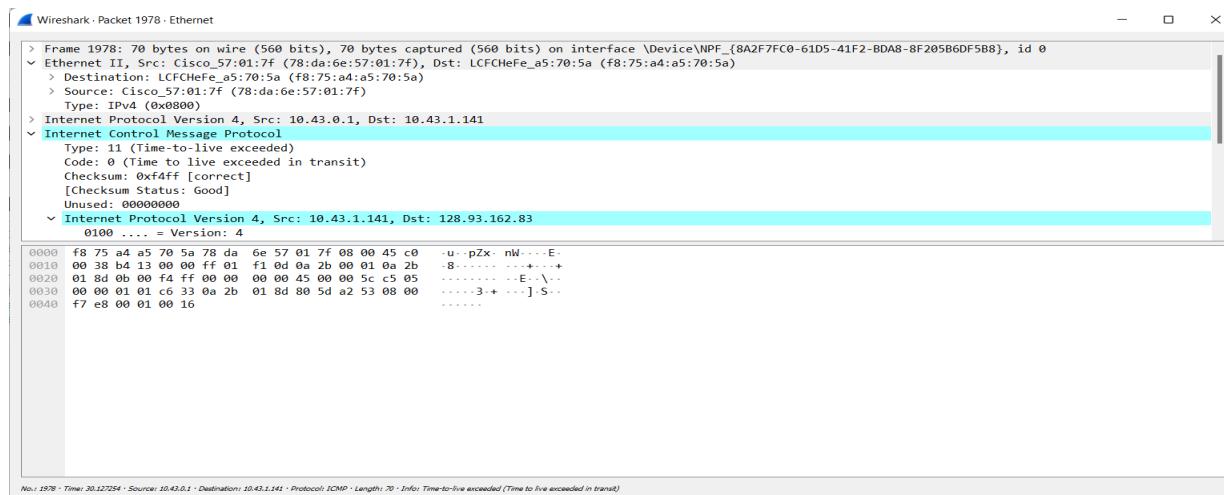
```

Ques5.) Source IP address : 192.168.137.192 Destination IP address : 128.93.162.83

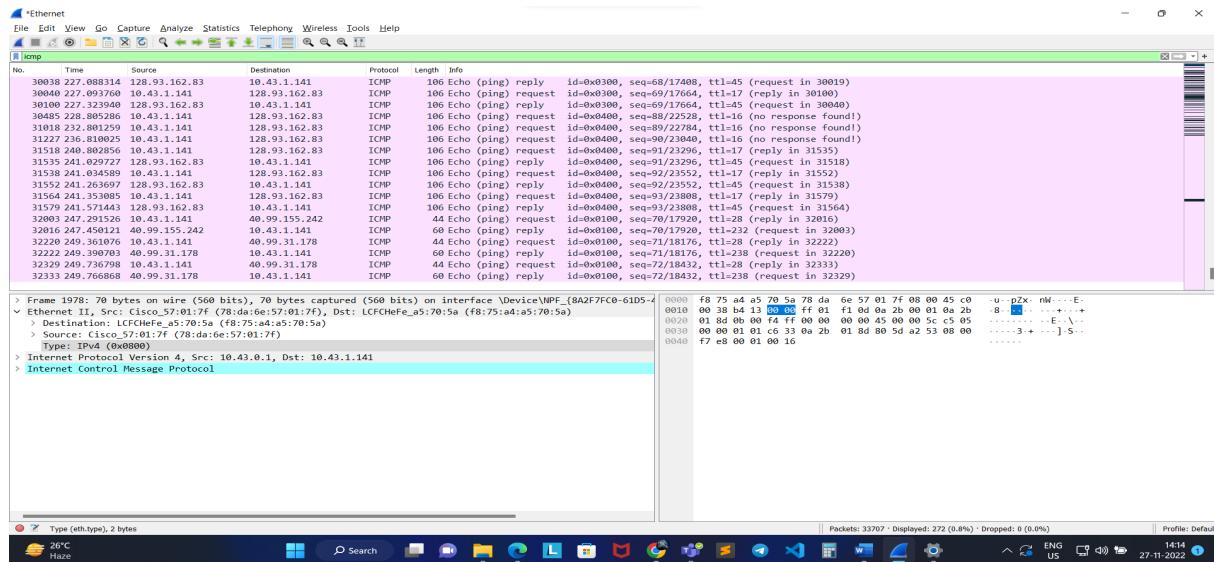
Ques6.) No, the protocol number would be 0x11.

Ques7.) The ICMP echo packet has the same fields as the ping query packets.

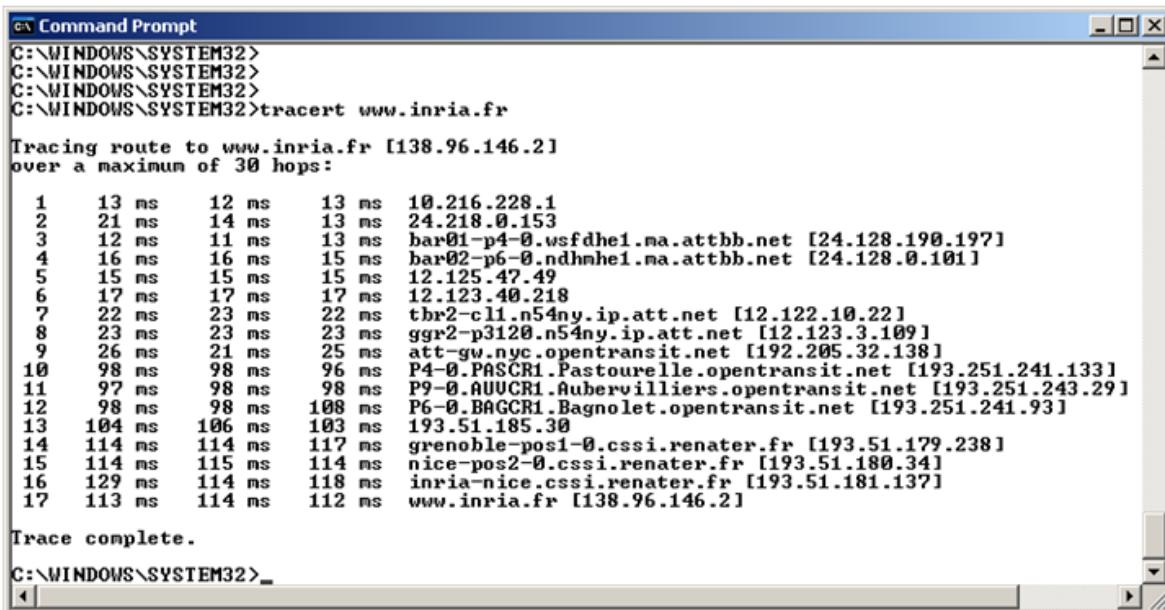
Ques8.) It contains both the IP header and the first 8 bytes of the original ICMP packet where the error is for.



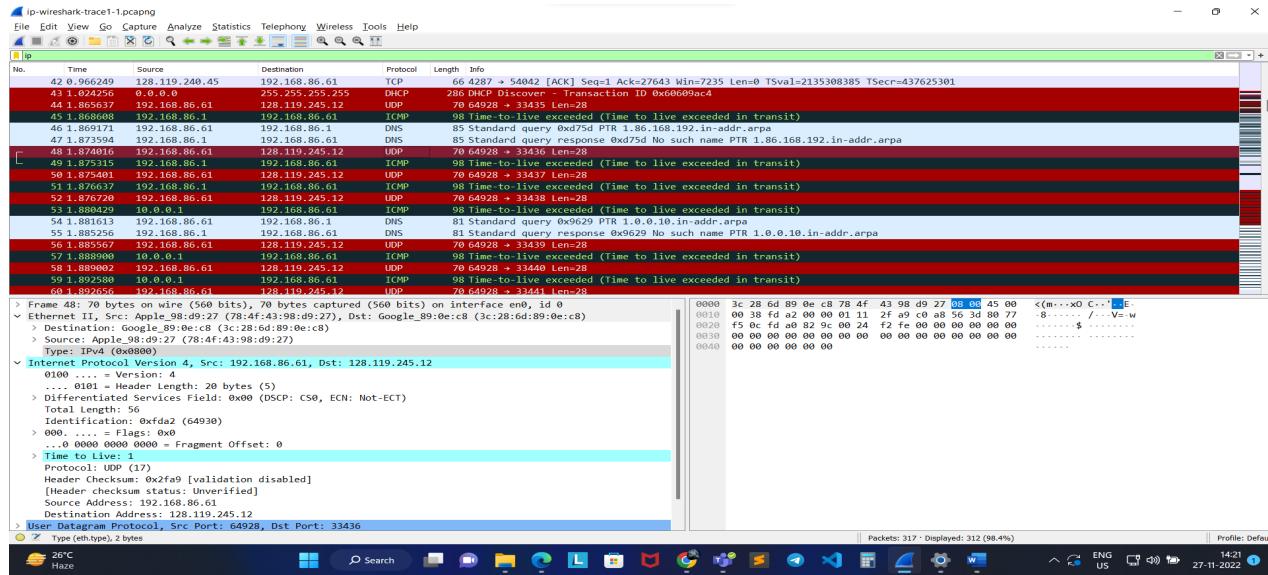
Ques9.) These last three are TYPE 0 instead of TTL expired. They are different because the datagrams have made it all the way to the destination host before the TTL expired.



Ques10.) Here we can see that from 9th message to 10th message there is a lot of time delay because we believe one router is located in New York City and the next hop router is located in France. Simply due to distance and speed of light it takes longer for the packet to respond.



## LAB : IP



Ques1.) Source IP address : 192.168.86.61 .

Ques2.) Time to Live field : 1

Ques3.) Upper layer Protocol : UDP (17)

Ques4.) Header Length = 20 bytes

Ques5.) Payload = Total length - Header = 56-20 = 36 bytes

Ques6.) No , fragmentation is not done as More Fragments bit is not set.

**0000 00.. = Differentiated Services Codepoint: De**  
**.... ..00 = Explicit Congestion Notification: Not**

Total Length: 56

Identification: 0xfdःa1 (64929)

✗ 000. .... = Flags: 0x0

0.... .... = Reserved bit: Not set

.0.. .... = Don't fragment: Not set

..0. .... = More fragments: Not set

....0 0000 0000 0000 = Fragment Offset: 0

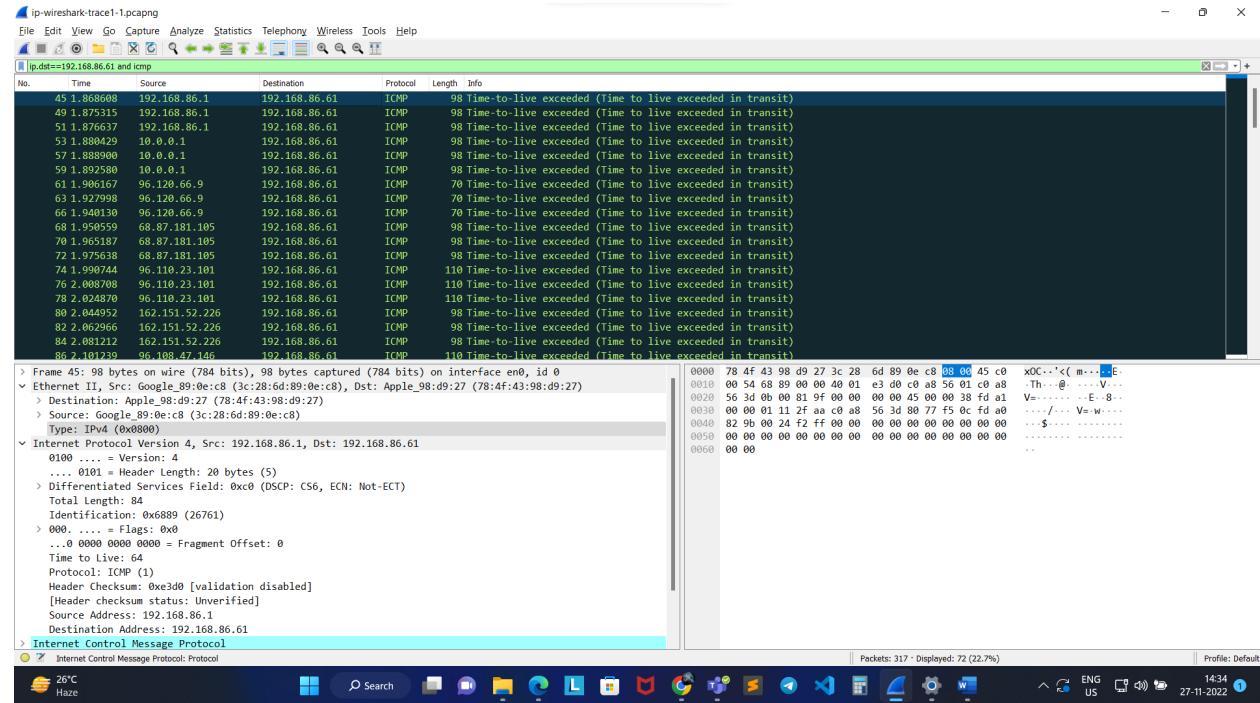
➢ Time to Live: 1

Protocol: UDP (17)

Ques7.) Checksum field always changes as the value of Time to live field changes on every hop.

Ques8.) Identification number, Source IP address , Destination IP address etc. remains constant .

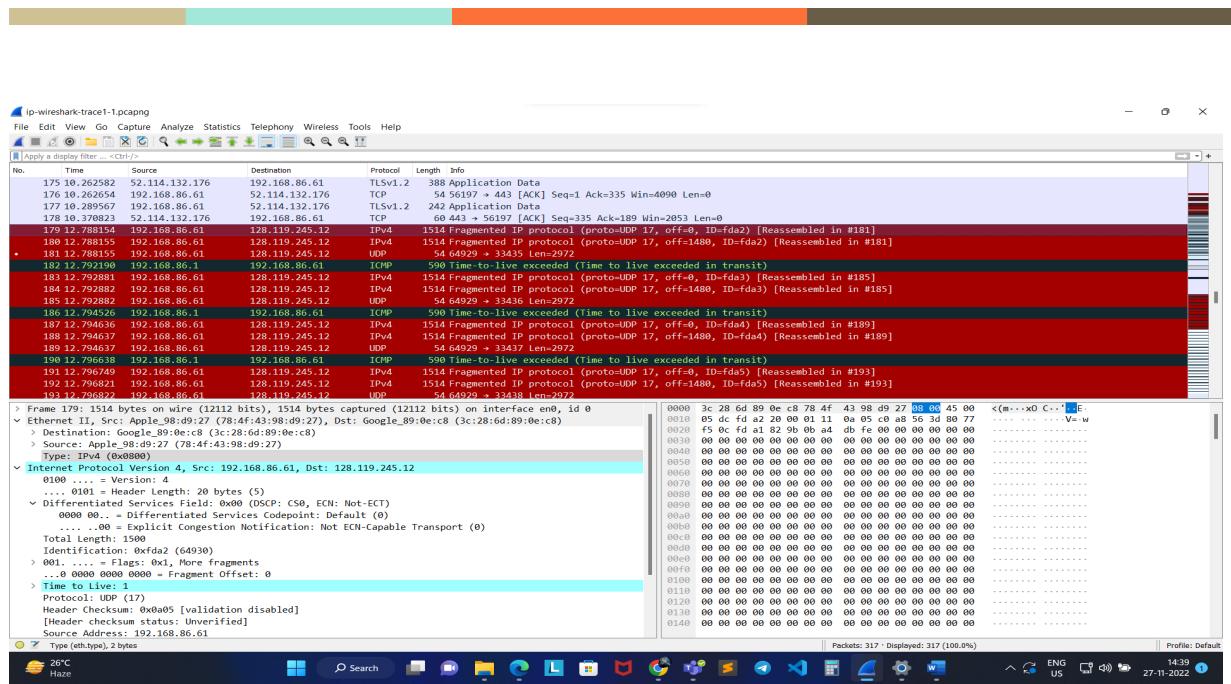
Ques9.) The pattern is that identification number increments after each ICMP ECHO request.



Ques10.) Upper layer protocol : ICMP(1)

Ques11.) Yes , the identification number in ICMP ECHO requests increments by one after each request..

Ques12.) The TTL field remains unchanged because the TTL for the first hop router is always the same.



Ques13.) Yes, there are 3 fragments.

Ques14.) The More Fragments bit is set to 1 in the IP header.

