ASSIGNMENT -02

WIRESHARK

SUBMITTED BY:

KOLLI JOGI NAIDU

B190605CS

CSE-B

Q1. Execute the following command in the terminal,

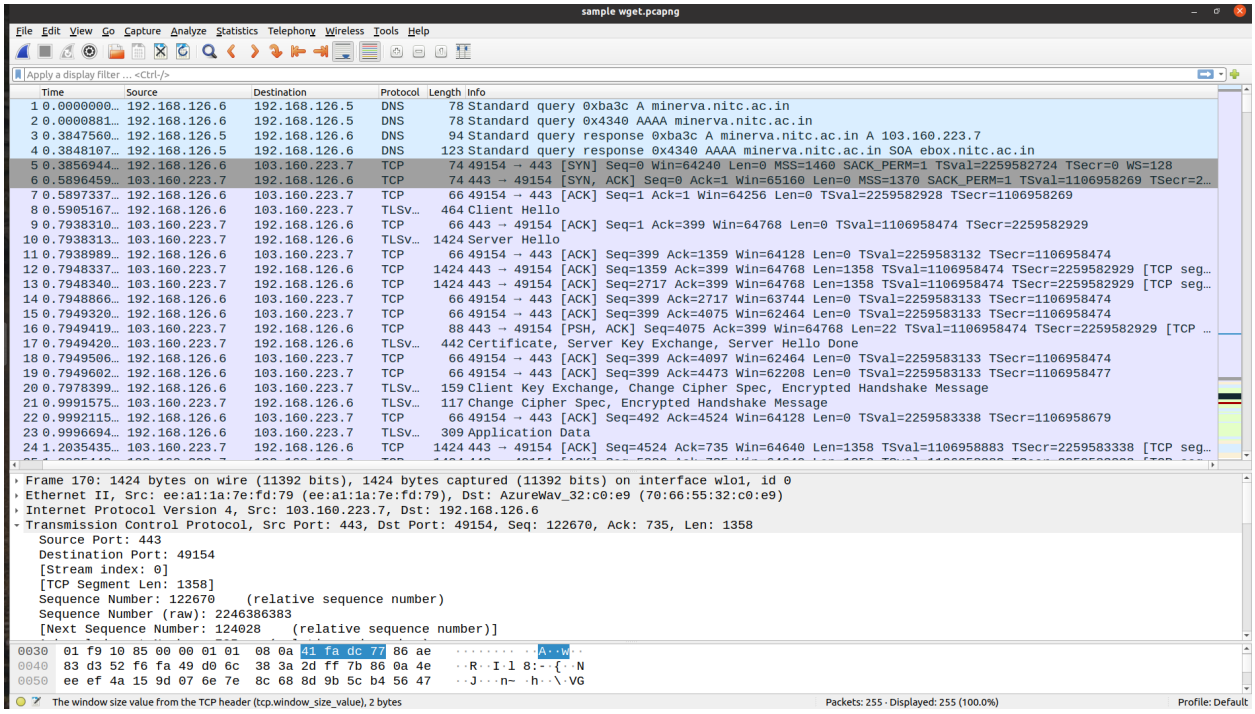
wget https://minerva.nitc.ac.in/sites/default/files/attachments/news/TT_Winter2021-2022%20%281%29.pdf

