Course Name
Networking and Data Security (COMP-8677)

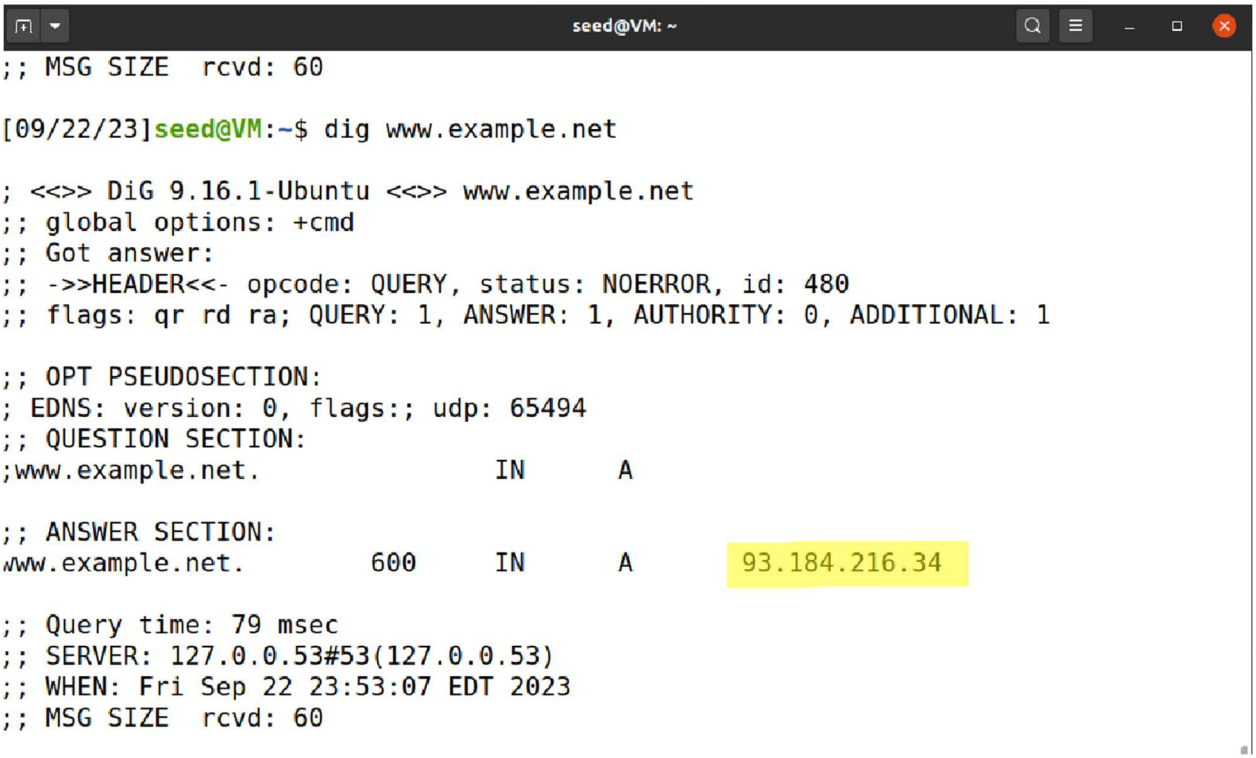
Document Type
Lab 2 Work

Professor
Dr. Shaoquan Jiang

Team - Members	Student ID
Manjinder Singh	110097177
Harbhajan Singh	110100089
Karan Vishavjit	110099867

Ques 1(a) . Try $ dig www.example.net to find out its ip address.

Ans 1(a) IP address of www.example.net is 93.184.216.34



Screenshot 1

Ques 1(b) run Wireshark on your VM, then $ dig www.example.net and stop wireshark. Look at the DNS request packet (using filter DNS to find it easily), confirm that the transport layer protocol is UDP. What are the values of this UDP header (you need to first check the header fields learned in class)?

Ans 1(b) On looking at the DNS request packet(using filter DNS to find it easily), it is confirm that the transport layer protocol is UDP and The field values of UDP header are as follows:

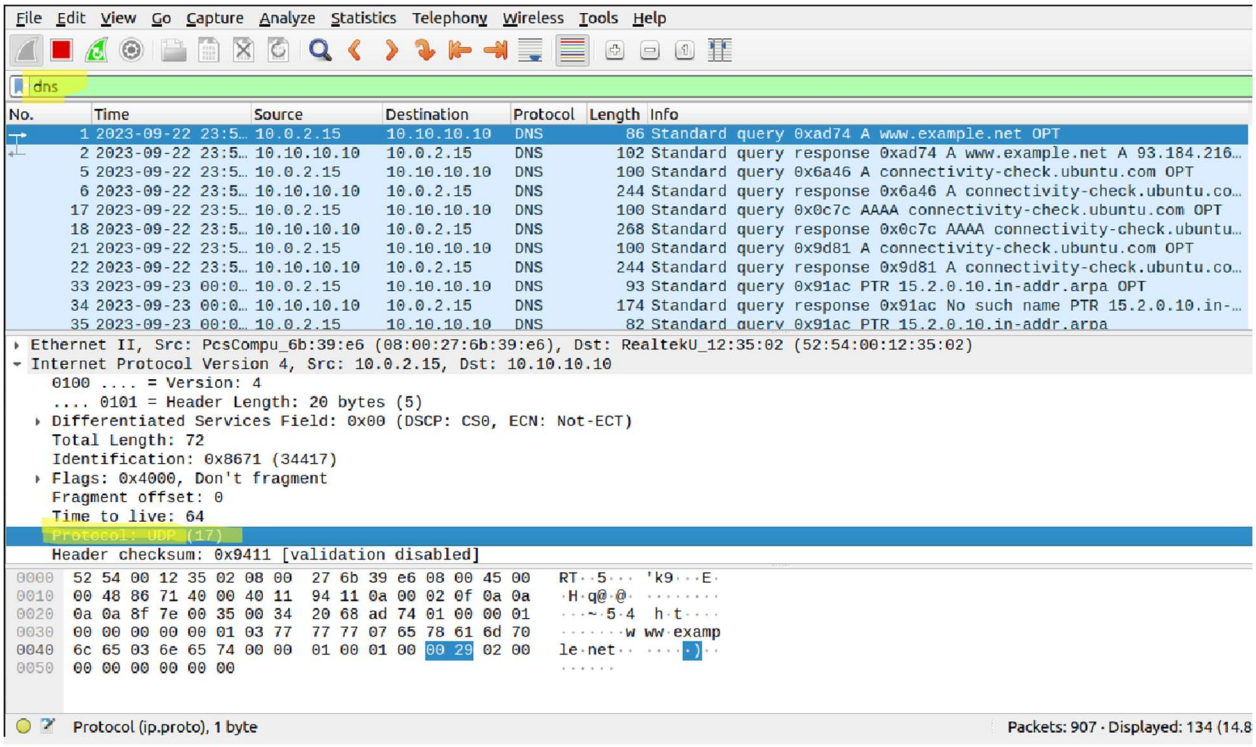
Source Port: 36734

Destination Port: 53

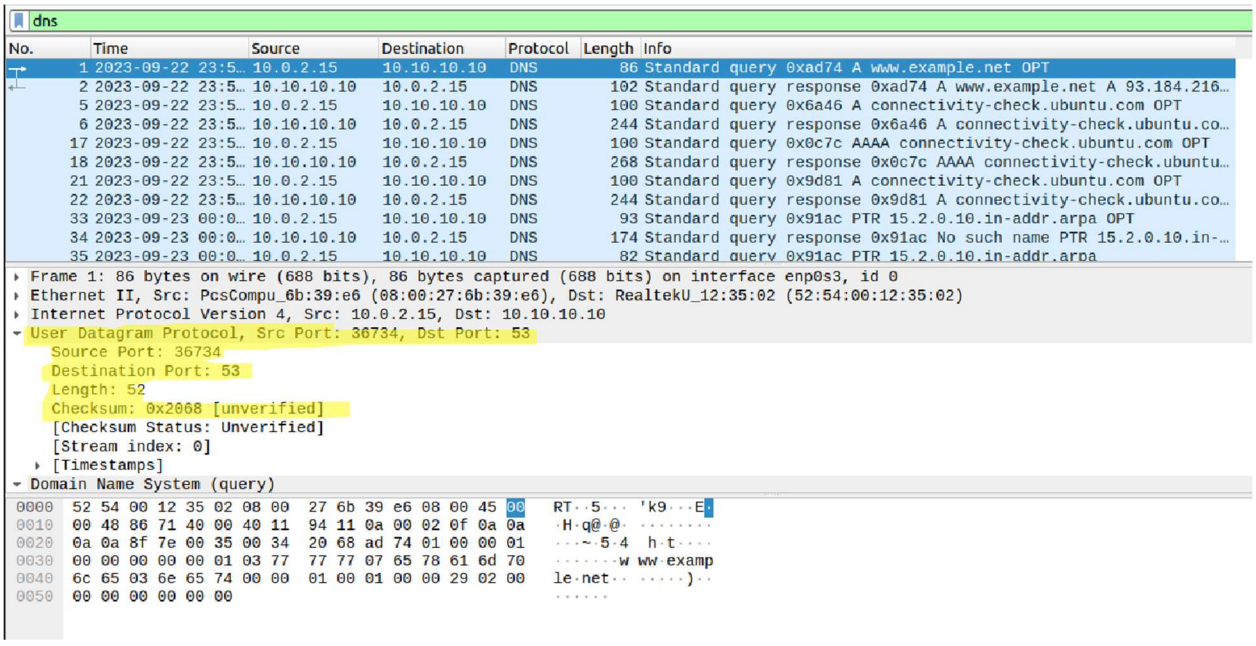
Length: 52

Checksum: 0x2068 [unverified]

This can be verified from the Screenshot 2 and Screenshot 3.



Screenshot 2



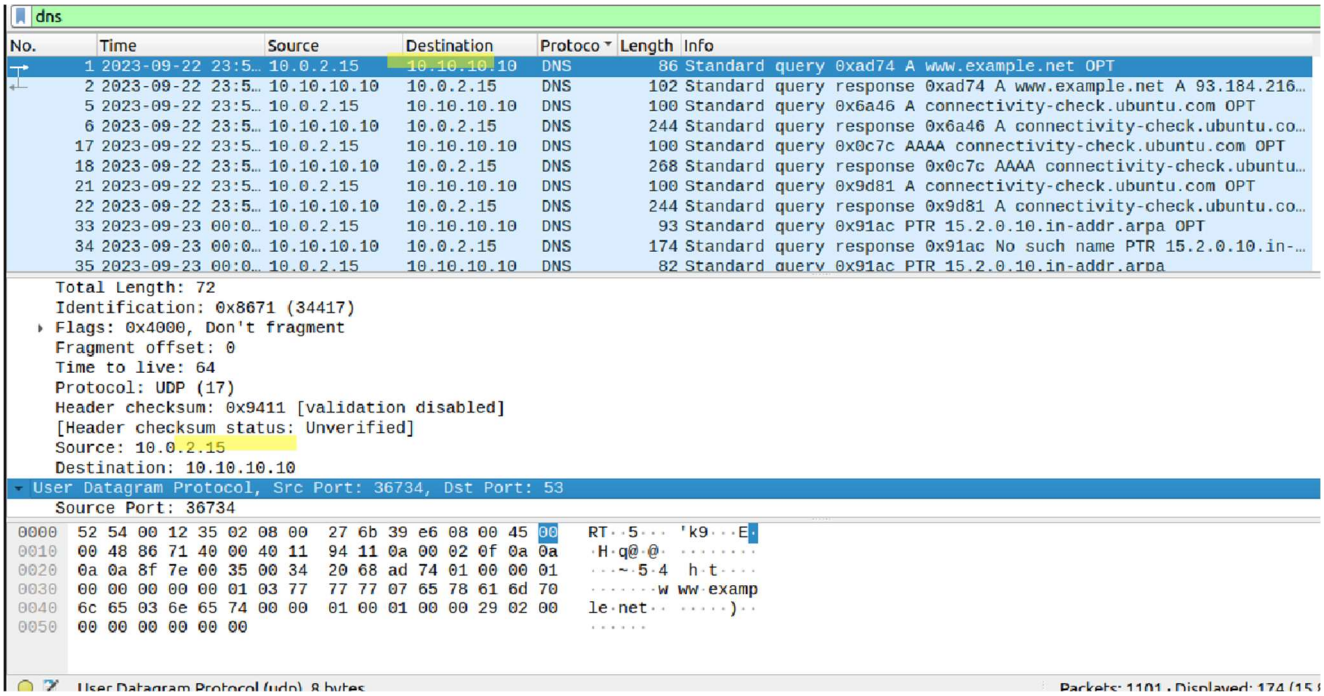
Screenshot 3

Ques 1(c) In the DNS request packet in step b, the destination IP is your local DNS server's IP. What is this value? As said, DNS is serviced by UDP and has no connection setup before sending DNS request. You can confirm this by checking that there is no any packet in Wireshark exchanged between your VM