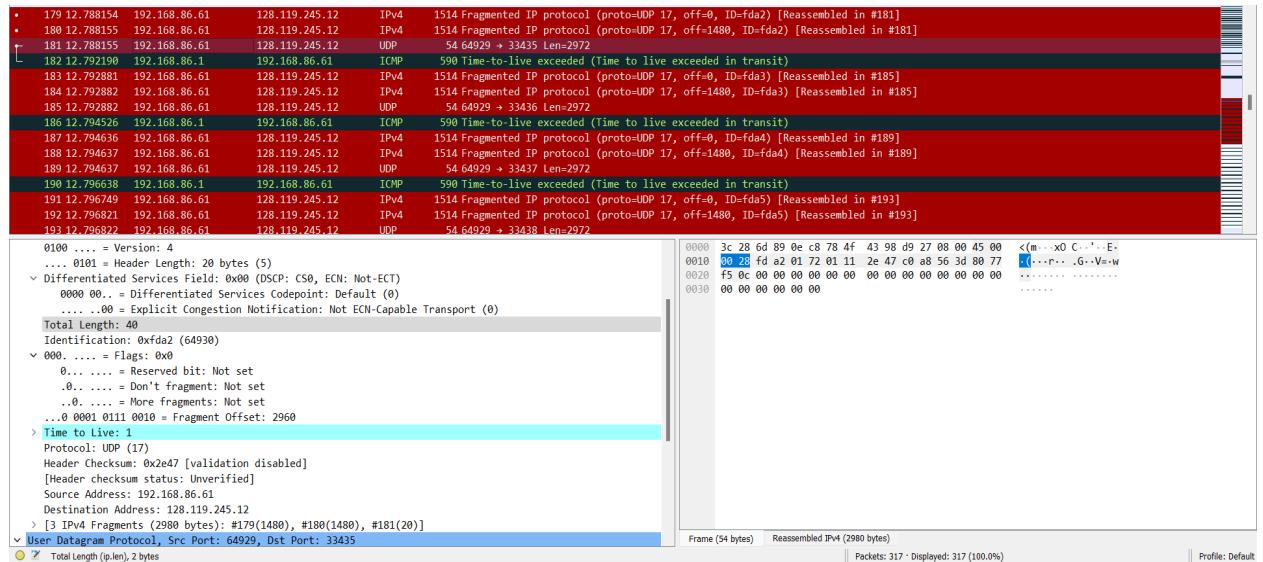
Ques15.) Since the Fragment Offset value is 0, this indicates that it is the first fragment.

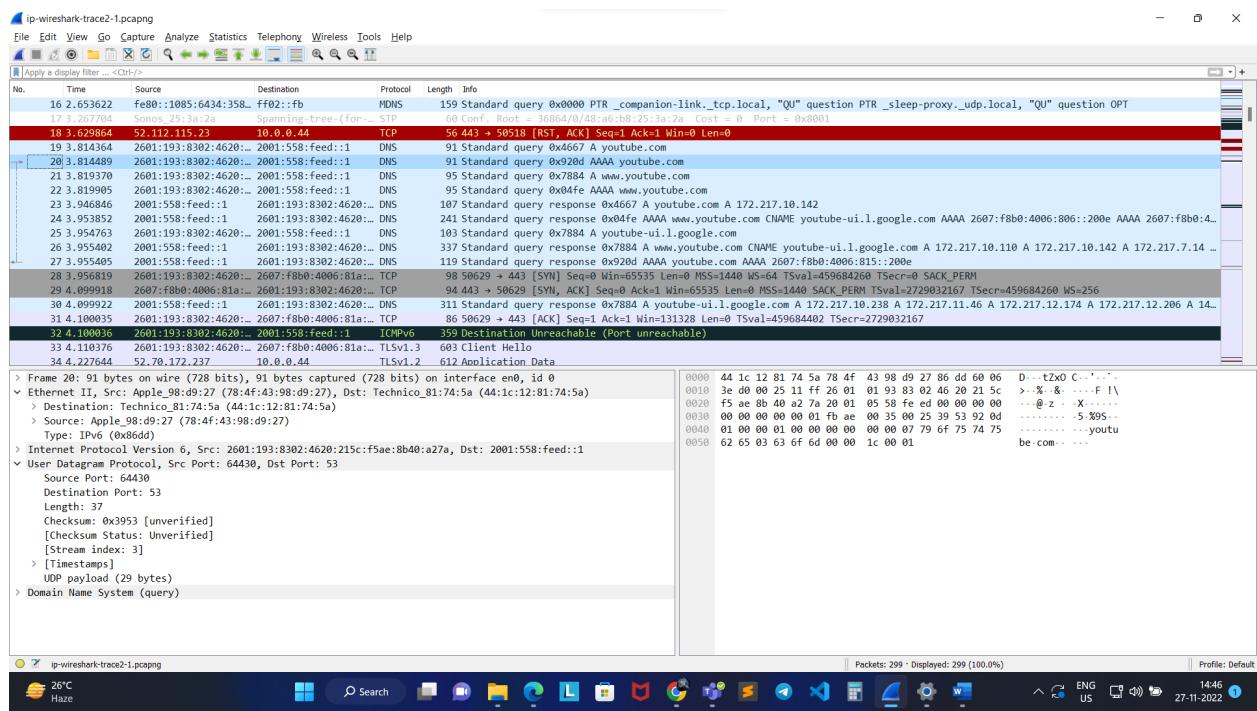
Ques16.) Total length = 1500 bytes.

Ques17.) Since the fragment offset is not equal to 0, this is not the first fragment.

Ques18.) The fields which change are fragment offset, time to live field, checksum etc.

Ques19.) Since the More Fragments bit is set to 0, this indicates that the fragment is the last fragment.





Ques20.) IPv6 Source address : **2601:193:8302:4620:215c:f5ae:8b40:a27a**

Ques21.) IPv6 Destination address : **2001:558:feed::1**

Ques22.) Flow label : **Flow Label: 0x63ed0**

Ques23.) Payload length = 37

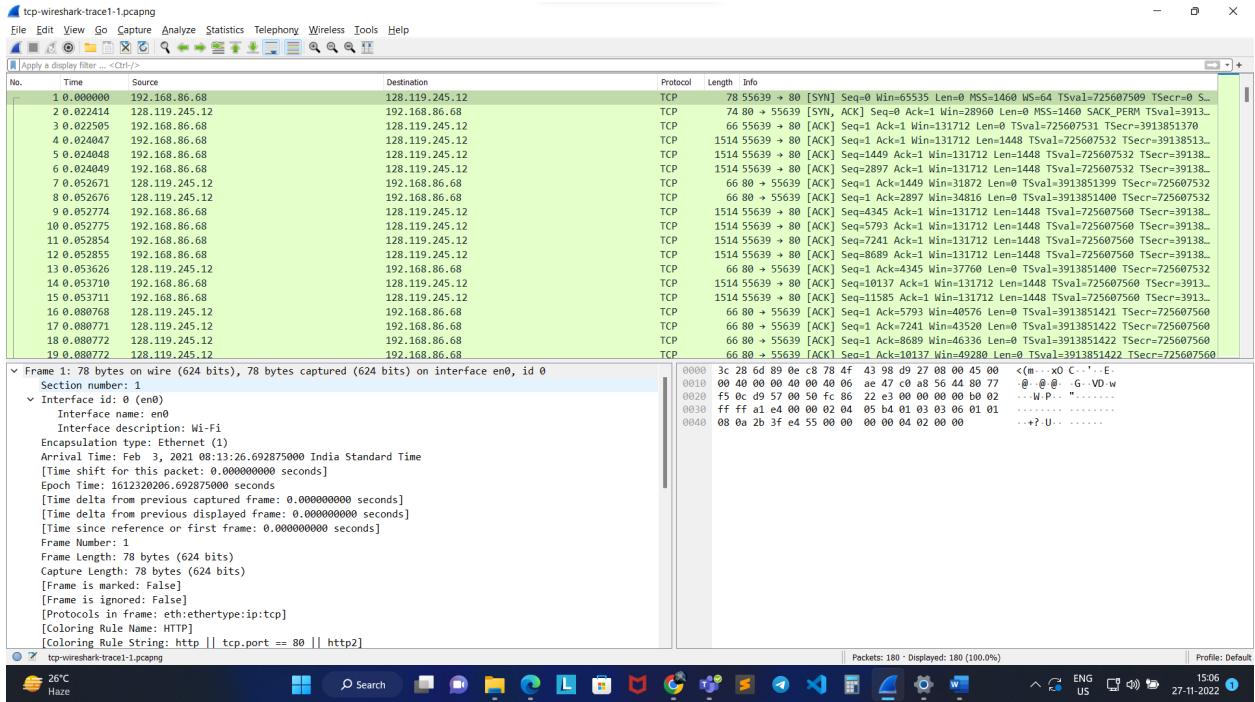
Ques24.) Upper layer protocol : UDP(17)

Ques25.) 1 IPv6 address is returned.

**addr 2607:f8b0:4006:815::200e**

Ques26.) IPv6 address returned :

## LAB : TCP



Ques1.) IP source address : 192.168.86.68    TCP source port number : 55639

Ques2.) IP address : 128.119.245.12

Source port number : 80   Destination port number : 55639

Ques3.) Sequence number of SYN segment (RAW) : 4236649187 . It is identified as a SYN segment as the SYN flag is set .

Ques4.) Sequence number of SYNACK segment (RAW) : 1068969752 , acknowledgement number of the SYNACK segment (RAW) : 4236649188 . It is identified as SYNACK as SYN and ACK flags are set in this segment.

Ques5.) Sequence number (RAW) : 4236649188 , TCP payload data = 1448 bytes . No , the file is transferred in multiple segments.

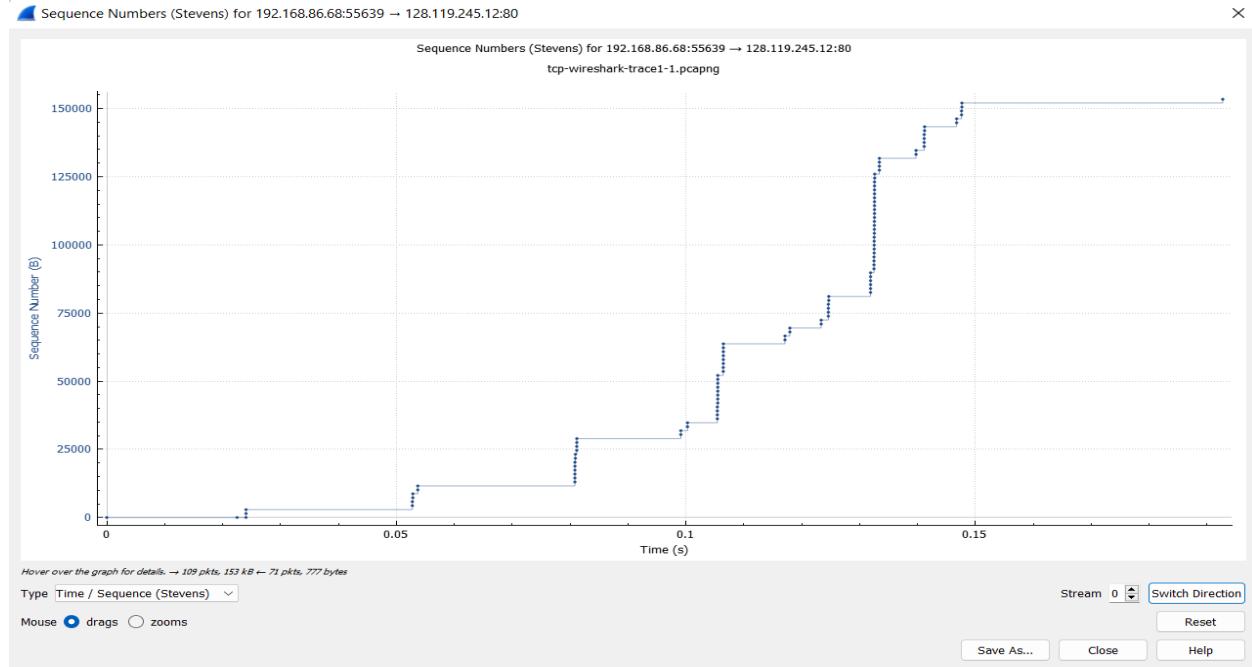
Ques6.)

- 1.) The time at which the first packet was sent is : 0.024047s
- 2.) The time at which the ack for first packet came : 0.052671s
- 3.) RTT for the first packet is : 0.028624s
- 4.) RTT for the second packet is : 0.028628s
- 5.) Estimated RTT =  $(1-0.125)*0.028624 + (0.125)*0.028628 = 0.0286245$

Ques7.) The length of (header+payload) of the first 4 segments : 1480 bytes

Ques8.) The minimum amount of buffer space advertised : 31872. No, the lack of receiver buffer space never throttles the sender for these first four data carrying segments.

Ques9.) There are no retransmitted segments in the trace file. We can verify this by checking the sequence numbers of the TCP segments in the trace file.



