

ASSIGNMENT -02

WIRESHARK

SUBMITTED BY:

KOLLI JOGI NAIDU

B190605CS

CSE-B

Q1.Execute the following command in the terminal,

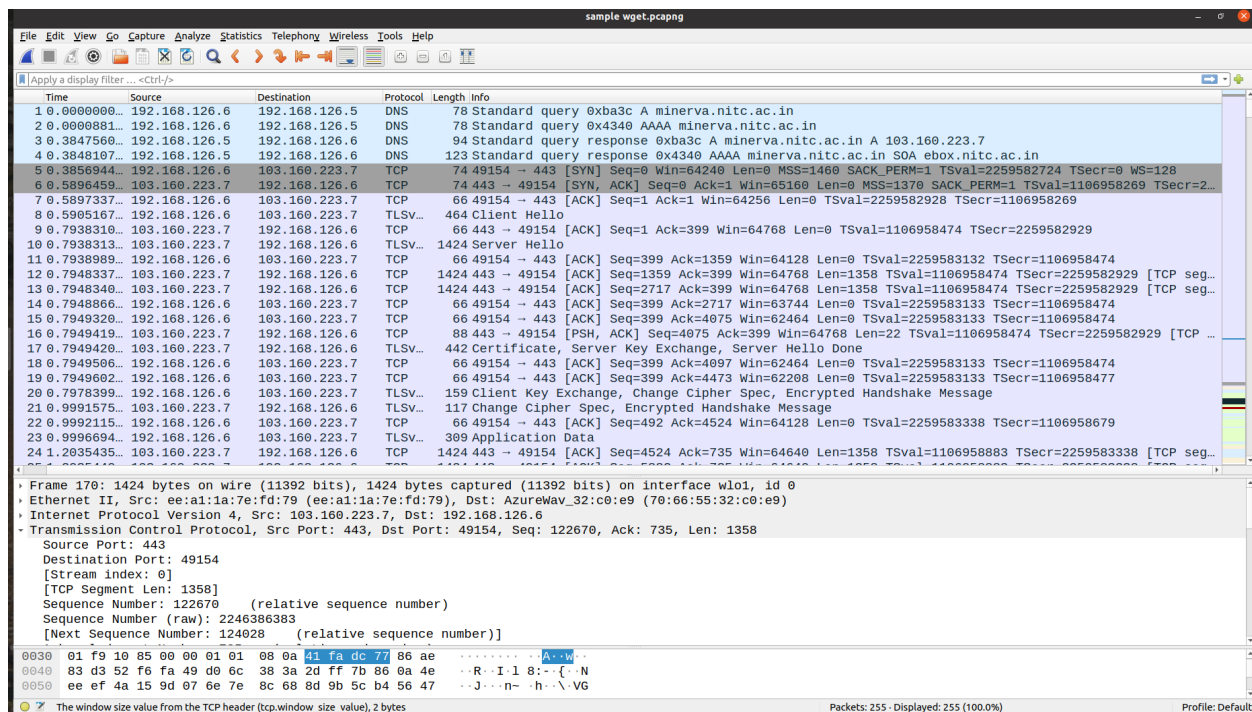
wget https://minerva.nitc.ac.in/sites/default/files/attachments/news/TT_Winter2021-2022%20%281%29.pdf

Parallely run the wireshark tool. Note down your network analysis of the command.

Sol: Once we run the command , the DNS server is looked up and it sends the IP address to initiate a tcp connection with the server located at port 443. The 3-way handshake will be made(packets 5,6,7). The data which is in pdf format will be downloaded(sent through tcp packets - we can see a stream of tcp packets sending data). The segments will be reassembled and the pdf will be downloaded into our local storage. Since the tcp connection should be reliable , the lost data packets will be retransmitted(figure 3) again to the client. In the end , the tcp connection is closed with FIN set to 1.