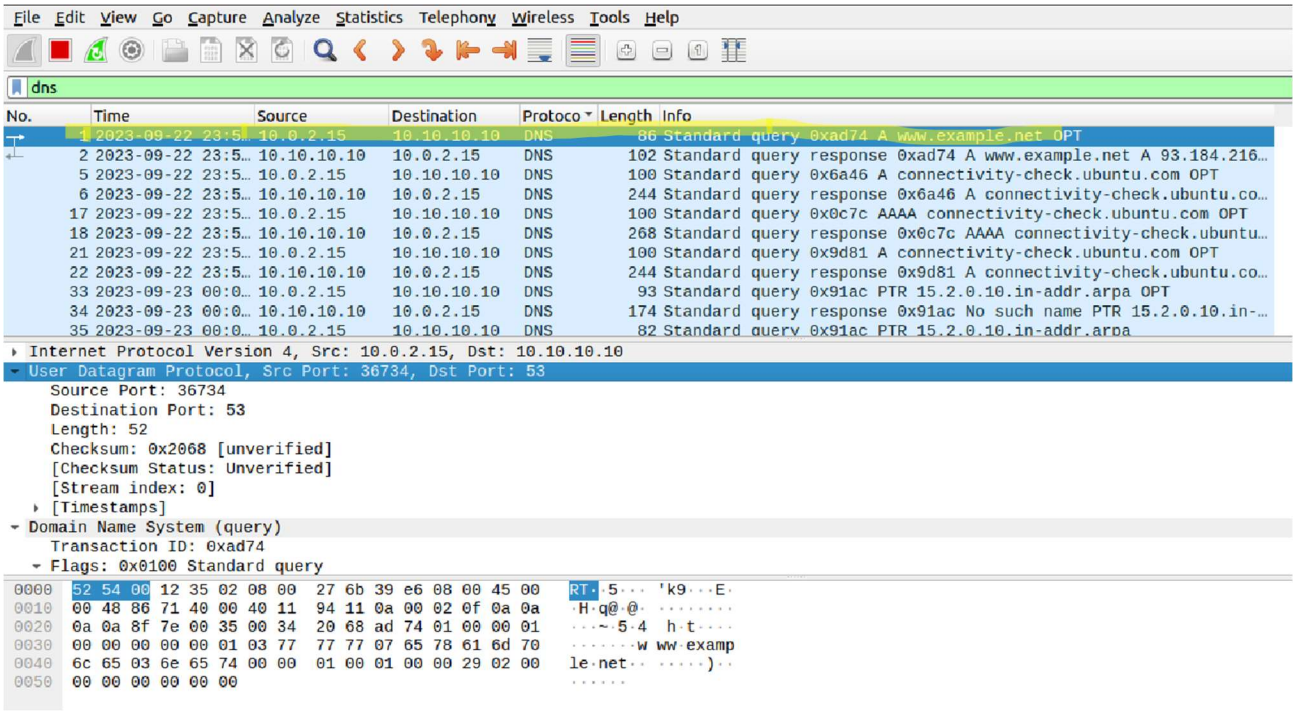
and local DNS server, prior to the DNS request packet (show the screen shot of the window of Wireshark for the list of packets).

Ans 1(c) The destination IP is your local DNS server's IP and its value is 10.10.10.10

Also, we examined that there were no packet(s) capture in Wireshark for any prior packets exchanged between our VM and the local DNS *server by referring to packet # 1 and 2. So in case* there are no preceding packets, that indicates that the DNS requests are sent independently without a prior connection. We can observe from the packet number(#1 and #2) as well from Screenshot 4 and 5.



Screenshot 4



Screenshot 5

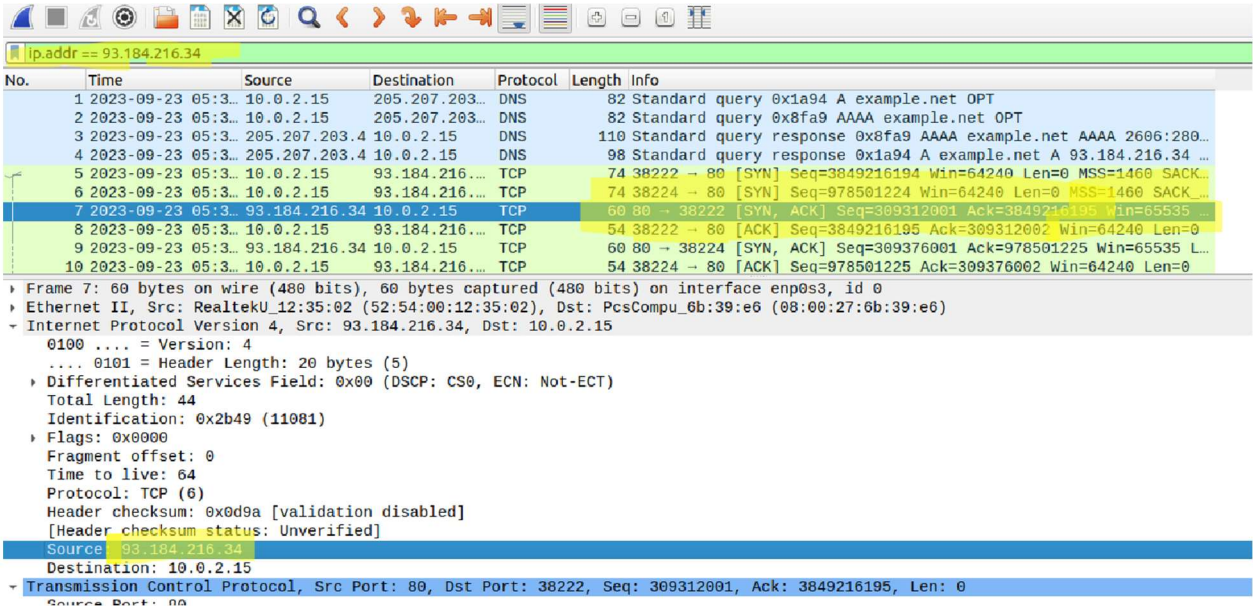
Ques. 2 Run Wireshark and then access www.example.net using Firefox (you might need to clear the browser history). Then stop the Wireshark. Check your list of packets in Wireshark window, filtered by the ip address of www.example.net. You can see that before the HTTP request to www.example.net, there is a connection stage with three packets: SYN packet, SYN-ACK packet and ACK packet. This is to provide the connection setup between your VM and www.example.net. Confirm this. Also, confirm that the transport layer protocol in these packets (check one of them is good enough) is TCP. When the message exchange starts, you can see ACK packet. This is to confirm the receipt of a packet. Find out such a packet. This is to find a packet with flags bit A=1. This provides an evidence that TCP is a reliable protocol. This is different from the UDP protocol. ACK packet might or might not contain the application data. Verify the ACK packet you consider (any of them is ok) to see if it contains application data.

Ans. 2 On accessing **www.example.net** using Firefox, we checked list of packets in Wireshark Window, filtered by ip address of 93.184.216.34 which is of website(**www.example.net**). **We observed that** before the HTTP request to www.example.net, there is a connection stage with three packets: SYN packet, SYN-ACK packet and ACK packet**(can be verified with Screenshot 6 - highlighted part)**. This is to provide the connection setup between your VM and www.example.net. Also, we observed that the transport layer protocol in these packets is TCP. When the message exchange starts, we can also see ACK packet which confirms the receipt of a packet. Packet with flag bit A=1 provides an evidence that the TCP is a reliable protocol**(can be verified with Screenshot 7 - highlighted part)**.

