**A logo of a university of windsor

Description automatically generated with low confidence**

|  |
| --- |
| **Course Name** |
| Networking and Data Security (COMP-8677) |

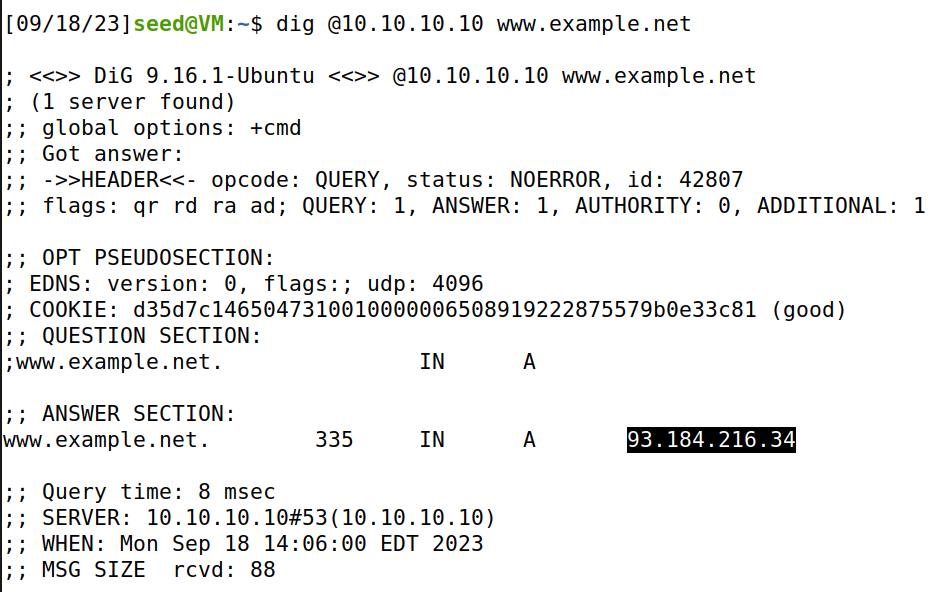
|  |
| --- |
| **Document Type** |
| Lab 2 Work |

|  |
| --- |
| **Professor** |
| Dr. Shaoquan Jiang |

|  |
| --- |
| **Team - Members                               Student ID** |
| Karan Vishavjit                                 110099867 |

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

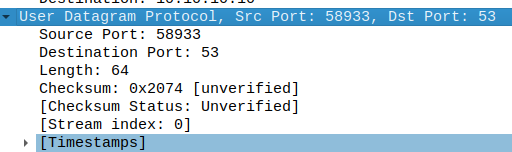
**Answer 1(a)** IP address of [www.example.net](http://www.example.net) is 93.184.216.34

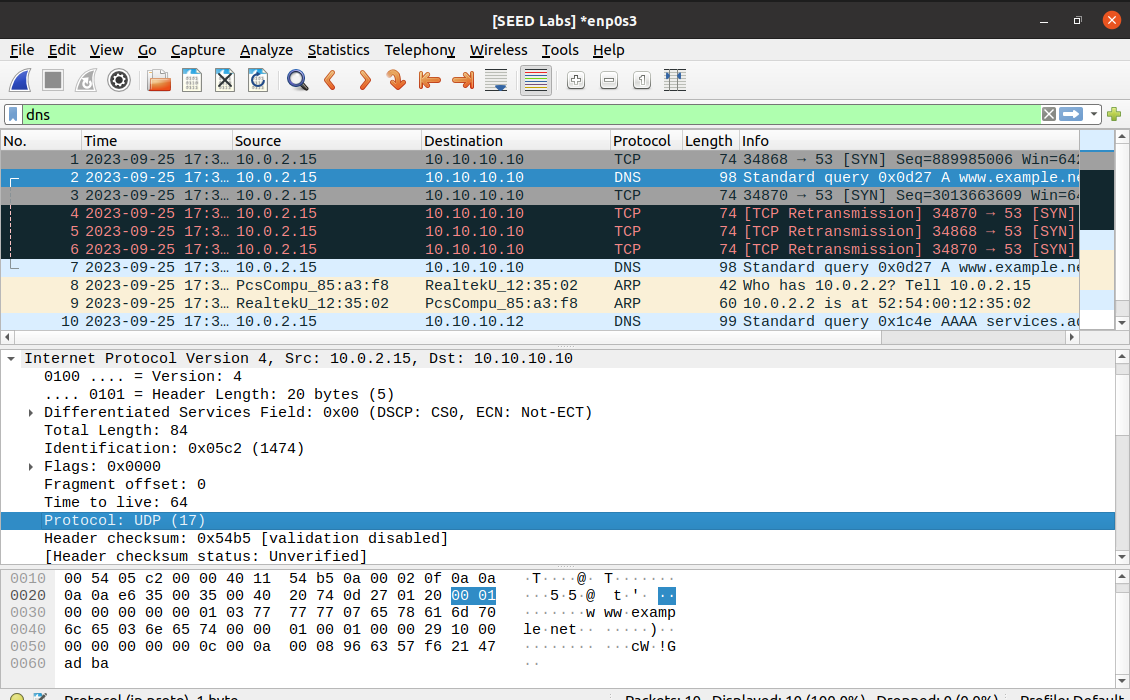


Screenshot 1

**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)?**

Answer 1(b) from the DNS request after dig [www.example.net](http://www.example.net) it can be confirmed that that transport layer protocol is UDP and UDP details are as below:



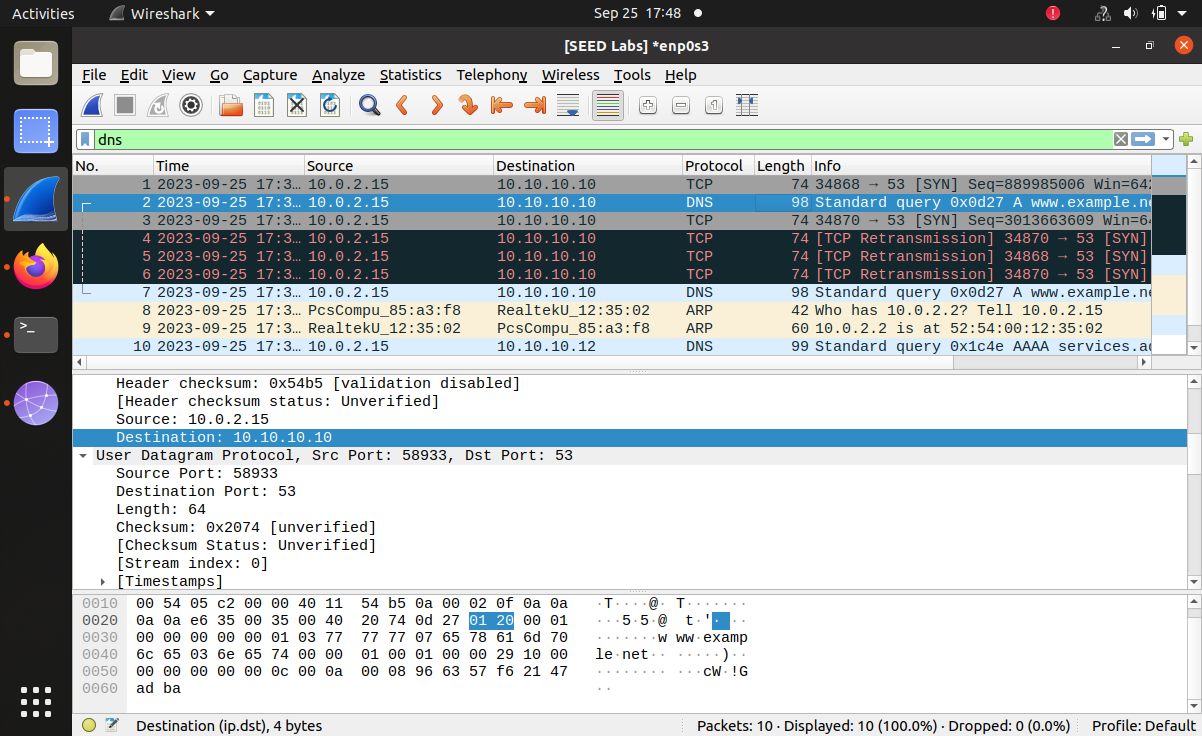


Screenshot 2

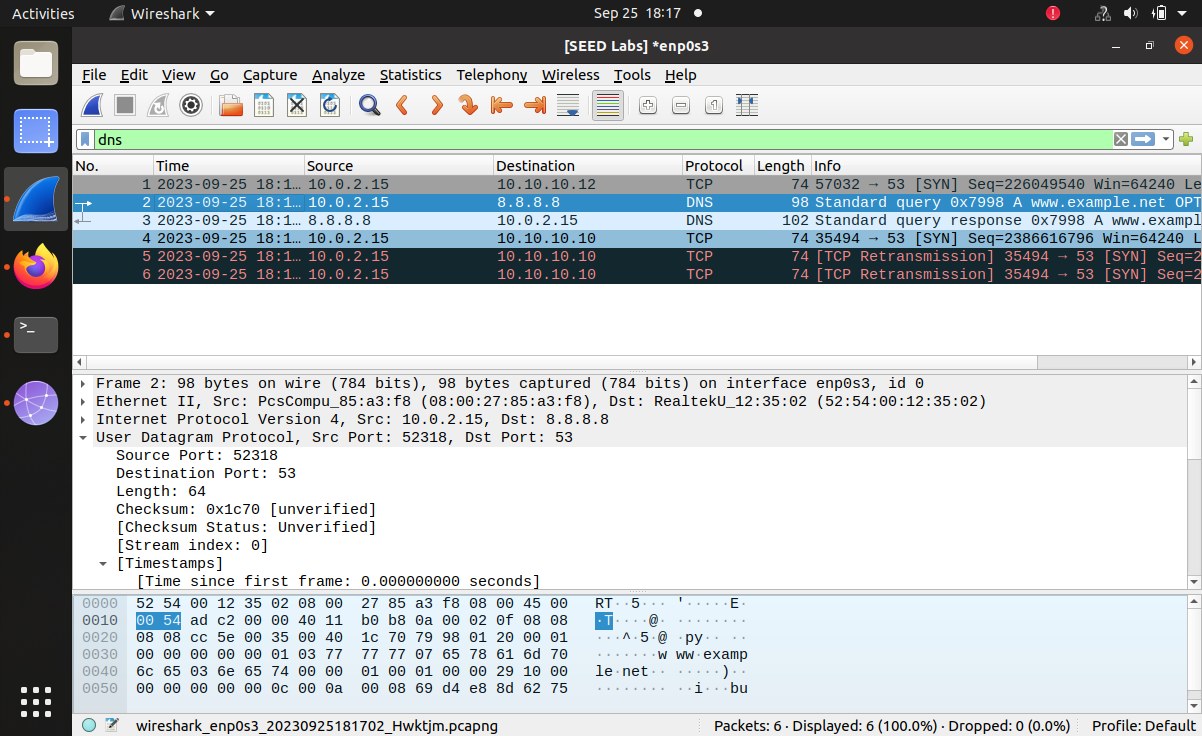
**Ques 1(c) In the DNS request packet in step b, the destination IP is your local DNS server’s IP. What is this value? As said, DNS is serviced by UDP and has no connection setup before sending DNS request. You can confirm this by checking that there is no any packet in Wireshark exchanged between your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

**Answer 1(c)** The value of destination IP is: 10.10.10.10

From the screenshot 3 we can see there is no packet exchange between Virtual Machine and local DNS server that indicate DNS requests are sent without any previous connection.



Screenshot 3





Screenshot 4

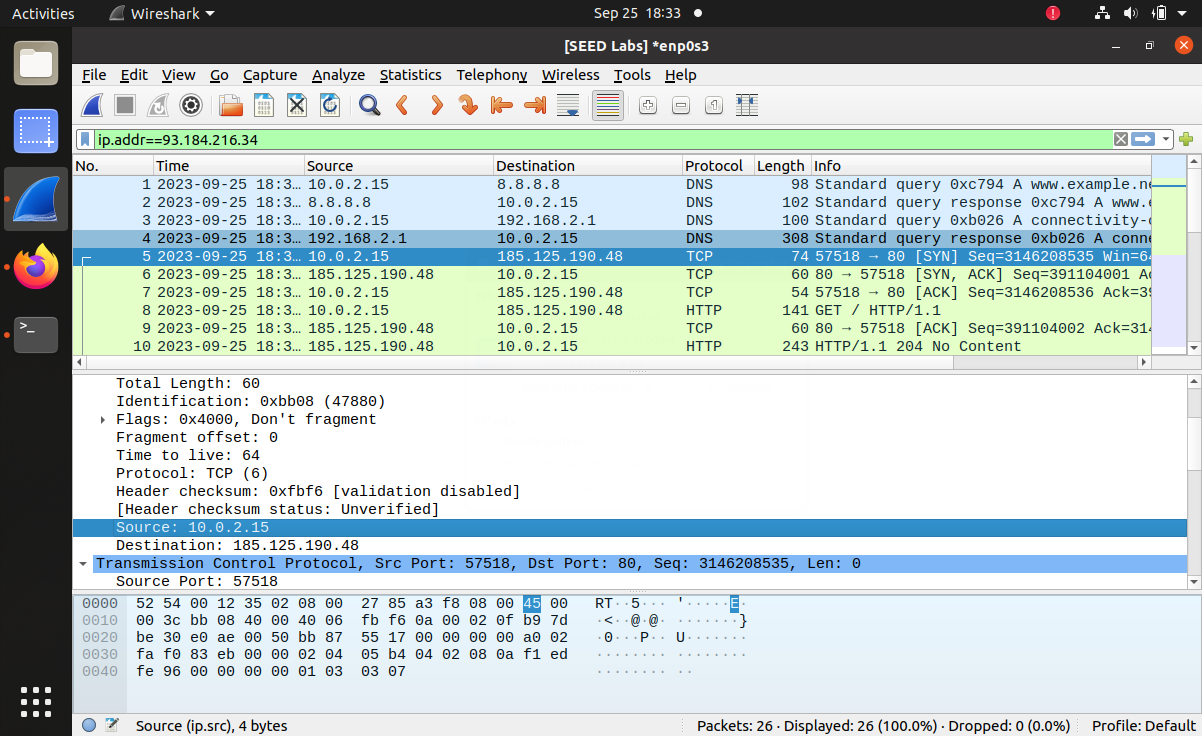
**Ques. 2 Run Wireshark and then access www.example.net using Firefox (you might need to clear the browser history). Then stop the Wireshark. Check your list of packets in Wireshark window, filtered by the ip address of www.example.net. You can see that before the HTTP request to www.example.net, there 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.**

**Answer. 2** When I accessed **www.example.net** using Firefox, and verified list of packets in Wire shark, filtered by IP address of 93.184.216.34 which is IP address of ([**www.example.**net](http://www.example.net)). It can be observed thatbefore the HTTP request is sent to www.example.net, a connection is built with three packets: [SYN], [SYN, ACK] and [ACK] packetscan be verified form screenshot 5. This creates a connection setup between your Virtual Machine and [www.example.net](http://www.example.net). Also, the protocol of this connection is TCP. [ACK] packet confirms the start of message exchange. Packet with acknowledgement set A=1 confirms that TCP is a reliable protocol can be seen in screenshot 6.