Source - IP address: **192.168.126.6** PORT: **49154**

Destination- IP address: **103.160.223.7** PORT: **443(HTTPS)**



sample wget.pcapng

FileEditViewGoCapture Analyze StatisticsTelephonyWirelessToolsHelp

sample wget.pcapng

Time	Source	Destination	Protocol	Length	Info
201.1.898760290	192.168.126.6	103.160.223.7	TCP	66	49154 → 443 [FIN, ACK] Seq=735 Ack=143907 Win=182784 Len=0 TSval=2259584237 TSecr=1106959483
202.1.976237692	103.160.223.7	192.168.126.6	TCP	66	443 → 49154 [FIN, ACK] Seq=143907 Ack=736 Win=64640 Len=0 TSval=1106959748 TSecr=2259584237
203.1.976280613	192.168.126.6	103.160.223.7	TCP	66	49154 → 443 [ACK] Seq=736 Ack=143908 Win=182784 Len=0 TSval=2259584315 TSecr=1106959748
204.6.323977840	ee:a1:1a:7e:fd:...	AzureWav_32:c0:...	ARP	42	Who has 192.168.126.6? Tell 192.168.126.5
205.6.324097840	AzureWav_32:c0:...	ee:a1:1a:7e:fd:...	ARP	42	192.168.126.6 is at 70:66:55:32:c0:e9
206.14.7494194.	192.168.126.6	192.168.126.5	DNS	89	Standard query 0xc457 A connectivity-check.ubuntu.com
207.15.1307327.	192.168.126.5	192.168.126.6	DNS	121	Standard query response 0xc457 A connectivity-check.ubuntu.com A 35.232.111.17 A 35.224.170.84
208.15.1324982.	192.168.126.6	35.224.170.84	TCP	74	53398 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=815583347 TSecr=0 WS=128
209.15.5401109.	35.224.170.84	192.168.126.6	TCP	74	80 → 53398 [SYN, ACK] Seq=0 Ack=1 Win=64768 Len=0 MSS=1370 SACK_PERM=1 TSval=2174886244 TSecr=8155833...
210.15.5401965.	192.168.126.6	35.224.170.84	TCP	66	53398 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=815583755 TSecr=2174886244
211.15.5403224.	192.168.126.6	35.224.170.84	HTTP	153	GET / HTTP/1.1
212.16.5713649.	192.168.126.6	35.224.170.84	TCP	153	[TCP Retransmission] 53398 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=87 TSval=815584786 TSecr=2174886...
213.17.9152665.	192.168.126.6	35.224.170.84	TCP	153	[TCP Retransmission] 53398 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=87 TSval=815586139 TSecr=2174886...
214.20.692980.	192.168.126.6	35.224.170.84	TCP	153	[TCP Retransmission] 53398 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=87 TSval=815588914 TSecr=2174886...
215.26.0753251.	192.168.126.6	35.224.170.84	TCP	153	[TCP Retransmission] 53398 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=87 TSval=815594290 TSecr=2174886...
216.35.7460461.	192.168.126.6	35.224.170.84	TCP	66	53398 → 80 [FIN, ACK] Seq=88 Ack=1 Win=64256 Len=0 TSval=815603960 TSecr=2174886244
217.35.7615692.	192.168.126.6	35.232.111.17	TCP	74	43464 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2063191176 TSecr=0 WS=128
218.36.2264131.	35.224.170.84	192.168.126.6	TCP	54	80 → 53398 [RST] Seq=1 Win=0 Len=0
219.36.2264134.	35.232.111.17	192.168.126.6	TCP	74	80 → 43464 [SYN, ACK] Seq=0 Ack=1 Win=64768 Len=0 MSS=1370 SACK_PERM=1 TSval=421683738 TSecr=20631911...
220.36.2265360.	192.168.126.6	35.232.111.17	TCP	66	43464 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2063191641 TSecr=421683738
221.36.2267609.	192.168.126.6	35.232.111.17	HTTP	153	GET / HTTP/1.1
222.36.7489323.	192.168.126.6	192.168.126.5	DNS	76	Standard query 0xddcb A daisy.ubuntu.com
223.36.7491946.	192.168.126.6	192.168.126.5	DNS	76	Standard query 0xd695 AAAA daisy.ubuntu.com
224.36.7492487.	192.168.126.6	35.232.111.17	TCP	74	43466 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2063192164 TSecr=0 WS=128
225.36.7535812.	192.168.126.5	192.168.126.6	DNS	76	Standard query response 0xd695 AAAA daisy.ubuntu.com
226.36.7852471.	35.232.111.17	192.168.126.6	TCP	66	80 → 43464 [ACK] Seq=1 Ack=88 Win=65024 Len=0 TSval=421684248 TSecr=2063191641