Ques10.)

Acknowledgement Number	Packet Number	Data
4236650636	Ack1	1448
4236652084	Ack2	1448
4236653532	Ack3	1448
4236654980	Ack4	1448
4236656428	Ack5	1448
4236657876	Ack6	1448
4236659324	Ack7	1448
4236660772	Ack8	1448
4236662220	Ack9	1448
4236663668	Ack10	1448

The difference between consecutive acknowledgement numbers indicates the size of the packet.

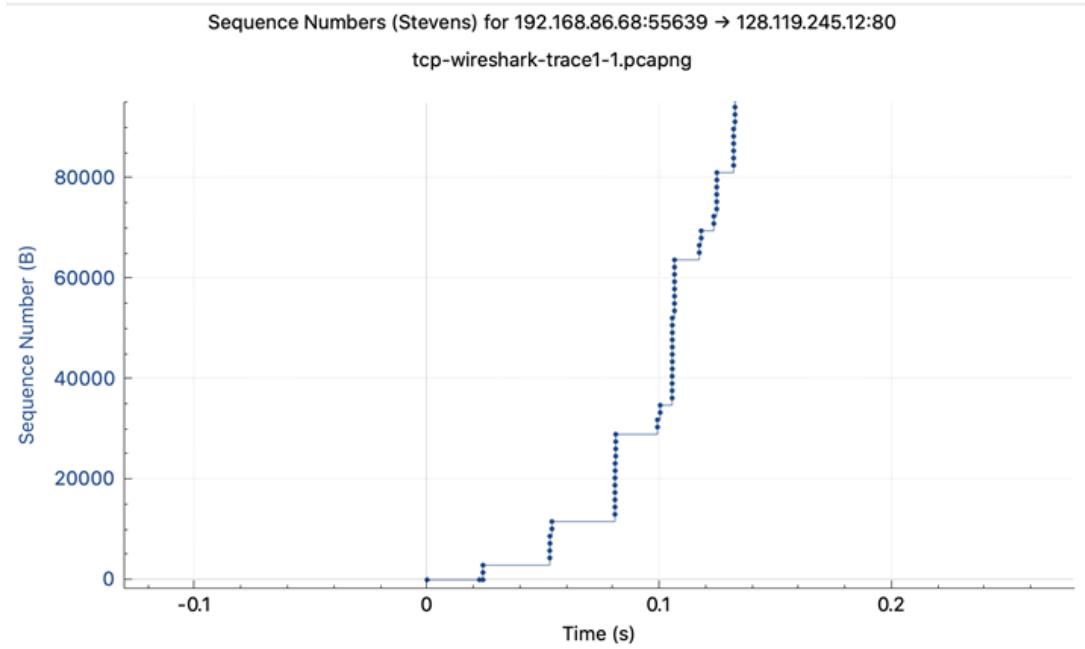
Ques11.) Throughput = Total data sent/ Total time

Total data sent = 150241 bytes

Total time = 0.191496

Throughput = 7,84,564.69 bytes/s

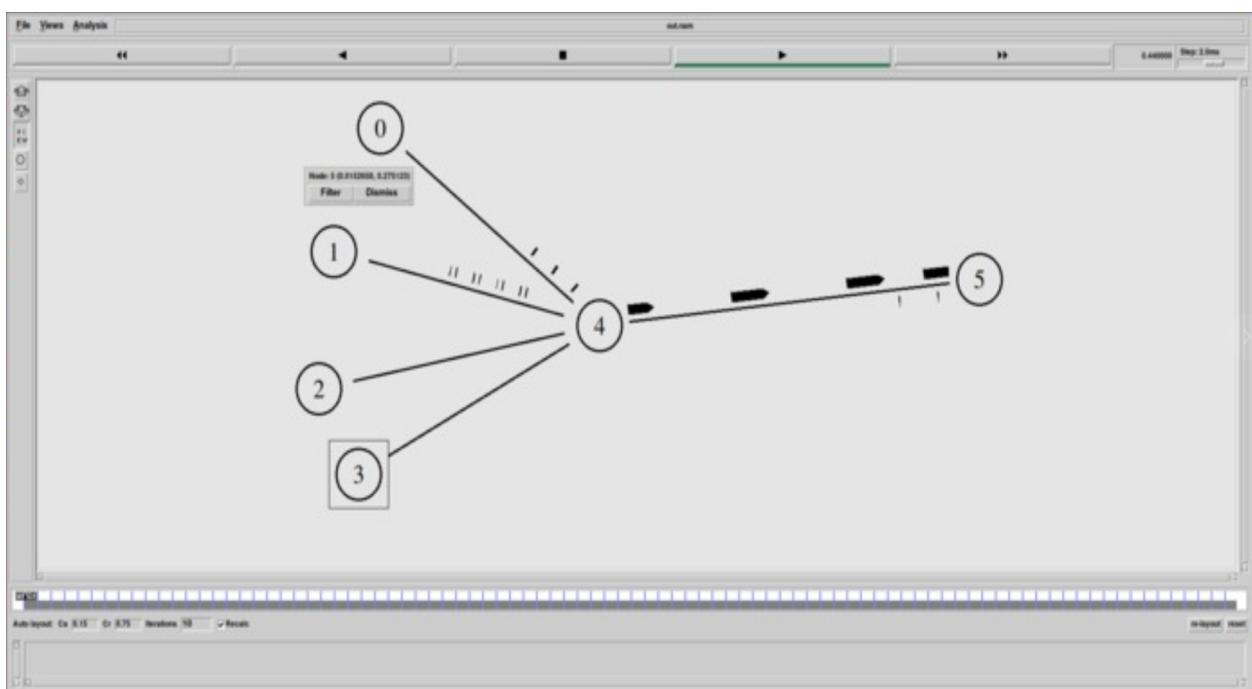
Ques12.)



# Task 2 : NS2

Github Link : <https://github.com/akshatagarwal18/CSN-341>

**Ques1.)**



Simulator for Task 1

### 1.) P = 0.1%

The screenshot shows a Linux desktop environment with a dark theme. On the left, there is a vertical dock containing icons for various applications like a browser, file manager, and system tools. Two windows are open:

- Terminal Window:** The title bar says "Activities Terminal". The command `ns almostcomplete.tcl` is being run, and the output shows configuration details for TCP sources and file sizes across three classes (0, 1, 2, 3). The terminal window has a dark background with light-colored text.
- Visual Studio Code Window:** The title bar says "Activities Visual Studio Code". It shows a Python script named "script.py" with some code. Below the editor, the terminal tab is active, displaying the command `/usr/bin/python3 /home/achintya/Desktop/folder/script.py` followed by several numerical values. The VS Code interface includes a sidebar with file navigation, a bottom status bar with Python-related information, and a toolbar with various icons.

- 
- Average Time Delay in every class :
    - 1.) 0.006468662564762488
    - 2.) 0.006636719583034258
    - 3.) 0.007961131620856099
    - 4.) 0.00653515642863158
  - Average Bandwidth in every class :
    - 1.) 0.7504982600956Mbps
    - 2.) 0.5725840835153Mbps
    - 3.) 0.4915482102755Mbps
    - 4.) 0.5327982639780Mbps
  - Wi/W1 for each class :
    - 1.) 1
    - 2.) 1.0259801800742006403498005839312
    - 3.) 1.2307229726626512438407044381402
    - 4.) 1.0102793835979808102661672255906