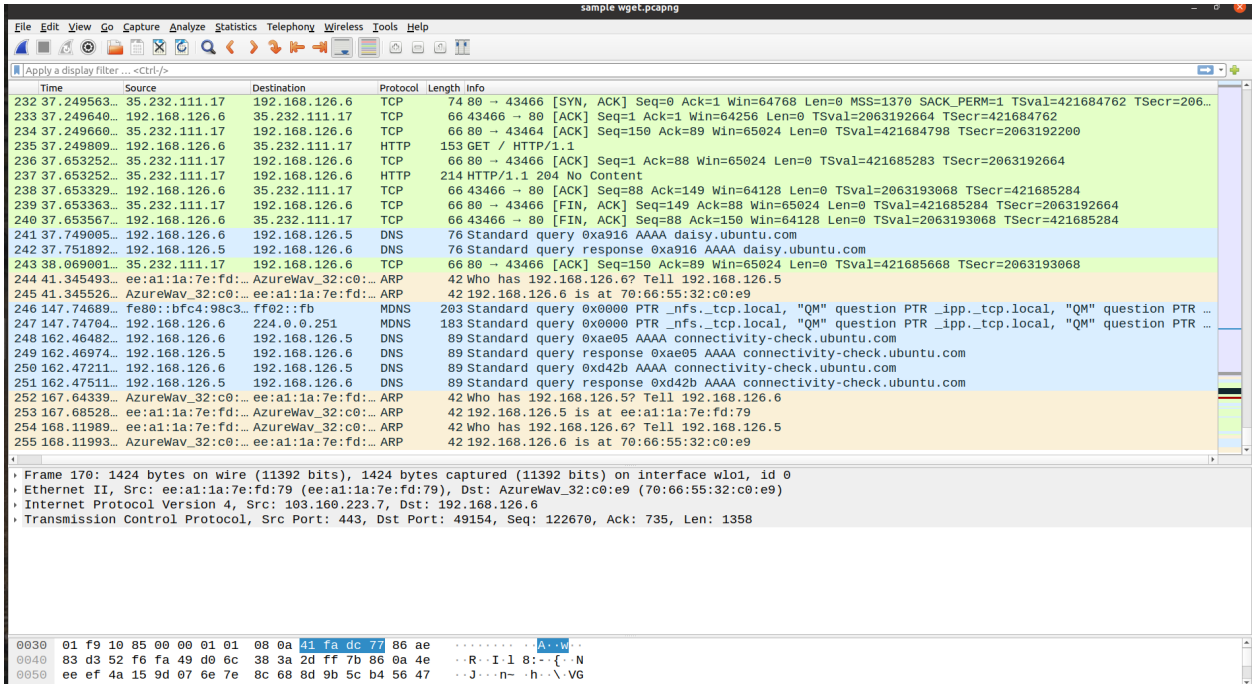
Parallely run the wireshark tool. Note down your network analysis of the command.

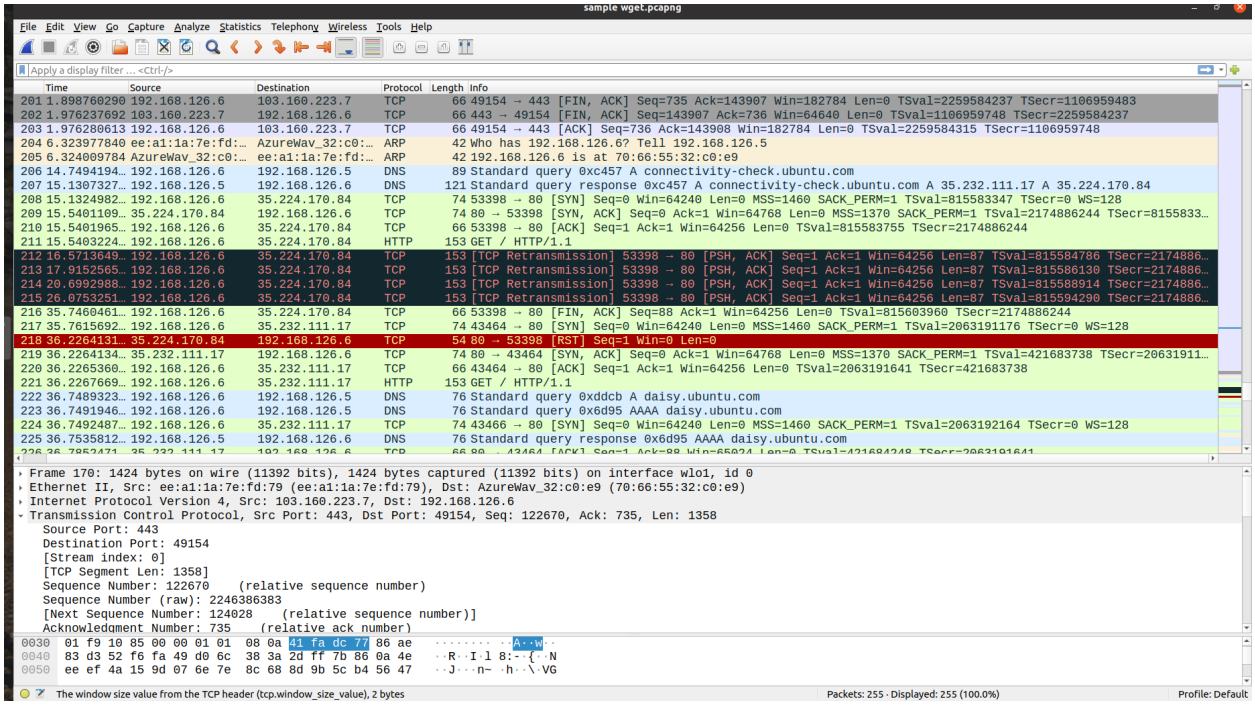
Sol: Once we run the command , the DNS server is looked up and it sends the IP address to initiate a tcp connection with the server located at port 443. The 3-way handshake will be made(packets 5,6,7). The data which is in pdf format will be downloaded(sent through tcp packets - we can see a stream of 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)