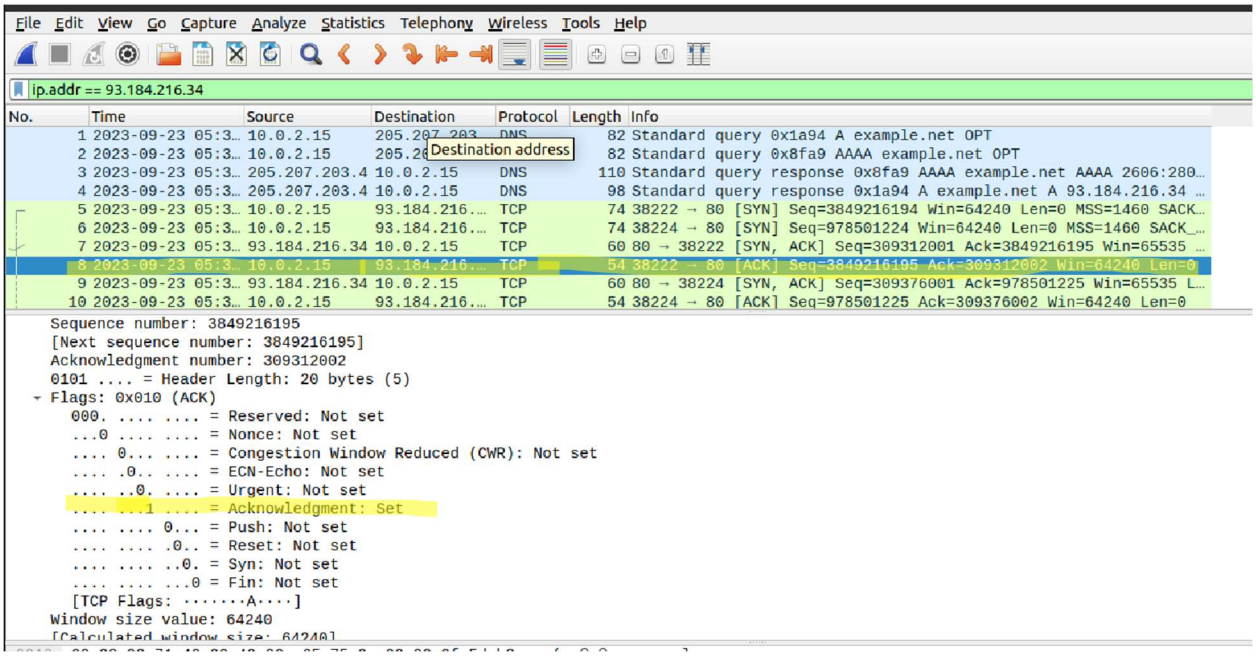
From packet Number – 8 in Screenshot 7, we observe that the provided network packet information represents a TCP acknowledgment (ACK) packet from source IP address 10.0.2.15 to destination IP address 93.184.216.34. This packet is traveling from source port 38222 to destination port 80, which is typically used for HTTP traffic. The main point is that the ACK packet itself does not contain application

data. Instead, it is acknowledging the receipt of data from a previous TCP segment. In our case, the ACK packet is acknowledging the receipt of data with the sequence number 309312002. The length of this ACK packet is 54 bytes, but it does not carry any application data.

ACK packets in TCP are used to confirm the successful reception of data and to manage the flow of data between the sender and receiver. They do not typically contain application data; that data is usually carried in separate data packets.



Screenshot 6



Screenshot 7

Ques. 3. Run Wireshark and access www.example.net and then close your webpage and stop your Wireshark. Answer the following questions.

a. **Find out the first packet from your VM to www.example.net (you should know the ip address of www.examplenet now). This should be the SYN-packet (i.e., the first packet of the 3- way handshake protocol). What is source port # and destination port #? Confirm that they are in the TCP header in the Wireshark packet window. What is source IP and destination IP? Confirm that they are in the ip header in the Wireshark packet window.**

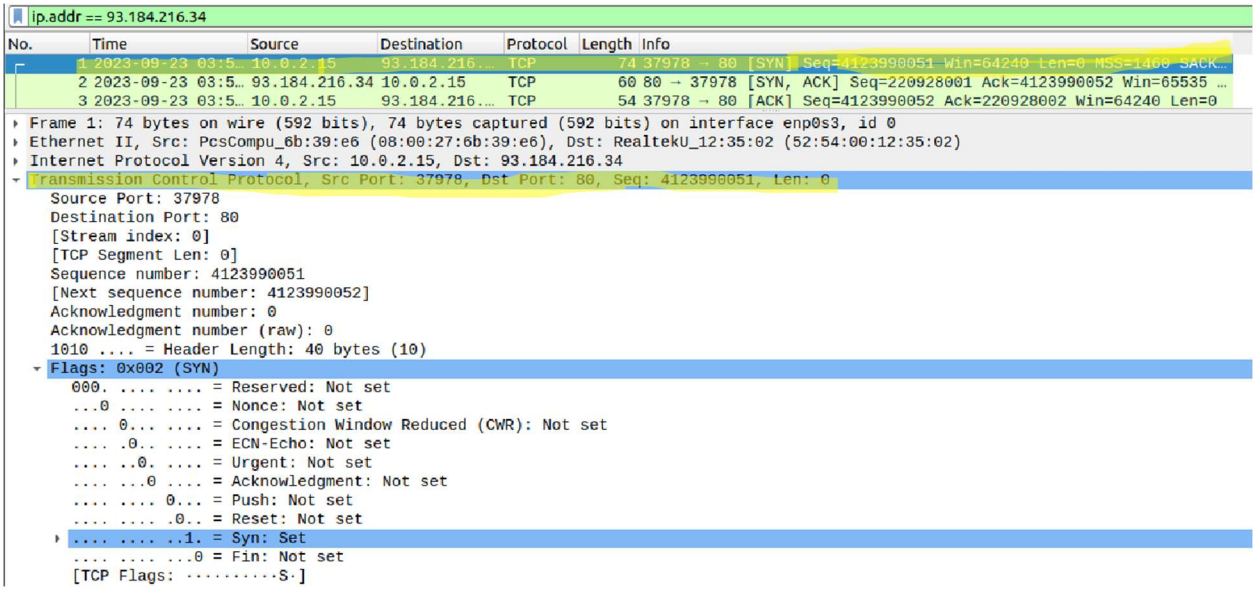
Ans. 3 (a).