  - Confidence Interval : A confidence interval (CI) is a range of estimates for an unknown parameter. A confidence interval is computed at a designated *confidence level*; the 95% confidence level is most common, but other levels, such as 90% or 99%, are sometimes used. The confidence level represents the long-run proportion of corresponding CIs that contain the true value of the parameter.

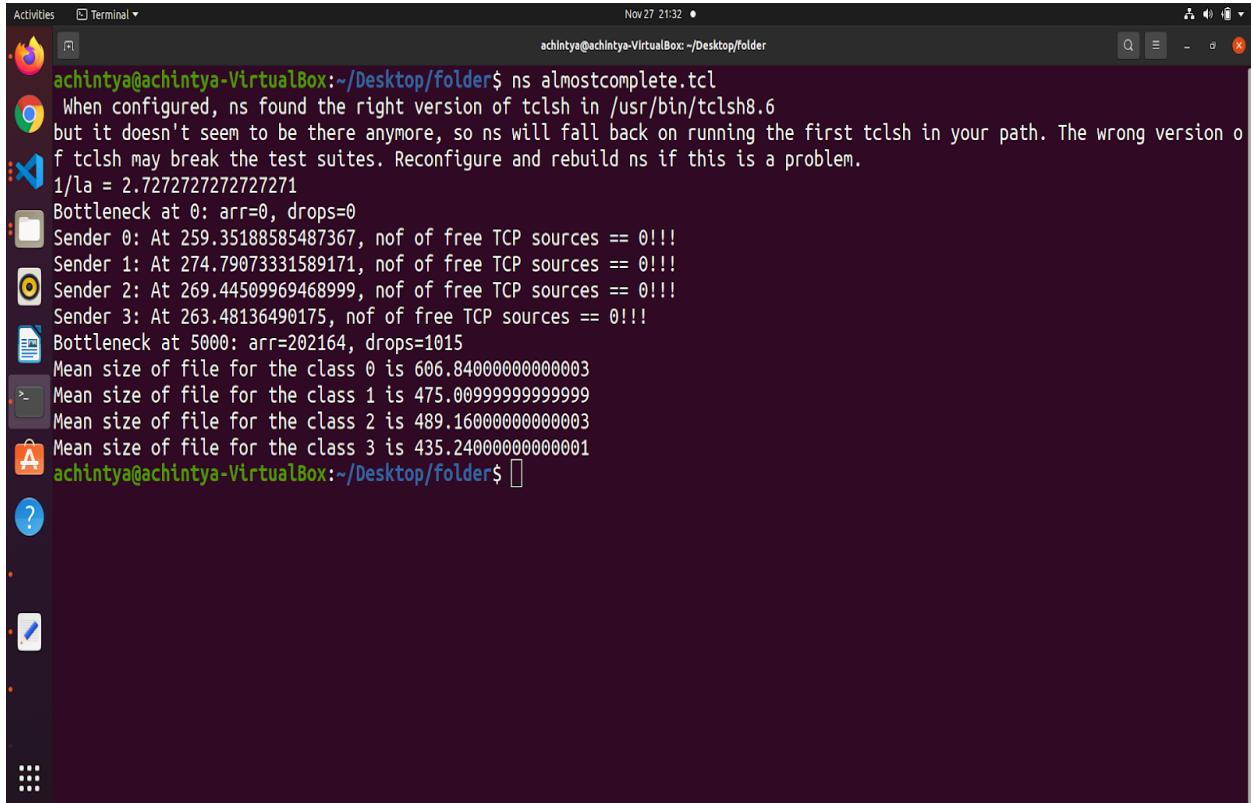
## 2.) P = 0.5

The screenshot shows a Linux desktop environment with two windows open:

- Terminal Window:** The terminal window is titled "Terminal" and shows the command "ns almostcomplete.tcl" being run. The output indicates that ns found the right version of tclsh in /usr/bin/tclsh8.6, but it doesn't seem to be there anymore, so ns will fall back on running the first tclsh in your path. The wrong version of tclsh may break the test suites. Reconfigure and rebuild ns if this is a problem. The terminal also displays various TCP source statistics and mean sizes for different classes.
- Visual Studio Code Window:** The VS Code window is titled "script.py - folder - Visual Studio Code". It shows the Python script "script.py" with code that prints the absolute differences between elements in four lists (a1, a2, a3, a4) and then runs it. The terminal tab in VS Code shows the command "/usr/bin/python3 /home/achintya/Desktop/folder/script.py" and its output, which includes several floating-point numbers.

- 
- Average Time Delay in every class :
    - 1.) 0.01022969990144449
    - 2.) 0.011194457725727289
    - 3.) 0.0078335763728402
    - 4.) 0.00436541910468866
  - Average Bandwidth in every class :
    - 1.) 0.47457110636397914126183660287422 Mbps
    - 2.) 0.33946083795256940743204367661661Mbps
    - 3.) 0.4995521603092729144210287527281Mbps
    - 4.) 0.79761413887162828687814906853867Mbps
  - Wi/W1 for each class :
    - 1.) 1
    - 2.) 1.0943094942742719049476472888537
    - 3.) 0.76576795490687423262695802441992
    - 4.) 0.42673970368106680875650475390967
  - Confidence Interval : A confidence interval (CI) is a range of estimates for an unknown parameter. A confidence interval is computed at a designated *confidence level*; the 95% confidence level is most common, but other levels, such as 90% or 99%, are sometimes used. The confidence level represents the long-run proportion of corresponding CIs that contain the true value of the parameter.

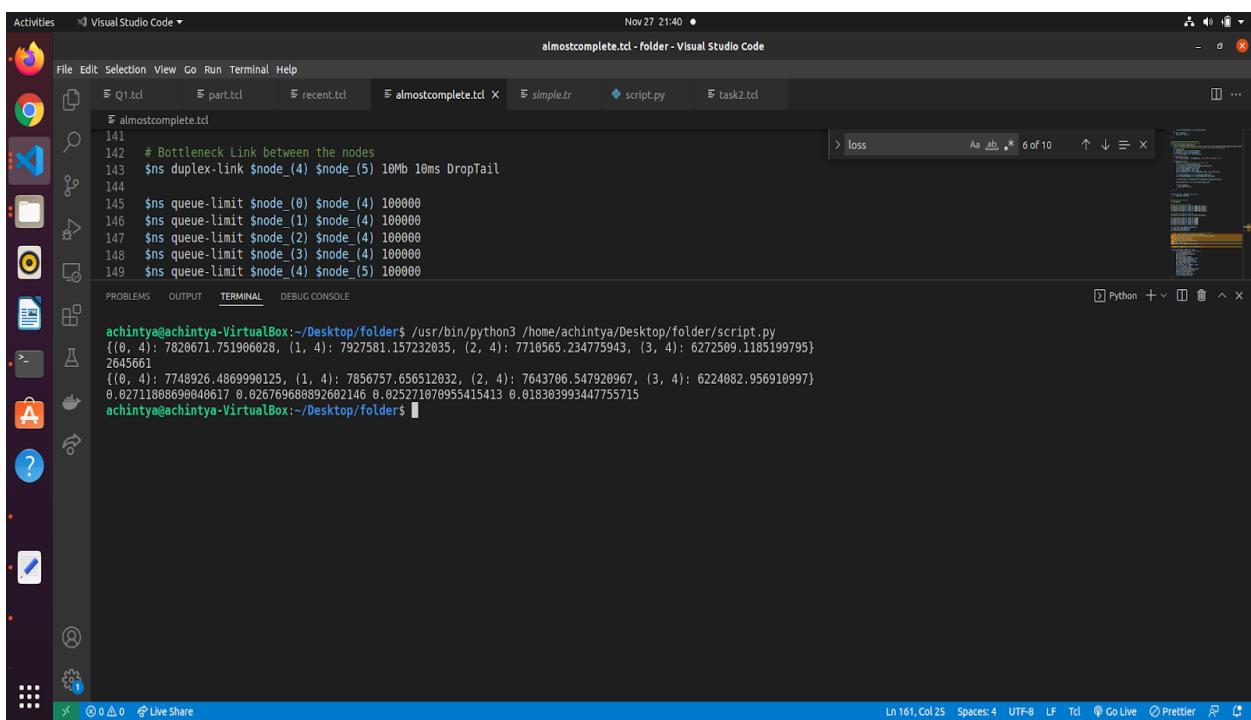
### 3.) P = 1%