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.

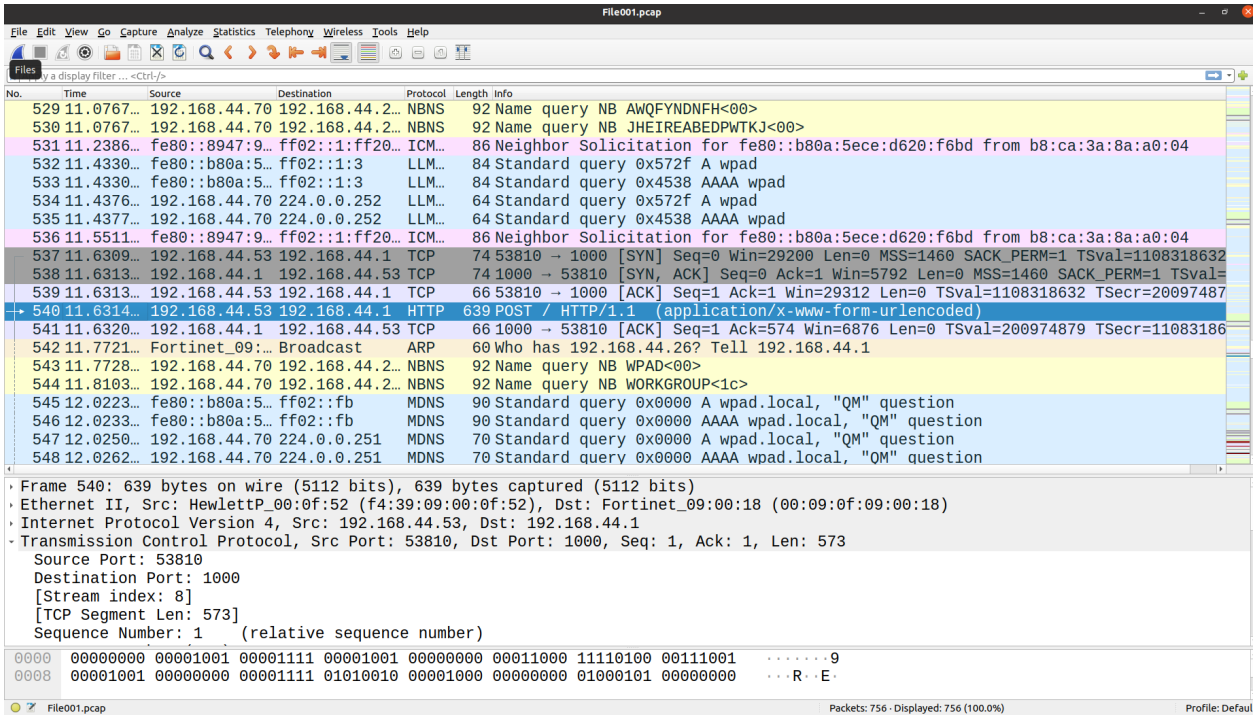
a. Indicate the IP addresses, Source and Destination, of the communicating end systems in which the login credentials are found.