The packet with No. 1 in below screenshot is a first packet from VM(Firefox) to www.example.net and the IP Address of www.example.net is 93.184.216.34 and

source port # = 37978

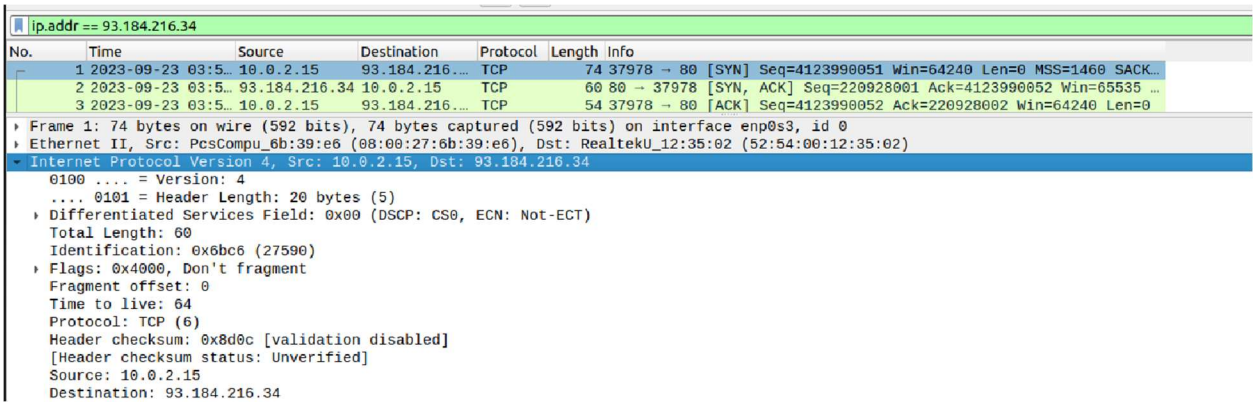
destination port # = 80

This can also be confirmed from the screenshot 8 that they are in the TCP header in the Wireshark packet window



Screenshot 8

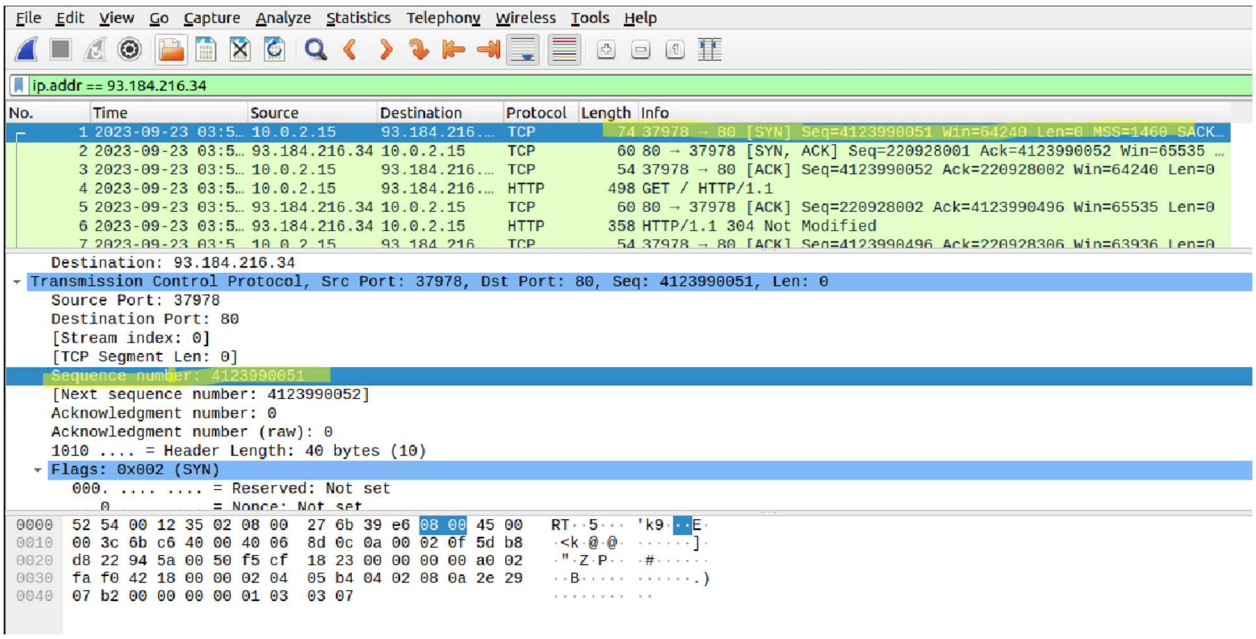
From the screenshot 9, it can be confirmed that the source IP = 10.0.2.15 and destination IP = 93.184.216.34 are in the IP header in the Wireshark packet window.



Screenshot 9

b. Look at SYN-packet. What is the sequence #? It is a random number. Confirm this.

Ans 3 (b). Sequence # is 4123990051 and can be verified from the screenshot 10:-

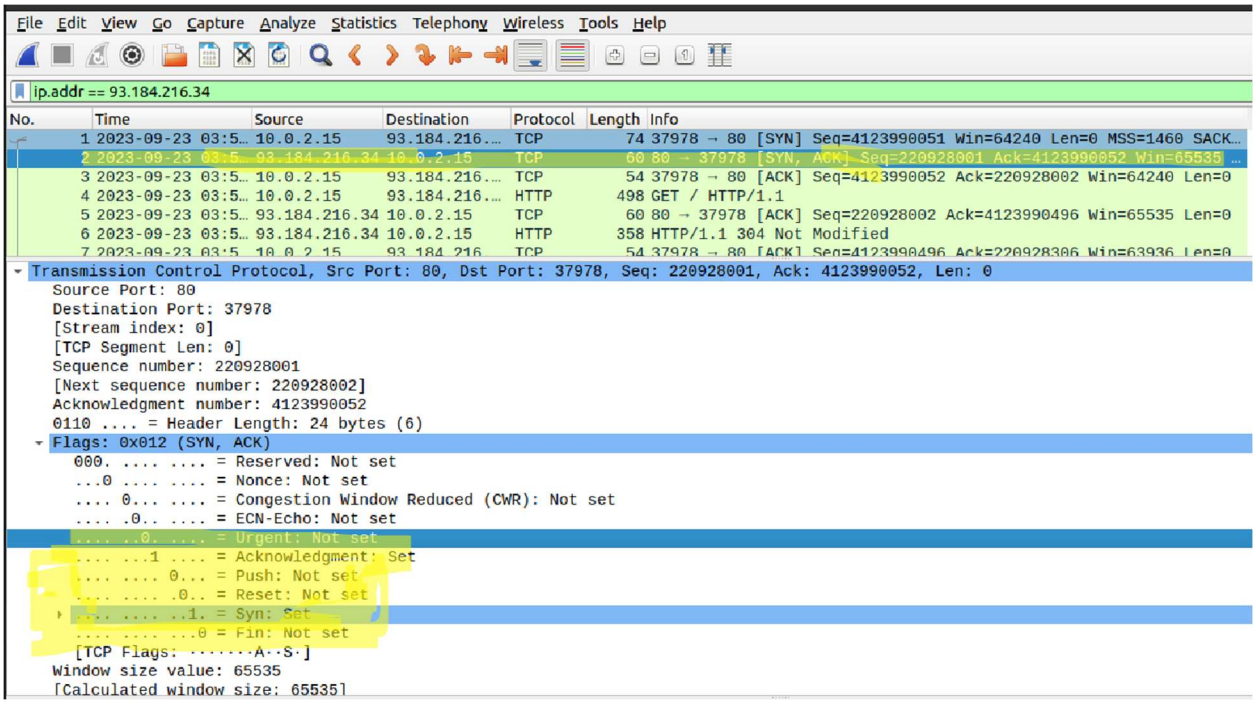


Screenshot 10

c. Find out in the TCP header the flag bits U|A|P|R|S|F in the SYN-ACK packet.