```

achintya@achintya-VirtualBox:~/Desktop/folder$ ns almostcomplete.tcl
When configured, ns found the right version of tclsh in /usr/bin/tclsh8.6
but it doesn't seem to be there anymore, so ns will fall back on running the first tclsh in your path. The wrong version o
f tclsh may break the test suites. Reconfigure and rebuild ns if this is a problem.
1/la = 2.7272727272727271
Bottleneck at 0: arr=0, drops=0
Sender 0: At 259.35188585487367, nof of free TCP sources == 0!!!
Sender 1: At 274.79073331589171, nof of free TCP sources == 0!!!
Sender 2: At 269.44509969468999, nof of free TCP sources == 0!!!
Sender 3: At 263.48136490175, nof of free TCP sources == 0!!!
Bottleneck at 5000: arr=202164, drops=1015
Mean size of file for the class 0 is 606.8400000000000
Mean size of file for the class 1 is 475.0099999999999
Mean size of file for the class 2 is 489.1600000000000
Mean size of file for the class 3 is 435.2400000000000
achintya@achintya-VirtualBox:~/Desktop/folder$ 

```



```

File Ed Selection View Go Run Terminal Help
File Q1.tcl part.tcl recent.tcl almostcomplete.tcl simple.tr script.py task2.tcl
almostcomplete.tcl - folder - Visual Studio Code
141
142 # Bottleneck Link between the nodes
143 $ns duplex-link $node_(4) $node_(5) 10Mb 10ms DropTail
144
145 $ns queue-limit $node_(0) $node_(4) 100000
146 $ns queue-limit $node_(1) $node_(4) 100000
147 $ns queue-limit $node_(2) $node_(4) 100000
148 $ns queue-limit $node_(3) $node_(4) 100000
149 $ns queue-limit $node_(4) $node_(5) 100000

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
achintya@achintya-VirtualBox:~/Desktop/folder$ /usr/bin/python3 /home/achintya/Desktop/folder/script.py
{(), 4}: 7826671.751966628, (1, 4): 7927581.157232035, (2, 4): 7710565.234775943, (3, 4): 6272589.1185199795
2645661
{(), 4}: 7748926.4869990125, (1, 4): 7856757.656512032, (2, 4): 7643706.547928967, (3, 4): 6224082.956910997
0.027118869840617 0.02676968892602146 0.025271870955415413 0.018303993447755715
achintya@achintya-VirtualBox:~/Desktop/folder$ 

```

- 
- Average Time Delay in every class :
    - 1.) 0.02711808690040617
    - 2.) 0.026769680892602146
    - 3.) 0.025271070955415413
    - 4.) 0.018303993447755715
  - Average Bandwidth in every class :
    - 1.) 0.17902147809428581948 Mbps
    - 2.) 0.14195462453383815968Mbps
    - 3.) 0.15485216304857122895Mbps
    - 4.) 0.19022734082255282091Mbps
  - Wi/W1 for each class :
    - 1.) 1
    - 2.) 0.98715226449846631197013598839331
    - 3.) 0.93188988766891617580610069941449
    - 4.) 0.67497362608870321756700143232157
  - Confidence Interval : A confidence interval (CI) is a range of estimates for an unknown parameter. A confidence interval is computed at a designated *confidence level*; the 95% confidence level is most common, but other levels, such as 90% or 99%, are sometimes used. The confidence level represents the long-run proportion of corresponding CIs that contain the true value of the parameter.

#### 4.) P = 5

Nov 27 21:58 • achintya@achintya-VirtualBox:~/Desktop/folder\$ ns almostcomplete.tcl

```
achintya@achintya-VirtualBox:~/Desktop/folder$ ns almostcomplete.tcl
When configured, ns found the right version of tclsh in /usr/bin/tclsh8.6
but it doesn't seem to be there anymore, so ns will fall back on running the first tclsh in your path. The wrong version of tclsh may break the test suites. Reconfigure and rebuild ns if this is a problem.
n,
1/la = 2.7272727272727271
Bottleneck at 0: arr=0, drops=0!!!
Sender 0: AT 259.35188585487367, nof of free TCP sources == 0!!!
Sender 1: AT 274.7907331589171, nof of free TCP sources == 0!!!
Sender 2: AT 269.44589969468999, nof of free TCP sources == 0!!!
Sender 3: AT 263.40136490175, nof of free TCP sources == 0!!!
Bottleneck at 5000: arr=202873, drops=2078
Mean size of file for the class 0 is 606.8400000000000003
Mean size of file for the class 1 is 475.0099999999999999
Mean size of file for the class 2 is 489.1600000000000003
Mean size of file for the class 3 is 435.2400000000000001
achintya@achintya-VirtualBox:~/Desktop/folder$
```

Nov 27 21:59 • script.py - folder - Visual Studio Code

File Edit Selection View Go Run Terminal Help

script.py > ...