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**

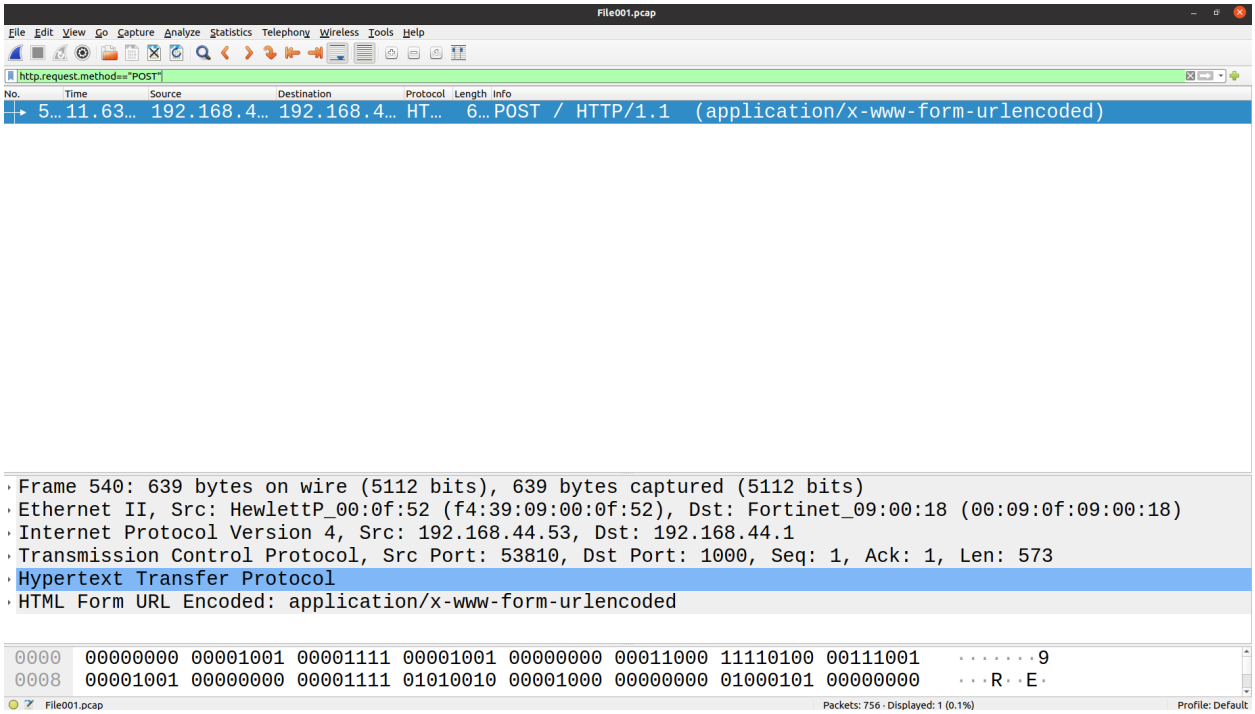
The corresponding details can be seen in packet:



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:



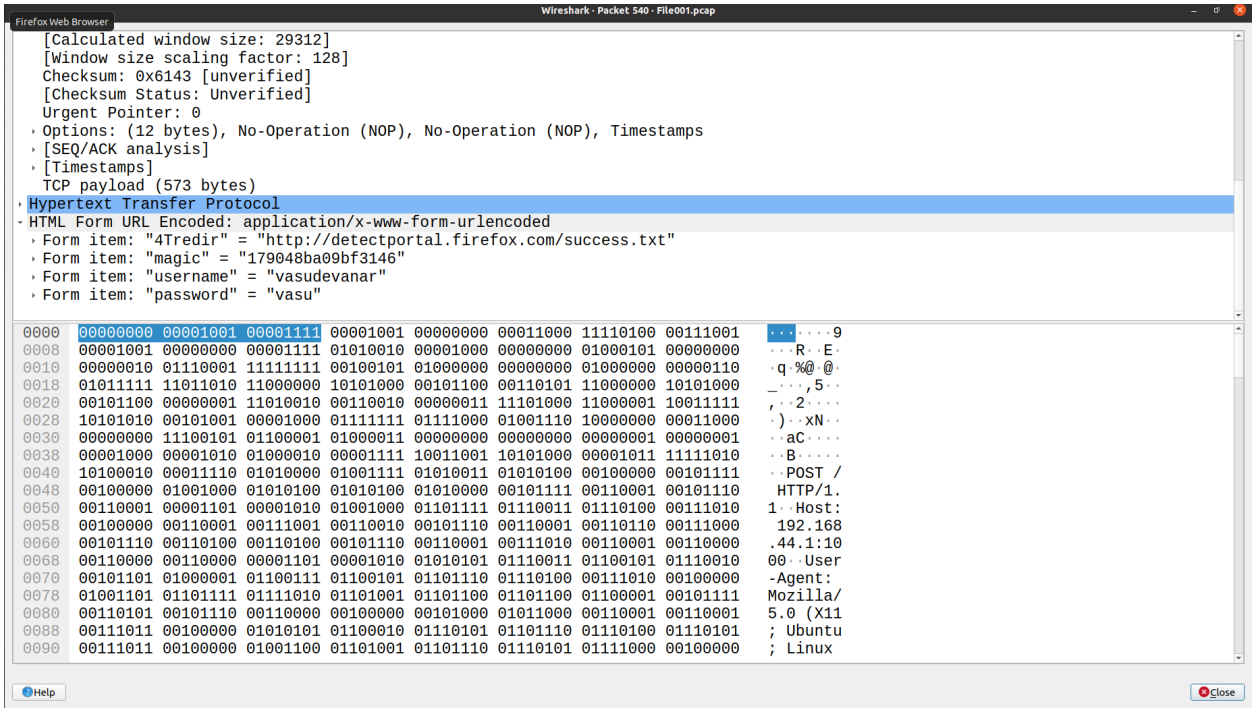
c. What are the login credentials?