Frame 170: 1424 bytes on wire (11392 bits), 1424 bytes captured (11392 bits) on interface wlo1, id 0
 Ethernet II, Src: ee:a1:1a:7e:fd:79 (ee:a1:1a:7e:fd:79), Dst: AzureWav_32:c0:e9 (70:66:55:32:c0:e9)
 Internet Protocol Version 4, Src: 103.160.223.7, Dst: 192.168.126.6
 Transmission Control Protocol, Src Port: 443, Dst Port: 49154, Seq: 122670, Ack: 735, Len: 1358

Source Port: 443
 Destination Port: 49154
 [Stream index: 9]
 [TCP Segment Len: 1358]
 Sequence Number: 122670 (relative sequence number)
 Sequence Number (raw): 2246386383
 [Next Sequence Number: 124028 (relative sequence number)]
 Acknowledgment Number: 735 (relative ack number)

```

0030 01 f9 10 85 00 00 01 01 08 0a 41 fa dc 77 86 ae .....A..w...
0040 83 d3 52 f6 fa 49 d0 6c 38 3a 2d ff 7b 86 0a 4e ...R..I..l.8:--..N
0050 ee ef 4a 15 9d 97 6e 7e 8c 68 8d 9b 5c b4 56 47 ...J...n--h-..VG

```

The window size value from the TCP header (tcp.window_size_value), 2 bytes

Packets: 255 - Displayed: 255 (100.0%) Profile: Default

Q2: Consider the pcap file, File001.pcap. The file contains captured packets sent over the network. It is noticed the system has made a connection to an unsecured host system and the user has sent his credentials over plaintext. Investigate File001.pcap to unearth the login credentials.

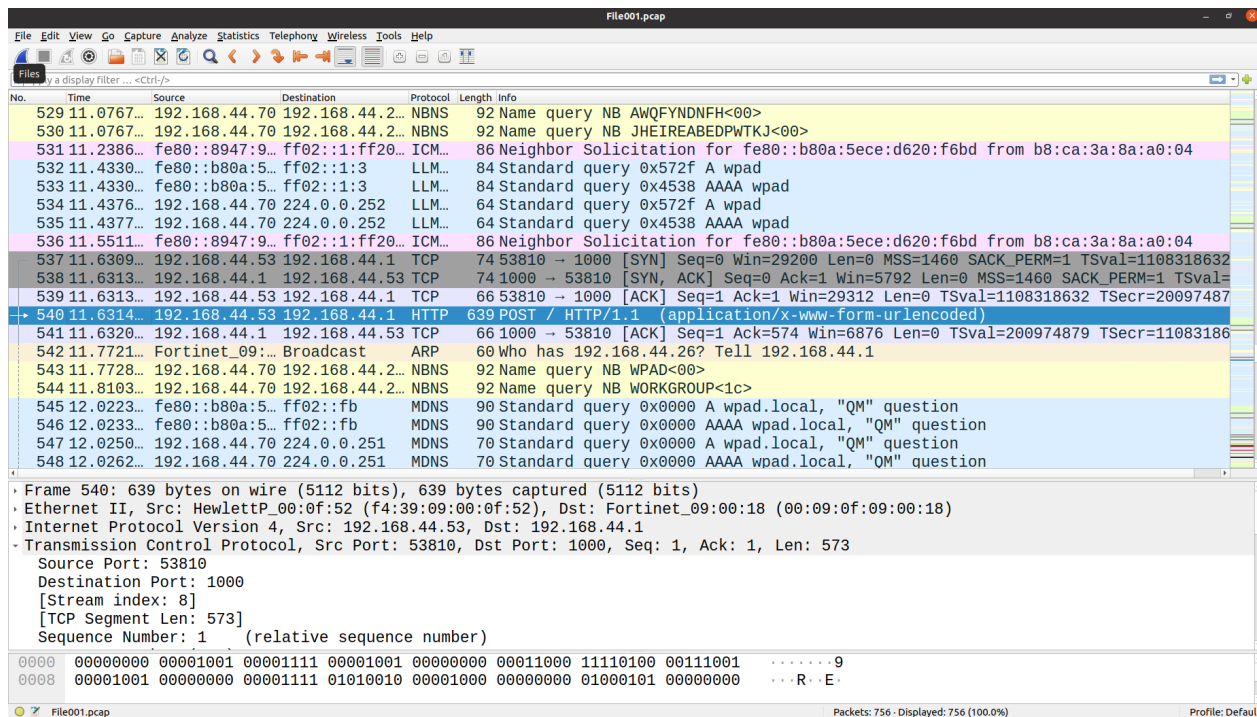
a. Indicate the IP addresses, Source and Destination, of the communicating end systems in which the login credentials are found.