```
76 print(map)
77 print(count)
78 print(map2)
79 a1 = (map2[0,4] - map[0,4])/count
80 a2 = (map2[1,4] - map[1,4])/count
81 a3 = (map2[2,4] - map[2,4])/count
82 a4 = (map2[3,4] - map[3,4])/count
83 print(abs(a1),abs(a2),abs(a3),abs(a4))
84
85
86
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

achintya@achintya-VirtualBox:~/Desktop/folder\$ /usr/bin/python3 /home/achintya/Desktop/folder/script.py

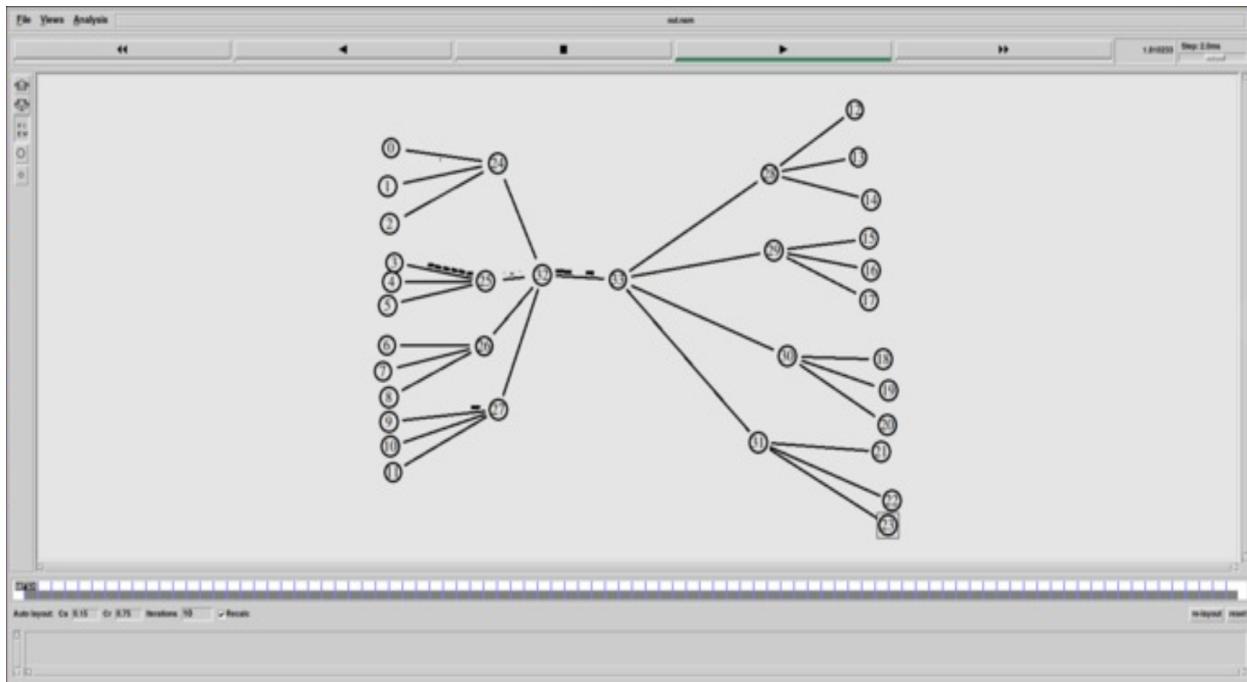
```
{(0, 4): 7820671.751906028, (1, 4): 7927581.157232035, (2, 4): 7710565.234775943, (3, 4): 6272509.1185199795}
2645661
{(0, 4): 7748926.4869990125, (1, 4): 7856757.656512032, (2, 4): 7643706.547920967, (3, 4): 6224082.956910997}
0.02711888690040617 0.026769688892602146 0.025271070955415413 0.018303993447755715
achintya@achintya-VirtualBox:~/Desktop/folder$
```

OUTLINE

Ln 86, Col 1 Spaces: 2 UTF-8 LF Python 3.8.10 64-bit Go Live Prettier

- 
- Average Time Delay in every class :
    - 1.) 0.02711808690040617
    - 2.) 0.026769680892602146
    - 3.) 0.025271070955415413
    - 4.) 0.018303993447755715
  - Average Bandwidth in every class :
    - 1.) 0.17902147809428581948 Mbps
    - 2.) 0.14195462453383815968Mbps
    - 3.) 0.15485216304857122895Mbps
    - 4.) 0.19022734082255282091Mbps
  - Wi/W1 for each class :
    - 1.) 1
    - 2.) 0.98715226449846631197013598839331
    - 3.) 0.93188988766891617580610069941449
    - 4.) 0.67497362608870321756700143232157
  - Confidence Interval : A confidence interval (CI) is a range of estimates for an unknown parameter. A confidence interval is computed at a designated *confidence level*; the 95% confidence level is most common, but other levels, such as 90% or 99%, are sometimes used. The confidence level represents the long-run proportion of corresponding CIs that contain the true value of the parameter.

## Ques2.)



Simulator for Task 2

# Task 3 : Networking Tools

**ping** : The ping command is used to check if a remote system is running or up. In short this command is used to detect whether a system is connected to the network or not.

```
 akshat_a18@LAPTOP-R86737RN:~$ ping www.google.com
PING www.google.com (142.250.192.196) 56(84) bytes of data.
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=1 ttl=116 time=7.55 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=2 ttl=116 time=8.66 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=3 ttl=116 time=8.08 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=4 ttl=116 time=8.84 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=5 ttl=116 time=8.49 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=6 ttl=116 time=7.95 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=7 ttl=116 time=8.43 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=8 ttl=116 time=8.09 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=9 ttl=116 time=8.72 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=10 ttl=116 time=7.80 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=11 ttl=116 time=7.79 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=12 ttl=116 time=8.47 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=13 ttl=116 time=8.42 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=14 ttl=116 time=9.42 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=15 ttl=116 time=8.36 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=16 ttl=116 time=8.44 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=17 ttl=116 time=7.98 ms
54 bytes from del11s12-in-f4.1e100.net (142.250.192.196): icmp_seq=18 ttl=116 time=7.48 ms
^C
--- www.google.com ping statistics ---
18 packets transmitted, 18 received, 0% packet loss, time 17038ms
rtt min/avg/max/mdev = 7.484/8.275/9.421/0.472 ms
akshat_a18@LAPTOP-R86737RN:~$ -
```

**host** : This command is used to obtain network address information about a remote system connected to your network. This information usually consists of system's IP address, domain name address and sometimes mail server also.

```
 akshat_a18@LAPTOP-R86737RN:~$ host www.facebook.com
www.facebook.com is an alias for star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com has address 157.240.16.35
star-mini.c10r.facebook.com has IPv6 address 2a03:2880:f12f:83:face:b00c:0:25de
akshat_a18@LAPTOP-R86737RN:~$
```

**finger** : One can obtain information about the user on its network and the who command to see what users are currently online on your system. The who command lists all users currently connected, along with when, how long, and where they logged in. finger can operate on large networks, though most systems block it for security reasons.

```
 akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ sudo apt install finger
[sudo] password for akshat_a18:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  finger
0 upgraded, 1 newly installed, 0 to remove and 25 not upgraded.
Need to get 16.9 kB of archives.
After this operation, 51.2 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 finger amd64 0.17-17 [16.9 kB]
Fetched 16.9 kB in 1s (21.3 kB/s)
Selecting previously unselected package finger.
(Reading database ... 37628 files and directories currently installed.)
Preparing to unpack .../finger_0.17-17_amd64.deb ...
Unpacking finger (0.17-17) ...
Setting up finger (0.17-17) ...
Processing triggers for man-db (2.9.1-1) ...
akshat_a18@LAPTOP-R86737RN:~$ finger www.google.com
finger: www.google.com: no such user.
akshat_a18@LAPTOP-R86737RN:~$ finger akshat_a18
Login: akshat_a18          Name:
Directory: /home/akshat_a18      Shell: /bin/bash
Never logged in.
No mail.
No Plan.
akshat_a18@LAPTOP-R86737RN:~$ -
```

**traceroute** : This command is used to track the sequence of computer networks. Mtr or xmtr tools can also be used to perform both ping and traces. Options are available for specifying parameters like the type of service.

```
 akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ traceroute www.google.com
traceroute to www.google.com (142.250.192.228), 30 hops max, 60 byte packets
 1 LAPTOP-R86737RN (172.27.224.1)  0.336 ms  0.315 ms  0.329 ms
 2 10.43.0.1 (10.43.0.1)  1.532 ms  1.736 ms  2.086 ms
 3 172.16.1.129 (172.16.1.129)  0.576 ms  0.627 ms  0.648 ms
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 192.168.137.133 (192.168.137.133)  503.198 ms !H * *
akshat_a18@LAPTOP-R86737RN:~$
```

**netstat** : This command is used to check the status of ports whether they are open, closed, waiting, and masquerade connections. Network Statistic (netstat) command displays the connection information, routing table information etc.

```
 akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 172.27.224.255:59588    kazooie.canonical.:http TIME_WAIT
tcp      0      1 172.27.224.255:58508    banjo.canonical.co:http FIN_WAIT1
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type      State           I-Node Path
unix    2      [ ]        DGRAM
unix    3      [ ]        STREAM   CONNECTED      21515  /var/run/chrony/chronyd.sock
unix    3      [ ]        STREAM   CONNECTED      22556
unix    3      [ ]        STREAM   CONNECTED      17879
unix    3      [ ]        STREAM   CONNECTED      21534
unix    3      [ ]        STREAM   CONNECTED      23608
unix    3      [ ]        STREAM   CONNECTED      22555
unix    3      [ ]        STREAM   CONNECTED      17956  /mnt/wslg/PulseAudioRDPSink
unix    3      [ ]        STREAM   CONNECTED      23580
unix    3      [ ]        STREAM   CONNECTED      21535
unix    3      [ ]        STREAM   CONNECTED      17882
unix    3      [ ]        STREAM   CONNECTED      23603
unix    3      [ ]        STREAM   CONNECTED      18449
unix    3      [ ]        STREAM   CONNECTED      23581
unix    3      [ ]        STREAM   CONNECTED      17888  /tmp/.X11-unix/X0
unix    3      [ ]        STREAM   CONNECTED      18448
unix    2      [ ]        STREAM   CONNECTED      17261
unix    3      [ ]        STREAM   CONNECTED      17881
unix    3      [ ]        STREAM   CONNECTED      23600
unix    3      [ ]        STREAM   CONNECTED      17891  @/tmp/dbus-6nm5n09RHZ
unix    3      [ ]        STREAM   CONNECTED      17880
akshat_a18@LAPTOP-R86737RN:~$
```