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 vreceiver 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].

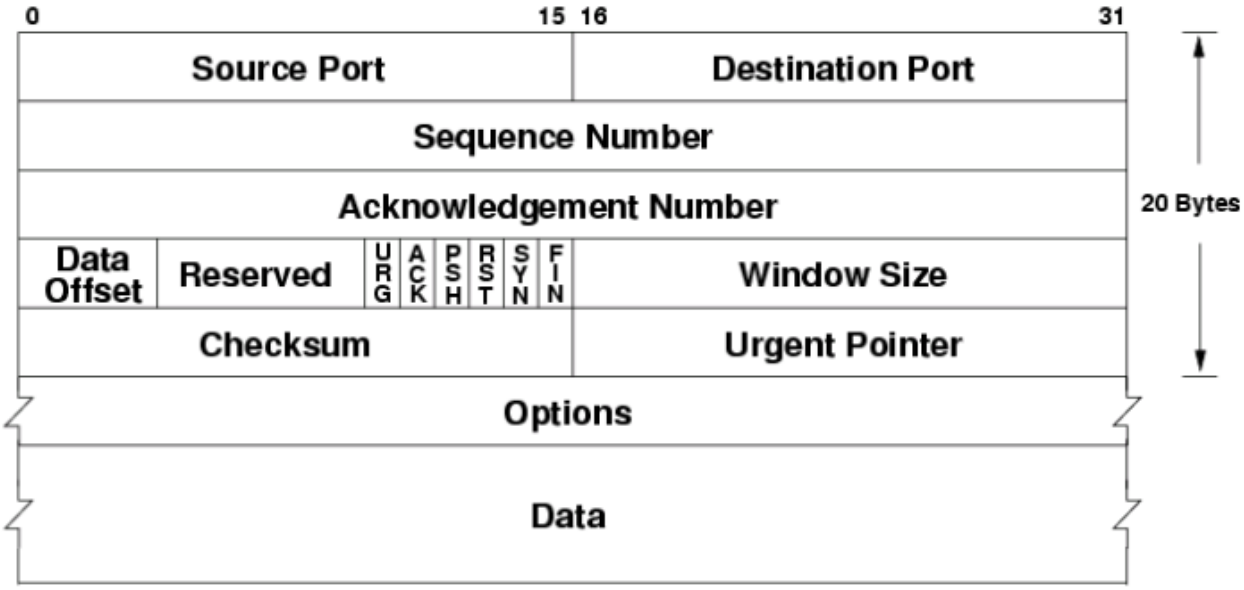


Figure 1: TCP Header

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.