Sol: The client is communicating with port 1000 and after entering into the details over plain text , the information could be seen at packet number 540.

The ip address of source: **192.168.44.53**

The ip address of destination: **192.168.44.1**

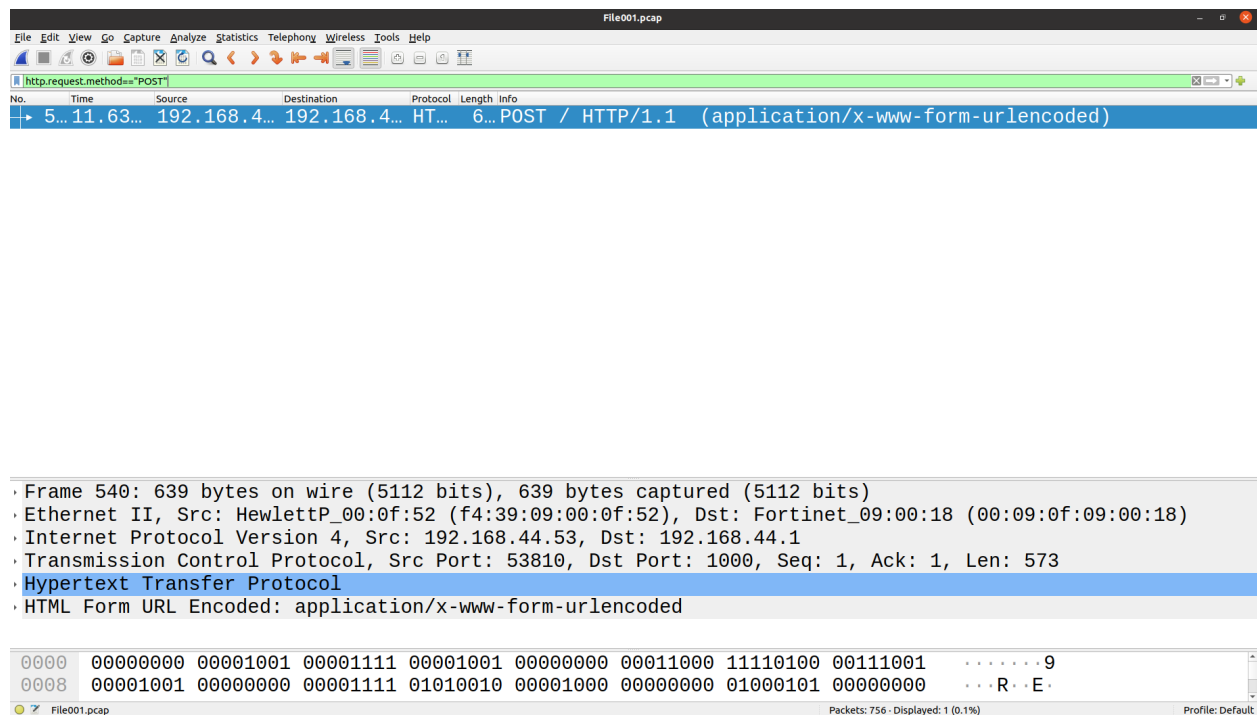
The corresponding details can be seen in packet:



b. Determine the protocol over which the user credentials are sent.

Sol: Since the user is sending the credentials, the underlying protocol would be **http**. We can also get the user credentials and the corresponding details by filtering the wireshark with **http.type.method=="POST"** (As the user is sending his credentials by http post method).