**tracepath** : tracepath performs a very similar function to that of traceroute command. The main difference between this command is that tracepath doesn't take complicated options. This command does not require root privileges.

```
 akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ tracepath www.google.com
1?: [LOCALHOST]                                pmtu 1468
1:  LAPTOP-R86737RN                           0.465ms
1:  LAPTOP-R86737RN                           0.919ms
2:  192.168.137.133                         915.792ms !H
     Resume: pmtu 1468
akshat_a18@LAPTOP-R86737RN:~$
```

**dig** : dig(Domain Information Groper) query DNS - related information like a record, cname, mxrecord etc. this command is used to solve DNS related queries.

```
akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ dig www.google.com

; <>> DiG 9.16.1-Ubuntu <>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29286
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.google.com.           IN      A

;; ANSWER SECTION:
www.google.com.          0       IN      A      142.250.192.196

;; Query time: 9 msec
;; SERVER: 172.27.224.1#53(172.27.224.1)
;; WHEN: Sun Nov 27 16:25:03 IST 2022
;; MSG SIZE  rcvd: 62

akshat_a18@LAPTOP-R86737RN:~$
```

**hostname** : This command is used to see the hostname of your computer. You can change hostname permanently in etc/sysconfig/network. After changing the hostname you need to reboot the computer .

```
akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ hostname
LAPTOP-R86737RN
akshat_a18@LAPTOP-R86737RN:~$ -
```

**route** : The route command is used to display or modify the routing table.

```
akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0         172.27.224.1   0.0.0.0        UG     0      0        0 eth0
172.27.224.0    0.0.0.0        255.255.240.0  U      0      0        0 eth0
akshat_a18@LAPTOP-R86737RN:~$ -
```

**nslookup** : We can use the nslookup command to find out DNS related queries or testing and for troubleshooting DNS servers.

```
akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ nslookup google.com
Server:          172.27.224.1
Address:         172.27.224.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.193.238
Name:   google.com
Address: 2404:6800:4002:81d::200e

akshat_a18@LAPTOP-R86737RN:~$ -
```

**ifconfig** : Not only will it provide IPv4 information, but it will also provide IPv6 addresses, MAC addresses, DNS servers, default gateways and data with regard to how much traffic is flowing over the interface along with errors and dropped packets.

```
akshat_a18@LAPTOP-R86737RN: ~
akshat_a18@LAPTOP-R86737RN:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1468
      inet 172.27.224.255  netmask 255.255.240.0  broadcast 172.27.239.255
      inet6 fe80::215:5dff:feff:c178  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:ff:c1:78  txqueuelen 1000  (Ethernet)
          RX packets 1243  bytes 399111 (399.1 KB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 306  bytes 31610 (31.6 KB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
          RX packets 6  bytes 234 (234.0 B)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 6  bytes 234 (234.0 B)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

akshat_a18@LAPTOP-R86737RN:~$
```

**whois :** whois is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource.

```
akshat_a18@LAPTOP-R86737RN:~$ whois www.google.com
No match for "WWW.GOOGLE.COM".
>>> Last update of whois database: 2022-11-27T11:05:39Z <<<

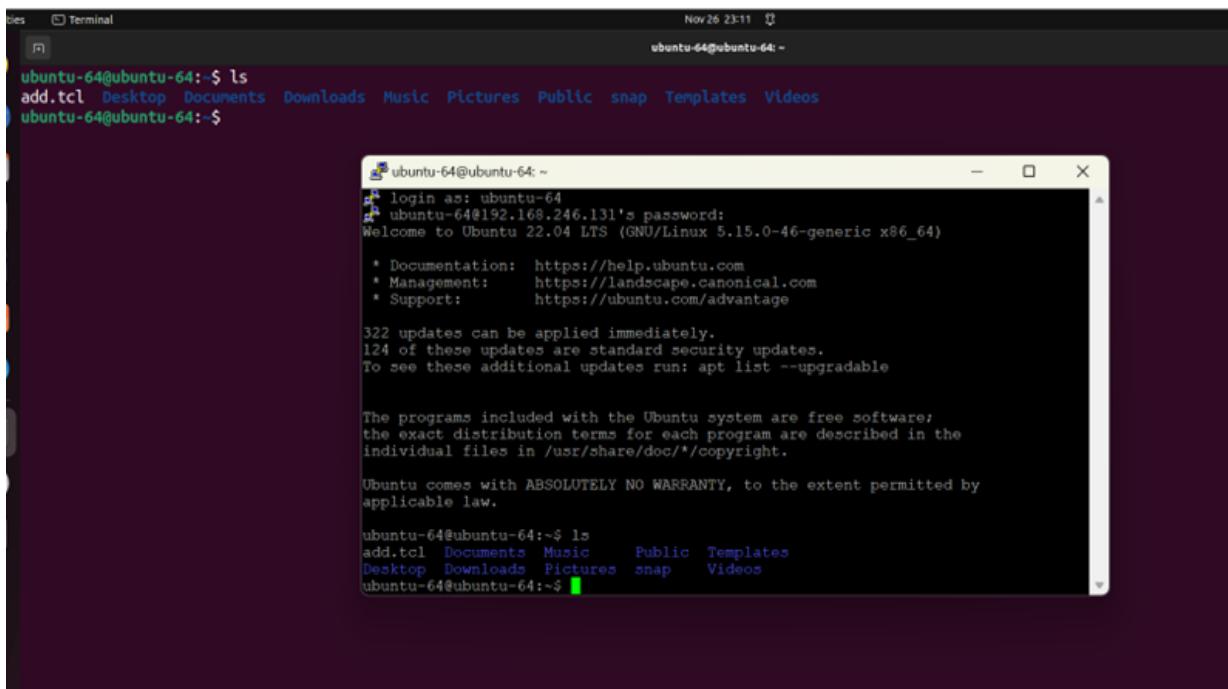
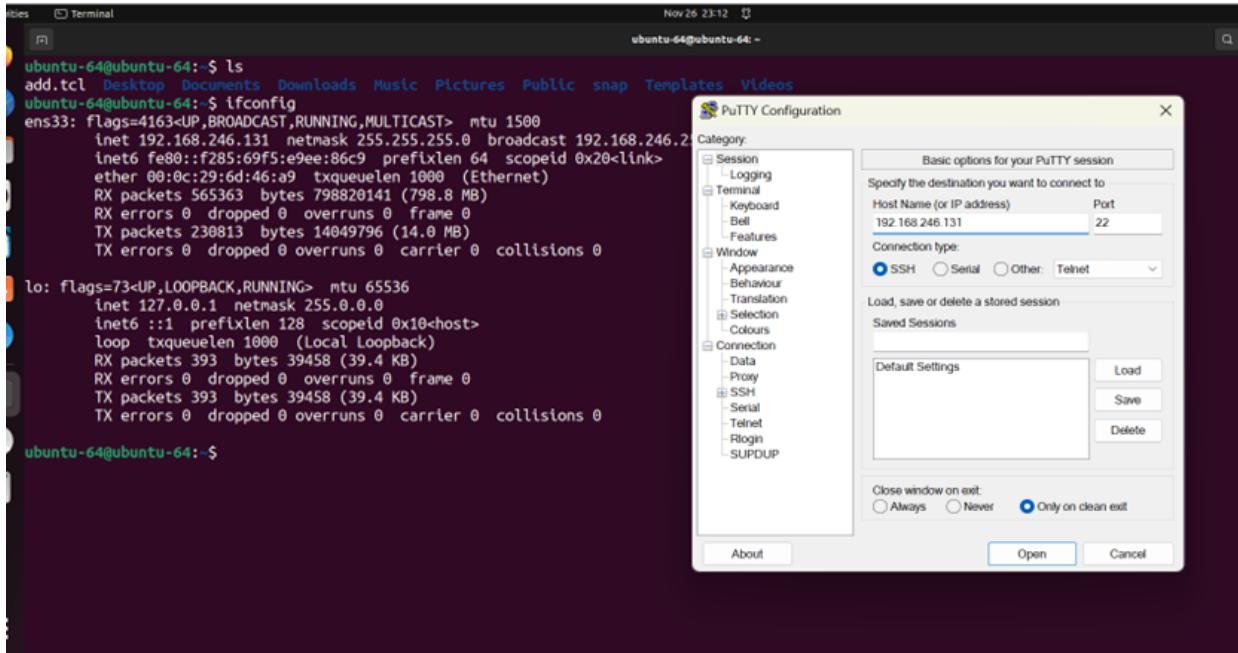
NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for
information purposes only, and to assist persons in obtaining information
about or related to a domain name registration record. VeriSign does not
guarantee its accuracy. By submitting a Whois query, you agree to abide
by the following terms of use: You agree that you may use this Data only
for lawful purposes and that under no circumstances will you use this Data
to: (1) allow, enable, or otherwise support the transmission of mass
unsolicited, commercial advertising or solicitations via e-mail, telephone,
or facsimile; or (2) enable high volume, automated, electronic processes
that apply to VeriSign (or its computer systems). The compilation,
repackaging, dissemination or other use of this Data is expressly
prohibited without the prior written consent of VeriSign. You agree not to
use electronic processes that are automated and high-volume to access or
query the Whois database except as reasonably necessary to register
domain names or modify existing registrations. VeriSign reserves the right
to restrict your access to the Whois database in its sole discretion to ensure
operational stability. VeriSign may restrict or terminate your access to the
Whois database for failure to abide by these terms of use. VeriSign
reserves the right to modify these terms at any time.
```