Ans 3 (c). In the TCP header, the corresponding flag bits for U | A | P | R | S | F in SYN – ACK packet are 0 | 1 | 0 | 0 | 1 | 0 respectively (Refer Screenshot 11).

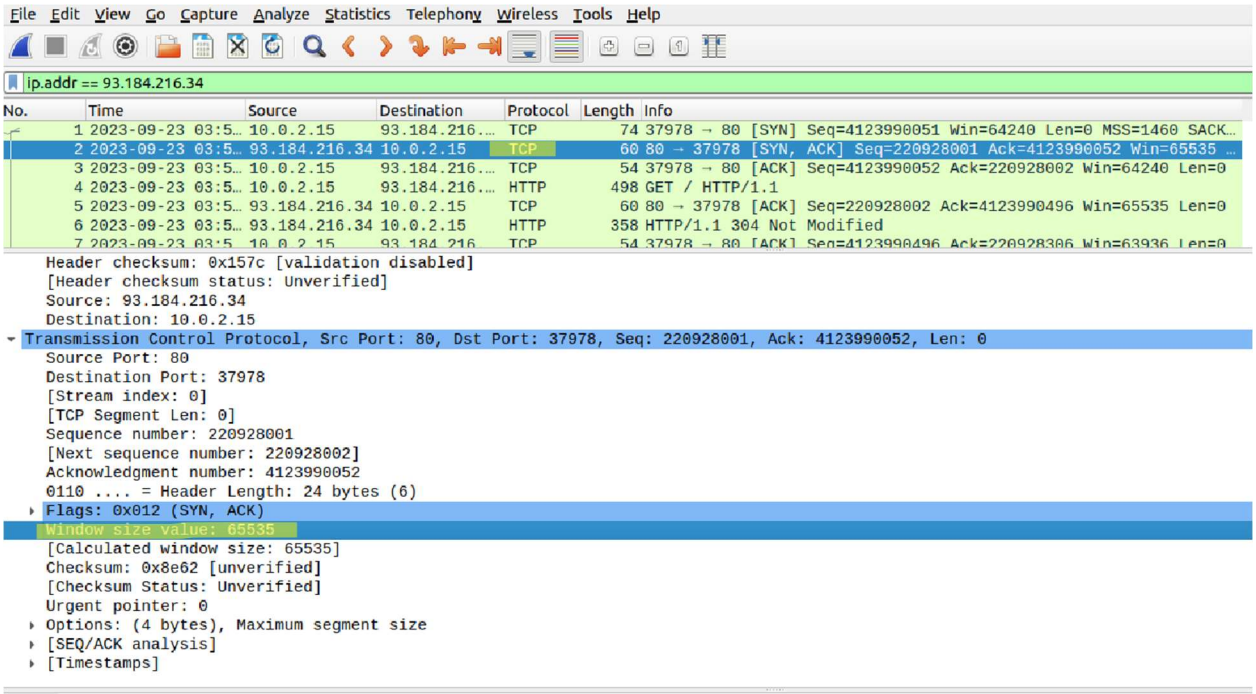
Only A and S – flag bits are set.



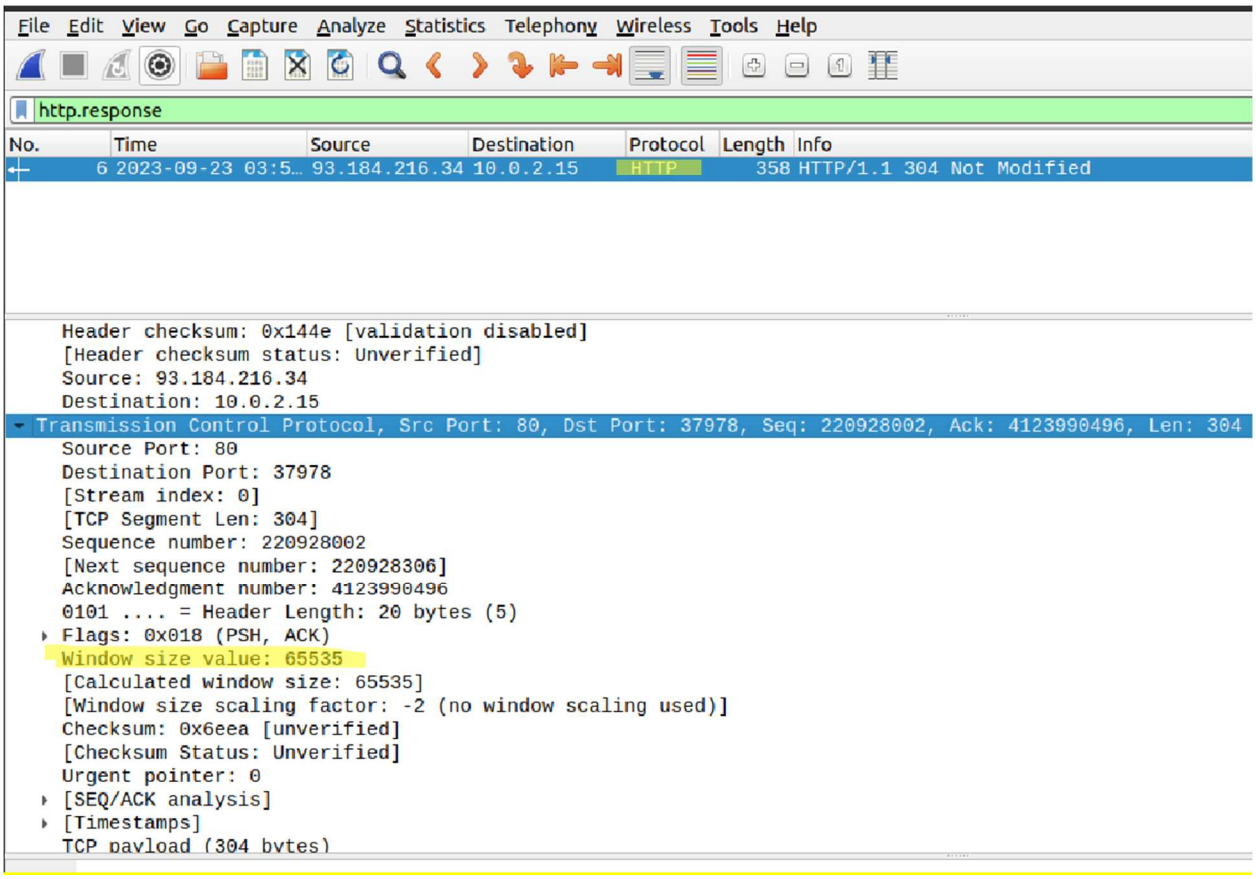
Screenshot 11

d. The receive window field is to tell its partner the current receive-buffer size it has. Find out the window size of SYN-ACK packet and that of http response packet. Are they equal?

And 3 (d). The Window size of **SYN-ACK packet and http response packet** are same which is 65535 and it can be verified from Screenshot 12 and 13.



Screenshot 12



Screenshot 13

e. Find out the sequence # of http request packet and its payload size (the segment len is the payload size). The next sequence # is the sum of these two numbers. Verify that this is indeed the sequence # of the next packet sent by your VM.