The following image shows the corresponding details:



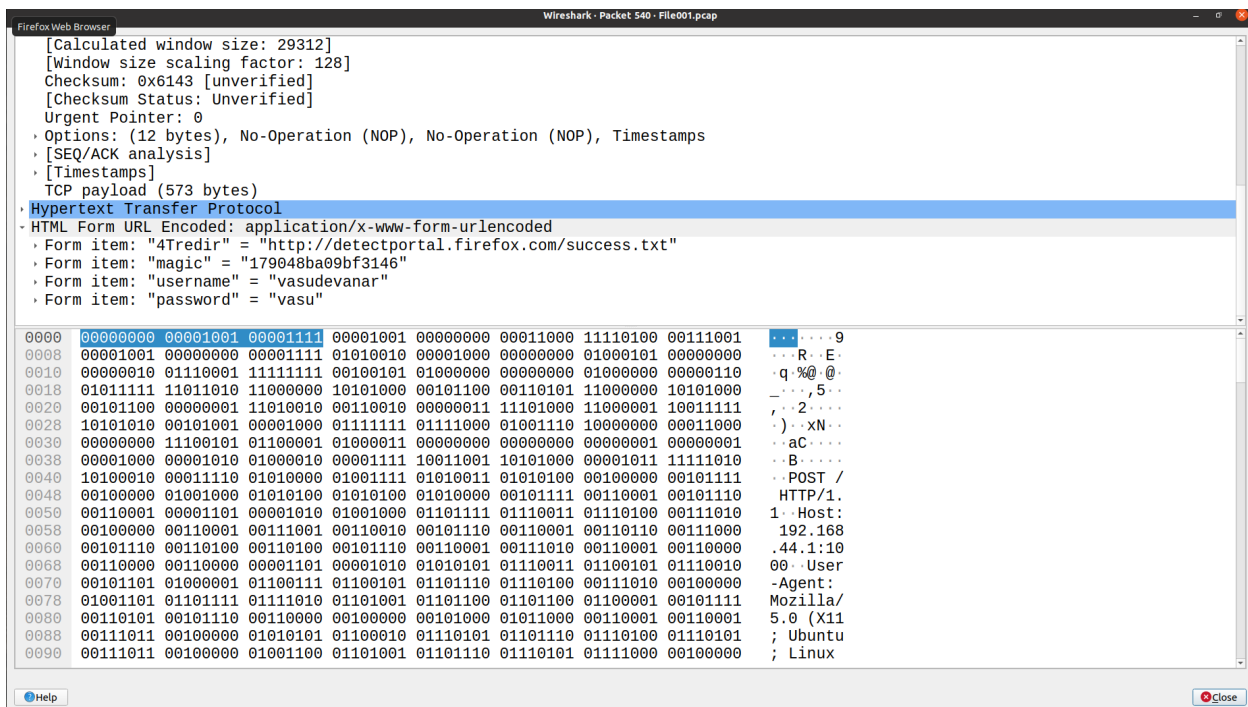
c. What are the login credentials?

Sol: The login credentials are:

Username = **"vasudevanr"**

Password = **"vasu"**

The above login credentials can be seen in the below wireshark image:



Q3: Consider the pcap file, File002.pcap. The file contains captured packets. Consider the packets numbered 27 and 32. Fill up the header details for the packets 27 and 32

Sol: **TCP HEADER**

The tcp header will contain the following fields:

- 1. Source port:** It identifies the sender of the application and it is of 16 bit length.
- 2. Destination port:** It identifies the receiver of the application and it is of 16 bit length.
- 3. Sequence number:** the unique number assigned to each byte of data contained in the tcp segment. It is of 32 bit length.
- 4. Acknowledge number:** It contains the sequence number of the data byte that receiver expects to receive next from the sender. It is of 32 bit length and is always the sequence number of the last received data plus 1.
- 5. Header Length(Data Offset):** It specifies the length of the tcp header and it is of 4 bit length. A scaling factor of 4 is used to represent the length since the tcp header length lies in the range : [20B, 60B].