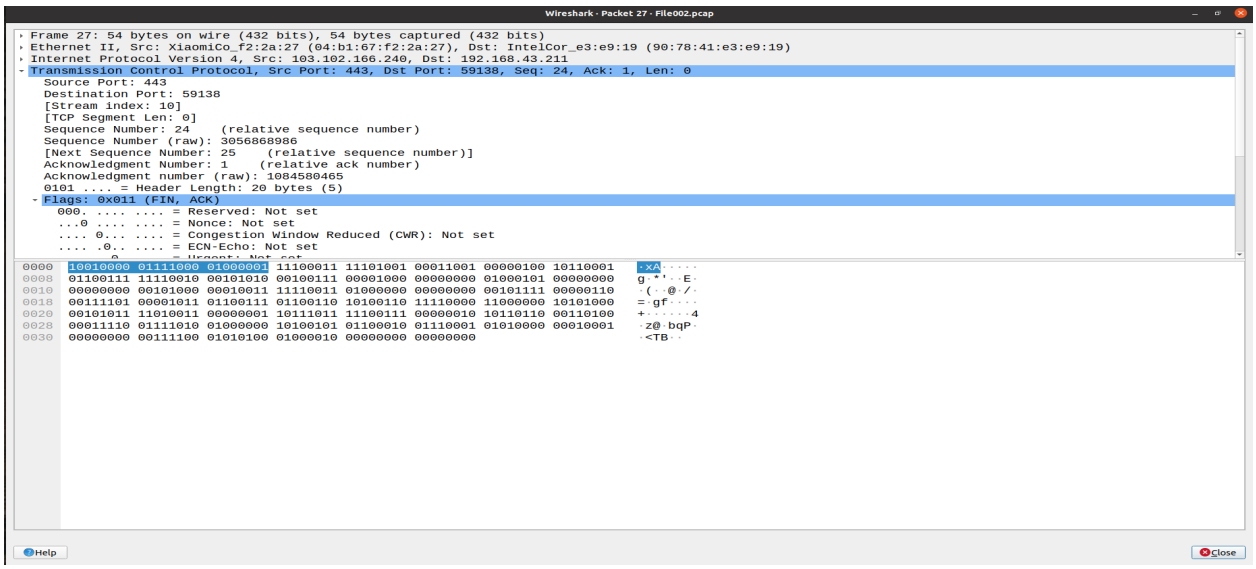
10. Urgent Pointer: It indicates how much data in the current segment counting from the first byte is urgent and is valid iff URG flag is set to 1.It is of 16 bit length.

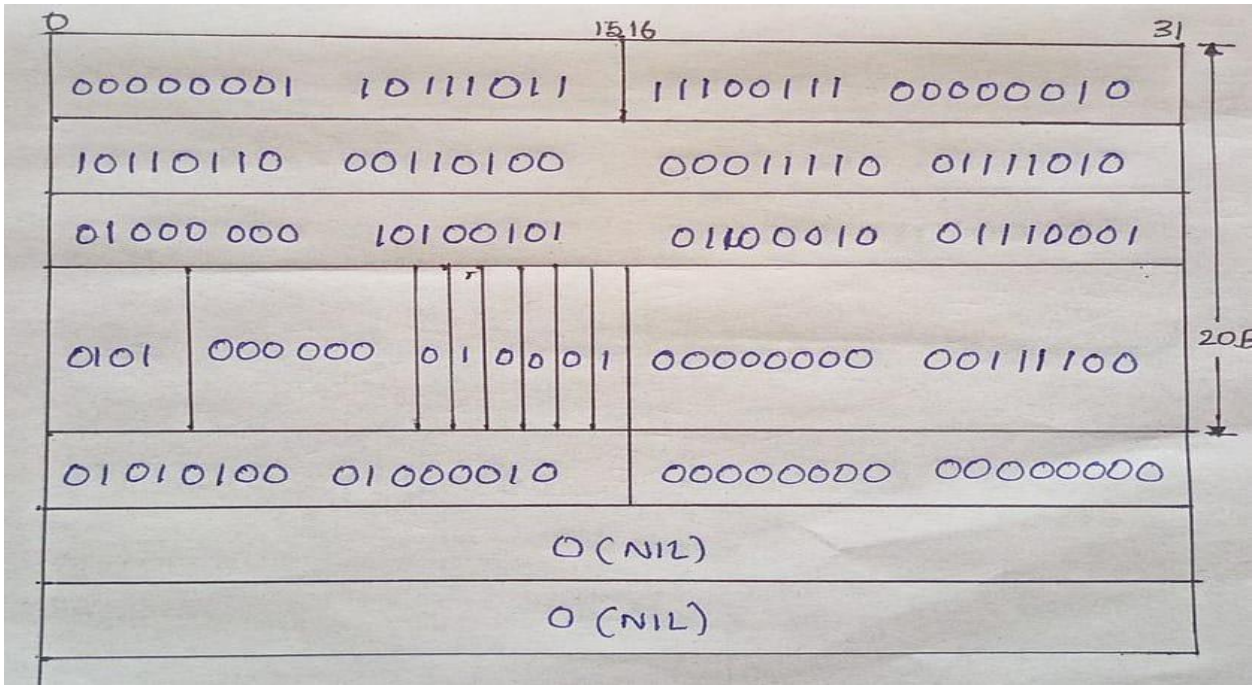
11. Options: It is used for several purposes including padding ,time stamp, window size extension, parameter negotiation and it varies from 0-40 bytes.

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

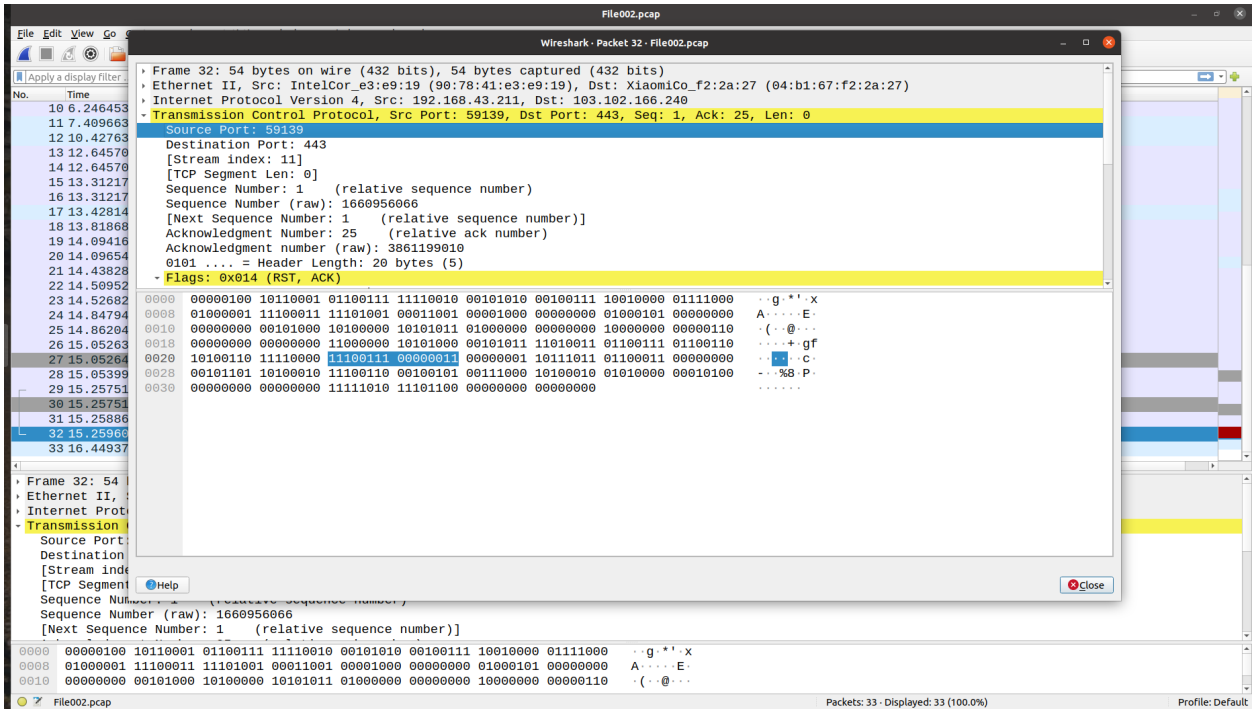
The following images show the tcp header details of both packets observed in wireshark and in listed in above mentioned header format:

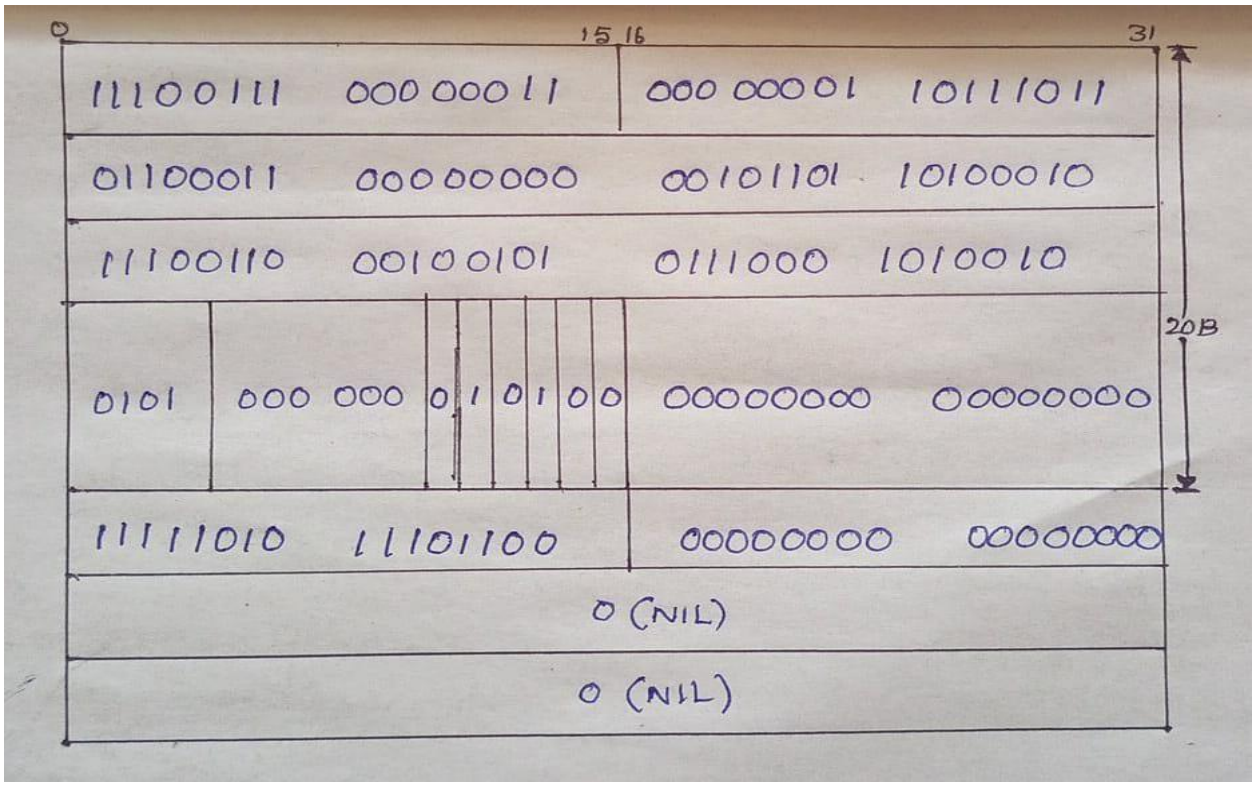
1.Packet 27:





2.Packet 32:


File002.pcap
Wireshark · Packet 32 · File002.pcap
Frame 32: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
Ethernet II, Src: IntelCor_e3:e9:19 (90:78:41:e3:e9:19), Dst: XiaomiCo_f2:2a:27 (04:b1:67:f2:2a:27)
Internet Protocol Version 4, Src: 192.168.43.211, Dst: 103.102.166.240
Transmission Control Protocol, Src Port: 59139, Dst Port: 443, Seq: 1, Ack: 25, Len: 0
Source Port: 59139
Destination Port: 443
[Stream index: 11]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1660956066
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 25 (relative ack number)
Acknowledgment number (raw): 3861199010
0101 .... = Header Length: 20 bytes (5)
Flags: 0x014 (RST, ACK)
0000 00000100 10110001 01100111 11110010 00101010 00100111 10010000 01111000
0008 01000001 11100011 11101001 00011001 00001000 00000000 01000101 00000000
0010 00000000 00101000 10100000 10101011 01000000 00000000 10000000 00000110
0018 00000000 00000000 11000000 10101000 00101011 11010011 01100111 01100110
0020 10100110 11110000 11100111 00000011 00000001 10111011 01100011 00000000
0028 00101101 10100010 11100110 00100101 00111000 10100010 01010000 00010100
0030 00000000 00000000 11111010 11101100 00000000 00000000
Help
Close
Packets: 33 · Displayed: 33 (100.0%)
Profile: Default


0
15 16
31
11100111 00000011 00000001 10111011
01100011 00000000 00101101 10100010
11100110 00100101 0111000 1010010
0101 000 000 0 1 0 1 0 0 00000000 00000000
20B
11111010 11101100 00000000 00000000
0 (NIL)
0 (NIL)