Ans 3 (e).

Refer Screenshot 14 –

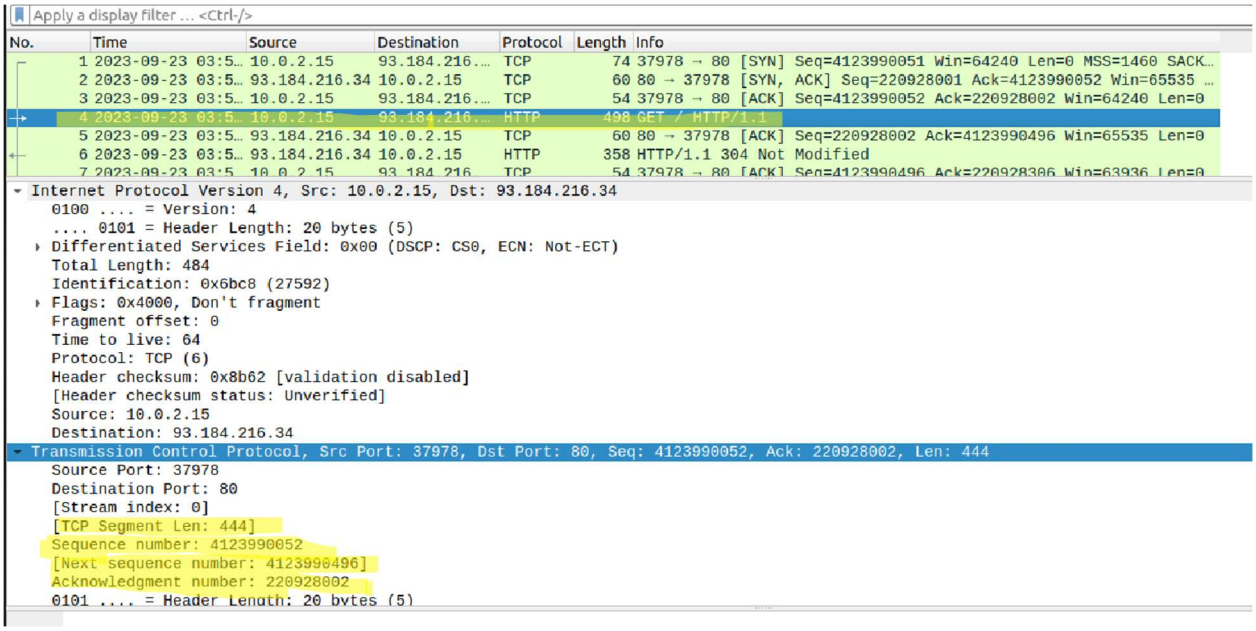
HTTP Request Packet (Frame 4):

Sequence number: 4123990052

Payload size (segment length): 444 bytes

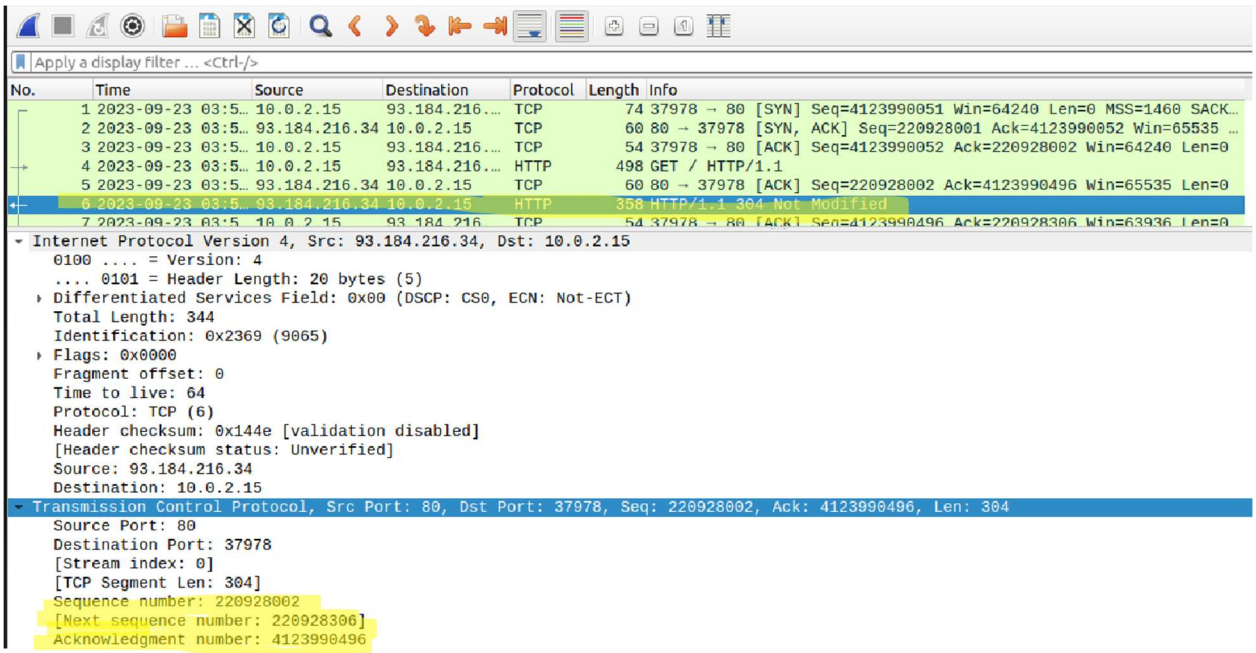
Therefore, Next Sequence Number = Current Sequence Number + Payload Size

Next Sequence Number = 4123990052 + 444 = 4123990496



Screenshot 14

Lets check now if the sequence number of the next packet sent by our VM. The provided packet capture shows the following information for the next packet (Frame 6)(Refer Screenshot 15)



Screenshot 15

Frame 6 - Sequence number: 220928002

The sequence number of Frame 6 is indeed 220928002, which matches the calculated next sequence number (4123990496) obtained from Frame 4.

Hence it is confirmed that the sequence number of Frame 6 is consistent with the calculated one based on the sequence number and payload size of the HTTP request packet in Frame 4.

f. Find out the acknowledgement # in http response packet. Is this the same as the next sequence # you calculated above for the request packet? Explain why?

Ans 3 (f). (By referring to the screenshots 14 and 15, and calculation from answer 3(e))

HTTP Response Packet (Frame 6) - Acknowledgment number: 4123990496

We will now compare the acknowledgment number in the HTTP response packet with the next sequence number calculated earlier for the HTTP request packet (Frame 4):

Next Sequence Number (Calculated for HTTP Request Packet, Frame 4): 4123990496

The acknowledgment number in the HTTP response packet (Frame 6) is indeed the same as the next sequence number calculated for the HTTP request packet (Frame 4).

This is mainly due to the reasons - In TCP, the **acknowledgment number in a packet** shows the next expected byte's sequence number. When the sender sends data (like the HTTP request), it increases its sequence number for the next packet. The receiver confirms data receipt by specifying the next expected sequence number. This **consistency in sequence and acknowledgment numbers** ensures efficient and reliable data transfer.