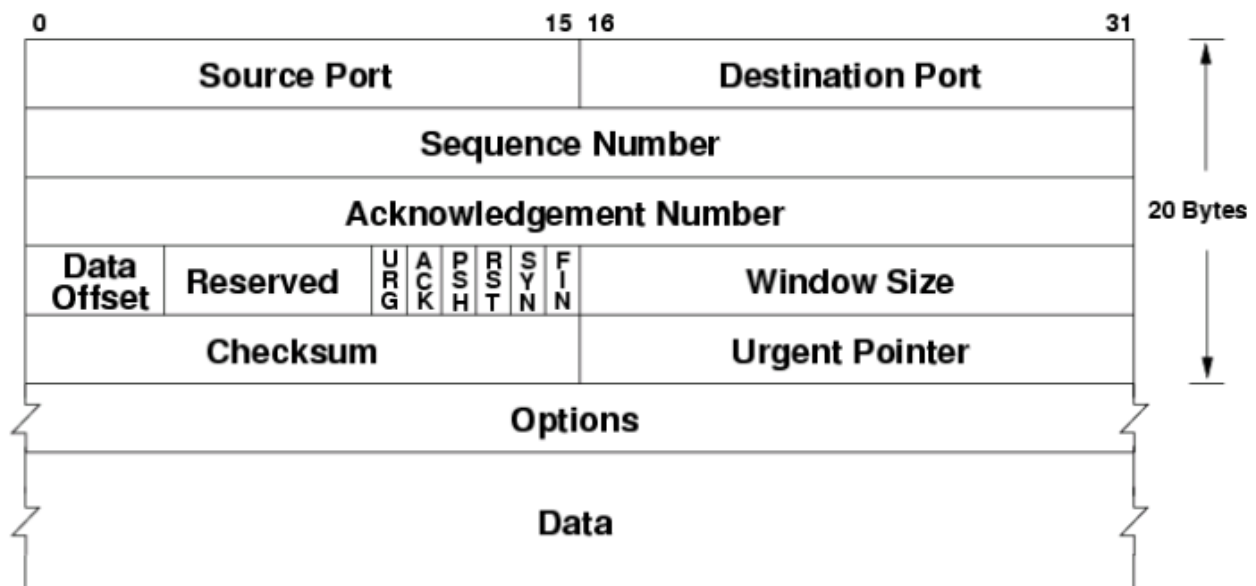


Figure 1: TCP Header

6. Reserved Bits: The length of 6 bits are reserved for the future purpose and are not used.

7.Flags: These are used to indicate the particular state of the tcp connection and also provide some useful information regarding troubleshooting and other connections.

- **URG:** It is used to treat certain data on an urgent basis.If the bit is set to 1, the urgent pointer field contains the urgent data location.
- **ACK:** It indicates whether the acknowledge number field is valid or not.
- **PSH:** It is used to push the entire buffer immediately to the application.
- **RST:** It is used to reset the tcp connection.
- **SYN:** It is used to synchronize the sequence numbers.
- **FIN:** It is used to terminate the connection.