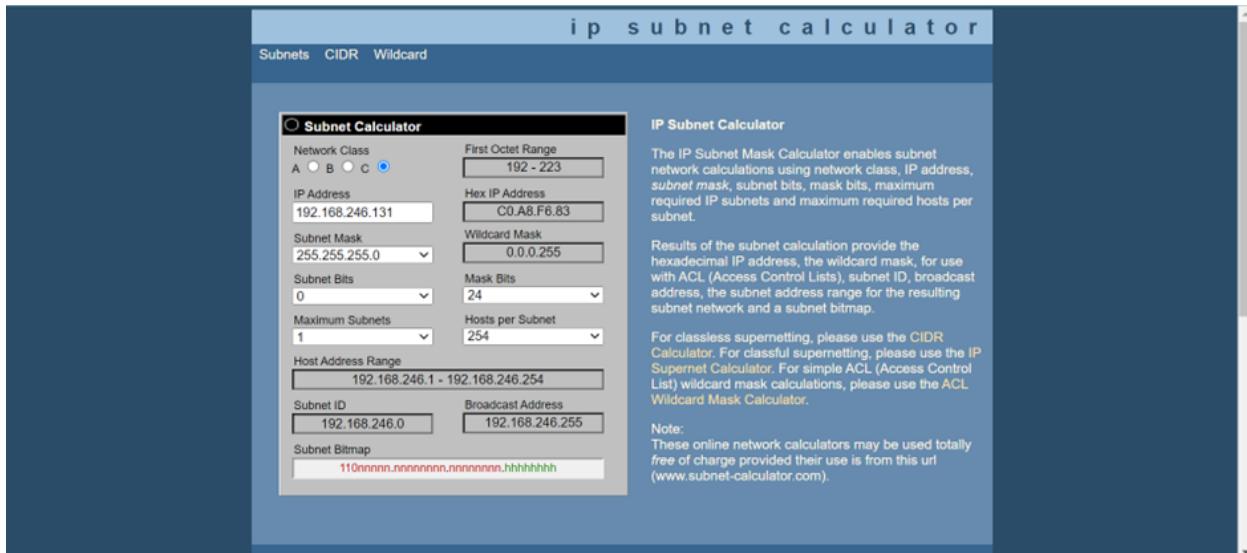
The Registry database contains ONLY .COM, .NET, .EDU domains and Registrars.

```
akshat_a18@LAPTOP-R86737RN:~$
```

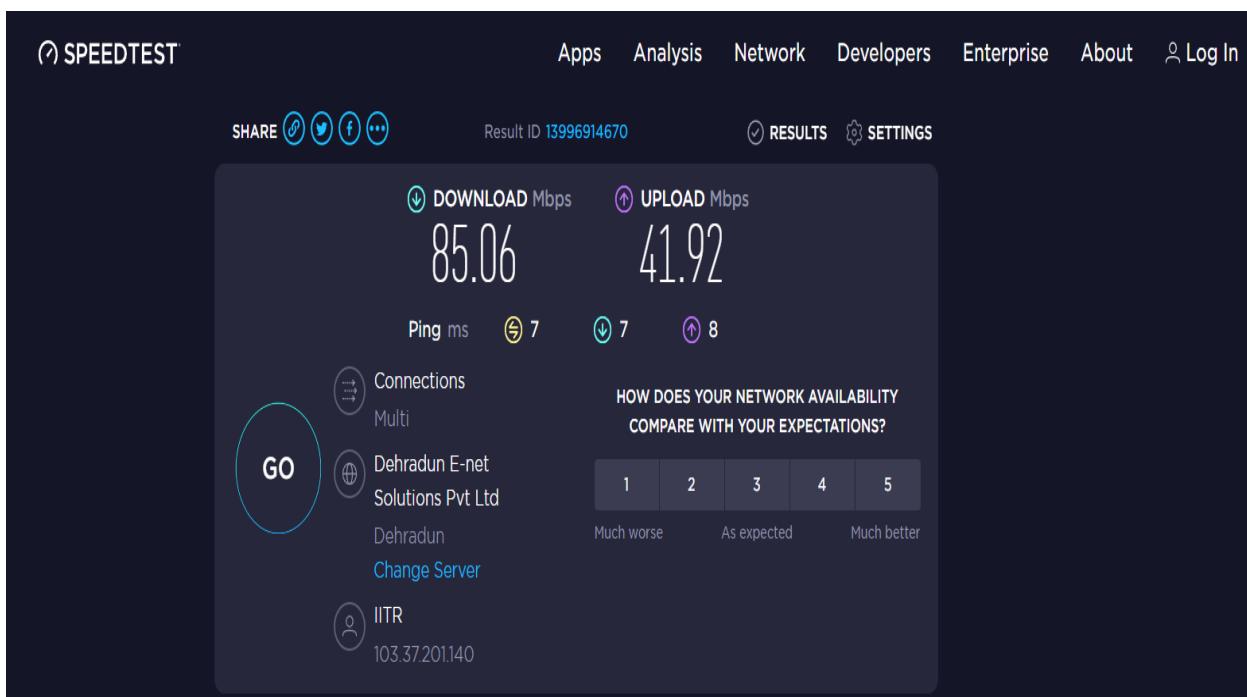
PuTTY/Tera Term : It is used for remote access to another host on the same network.



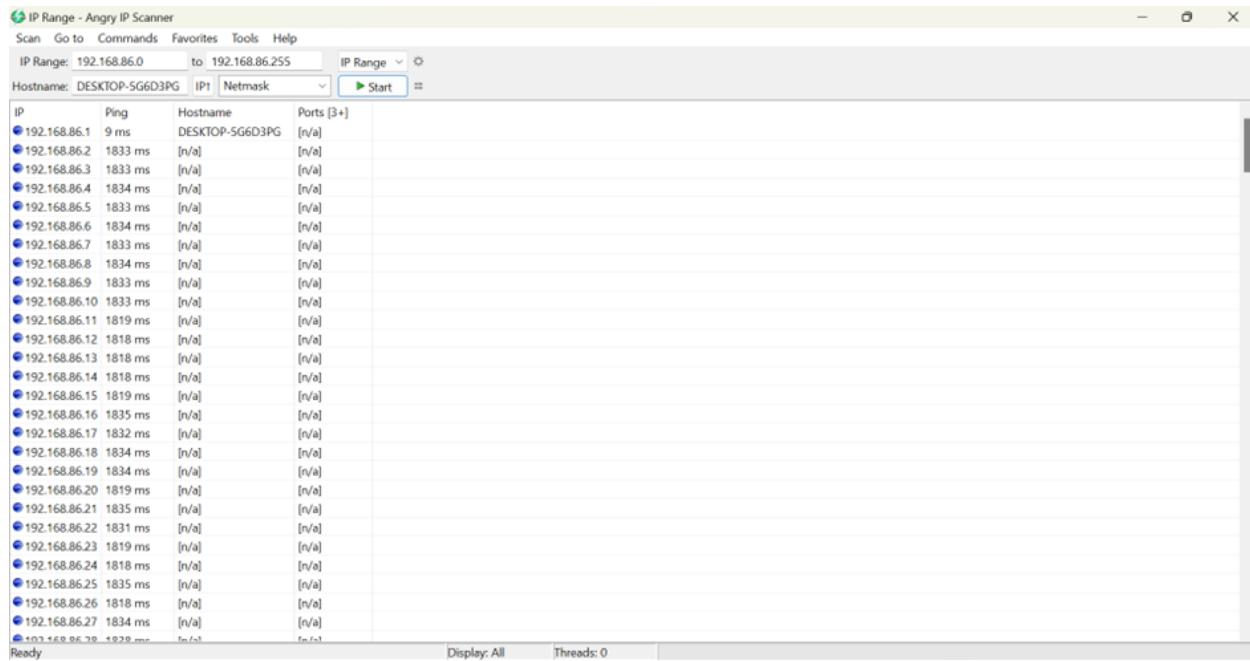
**Subnet and IP Calculator** : They give us a lot of information about subnet and related information in a jiffy which will take a lot of time for humans to calculate.



**Speed test** : It helps us to identify how much bandwidth we are having for in (download) and out (upload) on our network.



**IP Scanner :** There are many devices on a network that don't have a quick interface to tell us what their addresses are such as printers and scanners. An IP Scanner can quickly scan a subnet and display us IP addresses of devices that may otherwise not be visible to us because they do not have an interface.



The screenshot shows the interface of the Angry IP Scanner application. At the top, there's a menu bar with Scan, Go to, Commands, Favorites, Tools, and Help. Below the menu, it says "IP Range: 192.168.86.0 to 192.168.86.255". The main window displays a table of scanned hosts:

IP	Ping	Hostname	Ports [3+]
192.168.86.1	9 ms	DESKTOP-5G6D3PG	[n/a]
192.168.86.2	1833 ms	[n/a]	[n/a]
192.168.86.3	1833 ms	[n/a]	[n/a]
192.168.86.4	1834 ms	[n/a]	[n/a]
192.168.86.5	1833 ms	[n/a]	[n/a]
192.168.86.6	1834 ms	[n/a]	[n/a]
192.168.86.7	1833 ms	[n/a]	[n/a]
192.168.86.8	1834 ms	[n/a]	[n/a]
192.168.86.9	1833 ms	[n/a]	[n/a]
192.168.86.10	1833 ms	[n/a]	[n/a]
192.168.86.11	1819 ms	[n/a]	[n/a]
192.168.86.12	1818 ms	[n/a]	[n/a]
192.168.86.13	1818 ms	[n/a]	[n/a]
192.168.86.14	1818 ms	[n/a]	[n/a]
192.168.86.15	1819 ms	[n/a]	[n/a]
192.168.86.16	1835 ms	[n/a]	[n/a]
192.168.86.17	1832 ms	[n/a]	[n/a]
192.168.86.18	1834 ms	[n/a]	[n/a]
192.168.86.19	1834 ms	[n/a]	[n/a]
192.168.86.20	1819 ms	[n/a]	[n/a]
192.168.86.21	1835 ms	[n/a]	[n/a]
192.168.86.22	1831 ms	[n/a]	[n/a]
192.168.86.23	1819 ms	[n/a]	[n/a]
192.168.86.24	1818 ms	[n/a]	[n/a]
192.168.86.25	1835 ms	[n/a]	[n/a]
192.168.86.26	1818 ms	[n/a]	[n/a]
192.168.86.27	1834 ms	[n/a]	[n/a]
192.168.86.79	1929 ms	[n/a]	[n/a]

At the bottom, there are buttons for "Display: All" and "Threads: 0".