In our above case, the acknowledgment number in the HTTP response packet (Frame 6) matches the next sequence number calculated for the HTTP request packet (Frame 4) because it signifies that the receiver (the server) **acknowledges the receipt of the entire HTTP request packet** and **expects the next packet (if any)** to have a sequence number of 4123990496.

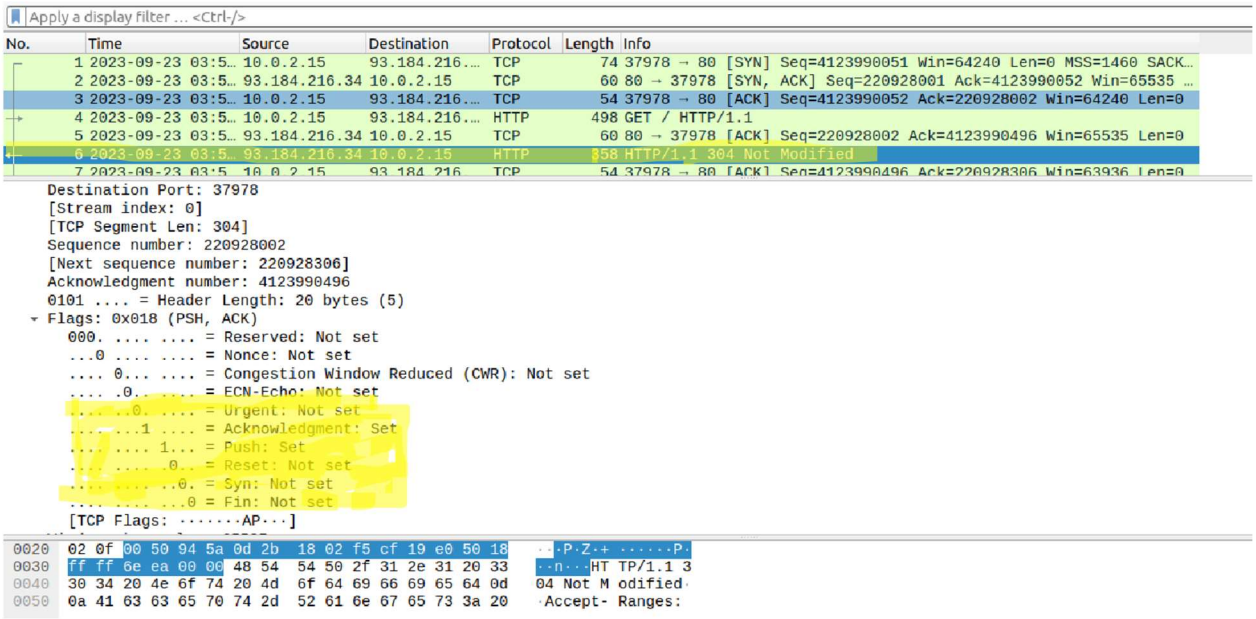
This **consistency in sequence and acknowledgment numbers is a fundamental aspect of TCP's reliability** and **flow control mechanisms,** ensuring that data is transmitted and acknowledged accurately and in the correct order.

g. What is the flags bits U|A|P|R|S|F in the http response packet?

Ans. 3 (g). Refer Screenshot 16 for answer:-

The corresponding flag bits of U|A|P|R|S|F in the http response packet are 0 | 1 | 1 | 0 | 0 | 0

Only A and P flag bits are set.



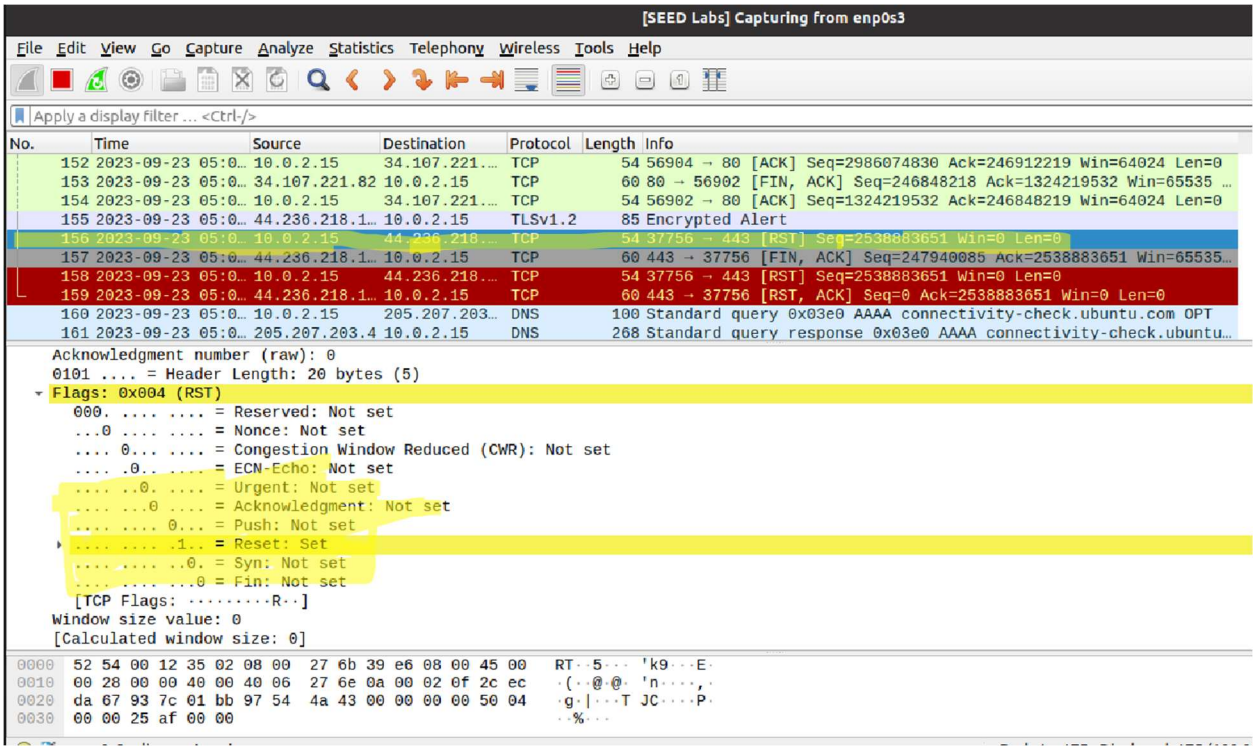
Screenshot 16

h. Find out the packet your VM requests to terminate the TCP connection. This packet will be sent when you close the webpage. What is the flags bit U|A|P|R|S|F in this packet?

Ans. 3 (h). The packet # 156 is the one that VM requests to terminate the TCP connection(refer screenshot 17).

The corresponding flag bits of U|A|P|R|S|F in the http response packet are 0 | 0 | 0 | 1 | 0 | 0

Only R flag bit is set.



Screenshot 17

References

1. **Week 2 – Class 2 Notes**
2. **Week 2 – Class 2 Instruction Document**