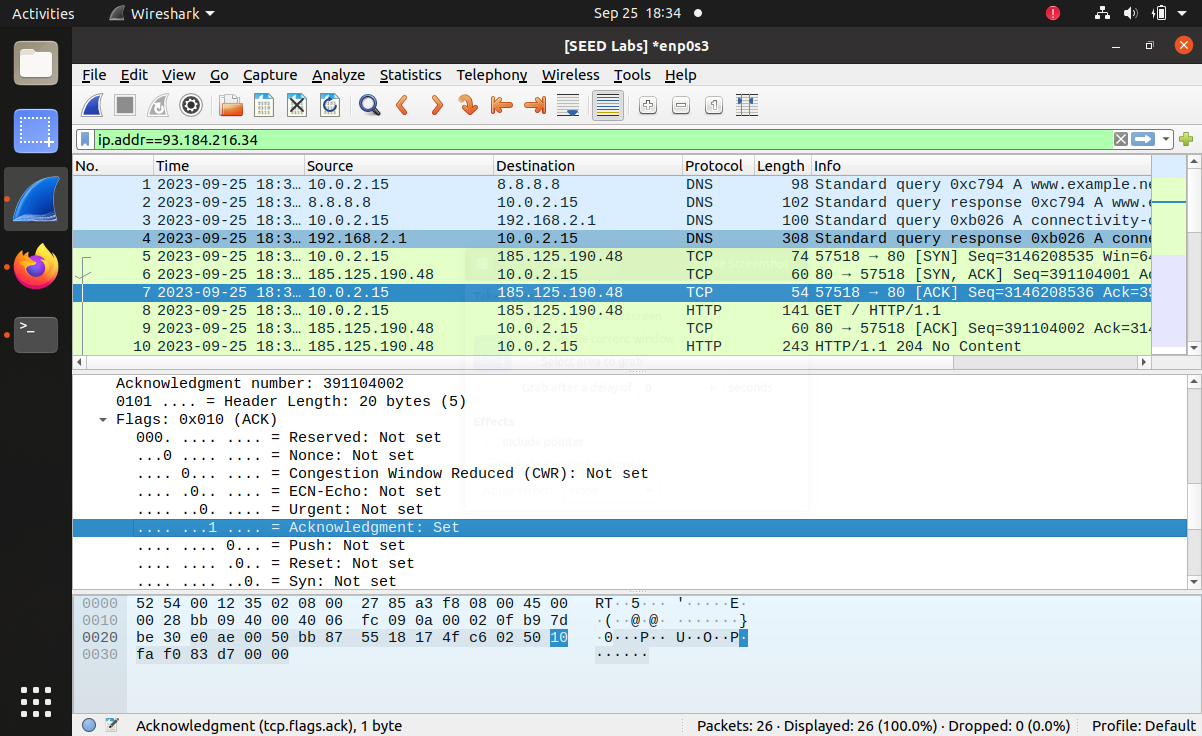


Packet 7 in screenshot 6, provides network information of the packet a TCP acknowledgment [ACK] packet from source IP address 10.0.2.15 to destination IP address 185.125.190.48. This packet is traveling from source port 57518 to destination port 80, which is used for HTTP traffic. The [ACK] packet itself 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 3146208535. The length of this ACK packet is 74 bytes, but there is no application data in it.

[ACK] in TCP is used to confirm the successful reception of data and manage data flow between the sender and receiver. Data is carried separately and [ACK] typically do not contain any application data.



Screenshot 5



Screenshot 6

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

1. **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.**

**Answer. 3 (a).**



From the first packet in the screenshot 7 from Virtual Machine to [www.example.net](http://www.example.net) and the IP Address of [www.example.net](http://www.example.net) is 93.184.216.34

SRC port number = 37978

DEST port number = 80

It can also be verified that both are in the TCP header in the wire shark packet window.

A screenshot of a computer

Description automatically generated



Screenshot 7

The Source IP address is: 10.0.2.15

Destination IP is 93.184.216.34

Both are in the IP header in the Wireshark packet window.

A screenshot of a computer

Description automatically generated

Screenshot 8

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

**Answer 3 (b).** Sequence number is 4123990051 and can be verified from the screenshot 9

A screenshot of a computer

Description automatically generated



Screenshot 9

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

**Answer 3 (c).** U | A | P | R | S | F in [SYN – ACK]

0 | 1 | 0 | 0 | 1 | 0.

Only (A and S) flag bits are set to one.

A screenshot of a computer

Description automatically generated

Screenshot 10

1. **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 is same for [**SYN-ACK] and HTTP** response packet = 65535.

A screenshot of a computer

Description automatically generated

Screenshot 11

A screenshot of a computer

Description automatically generated

Screenshot 12

1. **Find out the sequence # of http request packet and its payload size (the segment length 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.**

**Answer 3 (e).**

**Refer Screenshot 14 –**

HTTP Request Packet number 4:

Sequence #: 4123990052

Payload size (segment length): 444 bytes

Next Sequence Number = Current Sequence # + Payload Size

Next Sequence Number = 4123990052 + 444 = 4123990496

A screenshot of a computer program

Description automatically generated

Screenshot 13

A screenshot of a computer

Description automatically generated

Screenshot 14

Packet number 6 - Sequence number: 220928002 and Acknowledgement number is 4123990496

Which matches the calculated next sequence number (4123990496) obtained from packet number 4  
  
Hence it is verified that the sequence # of packet 6 is consistent with the calculation based on the sequence # and payload size of the HTTP request packet in packet 4.

1. **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?**

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

HTTP Response (packet # 6) - Acknowledgment number: 4123990496

We can compare the Ack number in the HTTP response packet with the next sequence number calculated earlier for the HTTP request packet (packet # 4)

Next Sequence # (Calculated for HTTP Request Packet, packet # 4): 4123990496

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

In TCP, the Ack numbershows the next expected byte's sequence number. When the sender sends data, it increases its sequence number for the next packet which is current sequence number + payload size. The receiver confirms data receipt by specifying the next expected sequence number. This sequence number acknowledgmentensures efficient and reliable data transfer.

In our above case, the acknowledgment number in the HTTP response packet (packet # 6) matches the next sequence number calculated for the HTTP request packet (packet # 4) because it signifies that 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 an aspect of TCP's reliability and flow control mechanisms**,** ensuring that data is transmitted and acknowledged accurately and in the correct order.

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

**Answer. 3 (g).**  Flag bits of U|A|P|R|S|F in the http response packet

0 | 1 | 1 | 0 | 0 | 0

Only A and P flag bits are set to ‘1’.

A screenshot of a computer

Description automatically generated

Screenshot 15

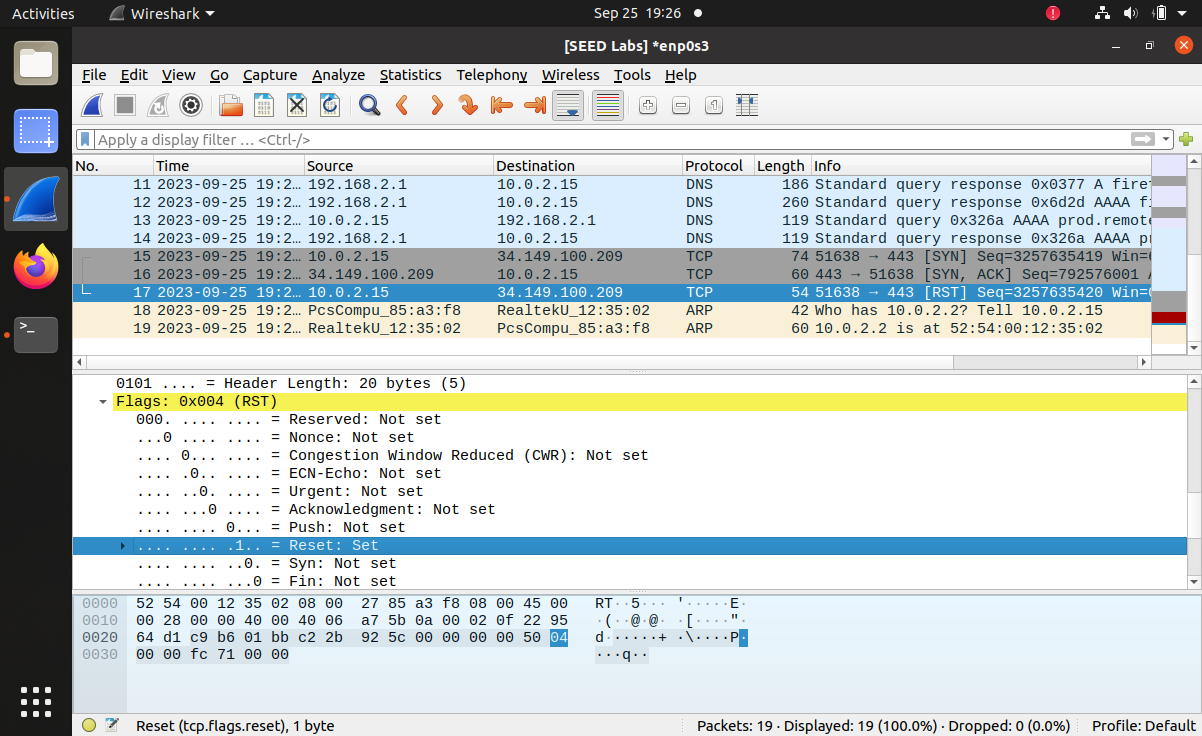
1. **Find out the packet your VM requests to terminate the TCP connection. This packet will be sent when you close the webpage. What is the flags bit U|A|P|R|S|F in this packet?**

**Answer. 3 (h).** The packet # 17 terminates the TCP connection

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

0 | 0 | 0 | 1 | 0 | 0

Only R flag bit is set.

 Screenshot 16

**References**

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