8. Window size: It specifies the size of the receiving window of the sender and it is of 16 bit length.

9. Check Sum: It verifies the integrity of the tcp payload(i.e. Error control) and it is of 16 bit length.

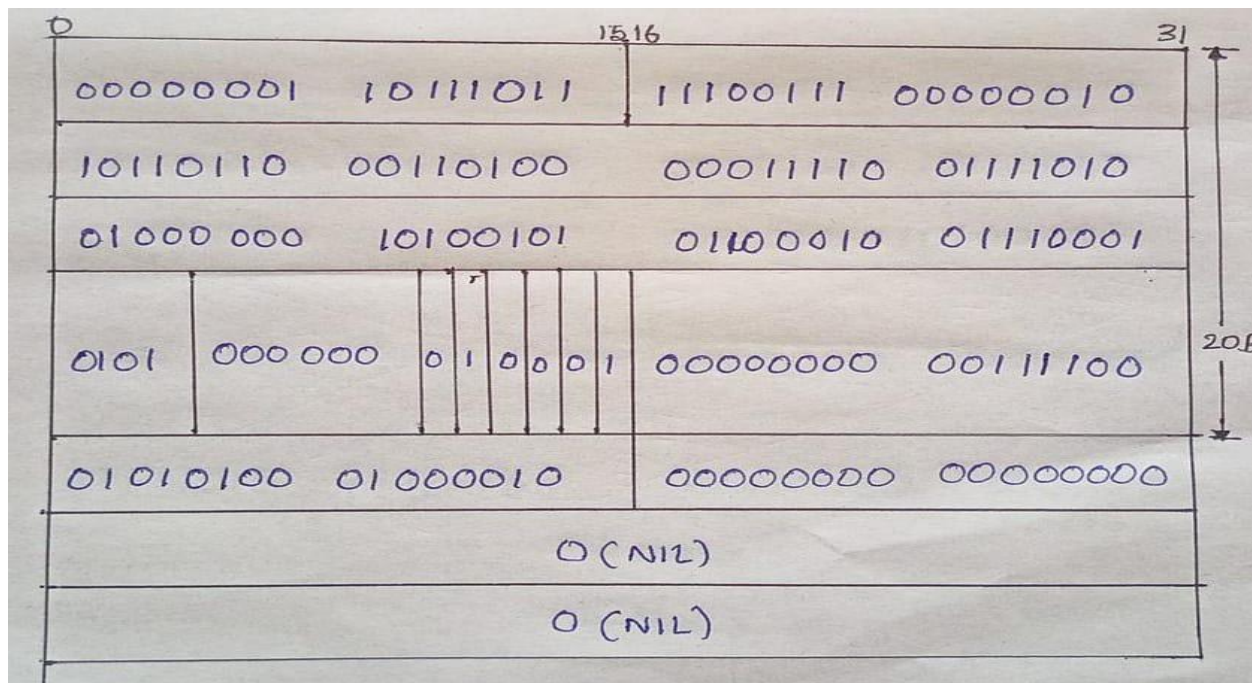
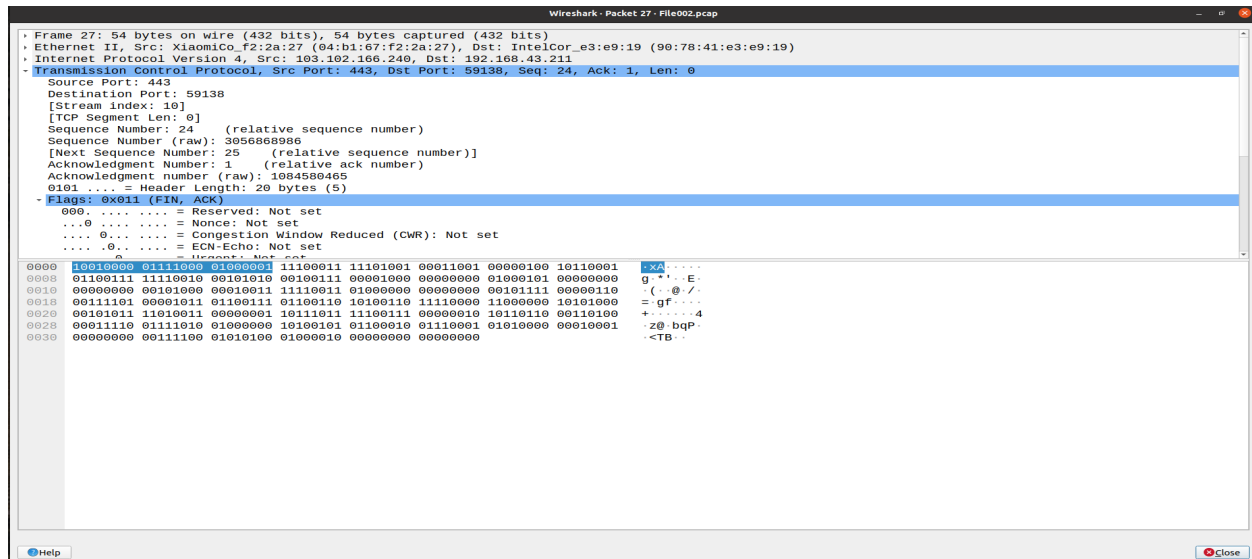
10. Urgent Pointer: It indicates how much data in the current segment counting from the first byte is urgent and is valid iff URG flag is set to 1.It is of 16 bit length.

11. Options: It is used for several purposes including padding ,time stamp, window size extension, parameter negotiation and it varies from 0-40 bytes.

12. Data: It specifies the data of the current tcp segment.

The following images show the tcp header details of both packets observed in wireshark and in listed in above mentioned header format:

1.Packet 27:



2.Packet 32:

