Lab 3: Packet Capture

**Introduction**

In this lab, you will use a “packet sniffer” called Wireshark to capture and analyze TCP packets generated between the PC browser and a web server, such as **matrix.senecacollege.ca**. When the application layer of the TCP/IP protocol stack creates an HTTP message, that message is “encapsulated” by a transport layer header. The header identifies the protocol TCP which is used to make a reliable connection to a web server. TCP uses a three-way handshake to establish a connection and a three-way handshake to take down a connection between the two hosts. The Internet layer adds a header indicating the logical IP address, but is also responsible to retrieve the MAC address which is passed to the Data Link layer for addition into the LAN header. You will see how the Internet layer uses a protocol called ARP (Address Resolution Protocol) to find the MAC or Ethernet address of the next link. Lastly, you will see the message syntax and sequence of the HTTP protocol.

**Objective:**

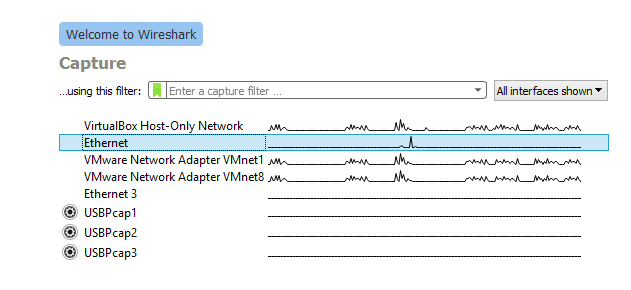
1. Demonstrate basic packet capturing with Wireshark
2. Examining the TCP handshake used to set and take down a reliable connection
3. Examine how the Internet layer uses ARP

**Instructions:**

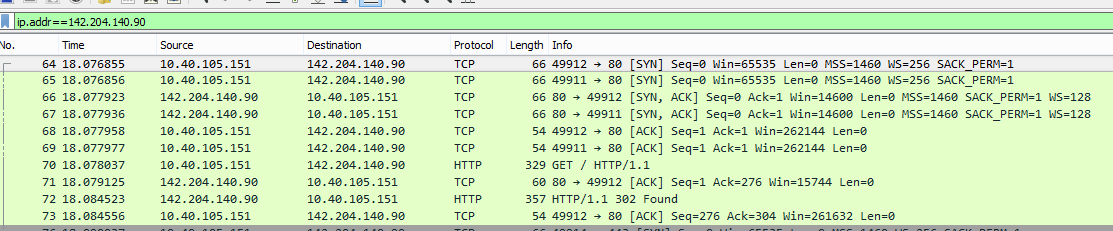
1. Use the MyApps folder to locate Wireshark
2. Click the Launch button to open Wireshark
3. Use **ipconfig /all** at a command prompt to get the IP and physical addresses of the local machine.

|  |  |
| --- | --- |
| Physical Address of host | DC-21-5C-00-E1-F0 |
| IP Address of host | 10.241.4.249 |
| IP Address of default gateway | 10.241.0.1 |
| Physical address of default gateway | fe80::4122:f0a7:147b:b221%13 |

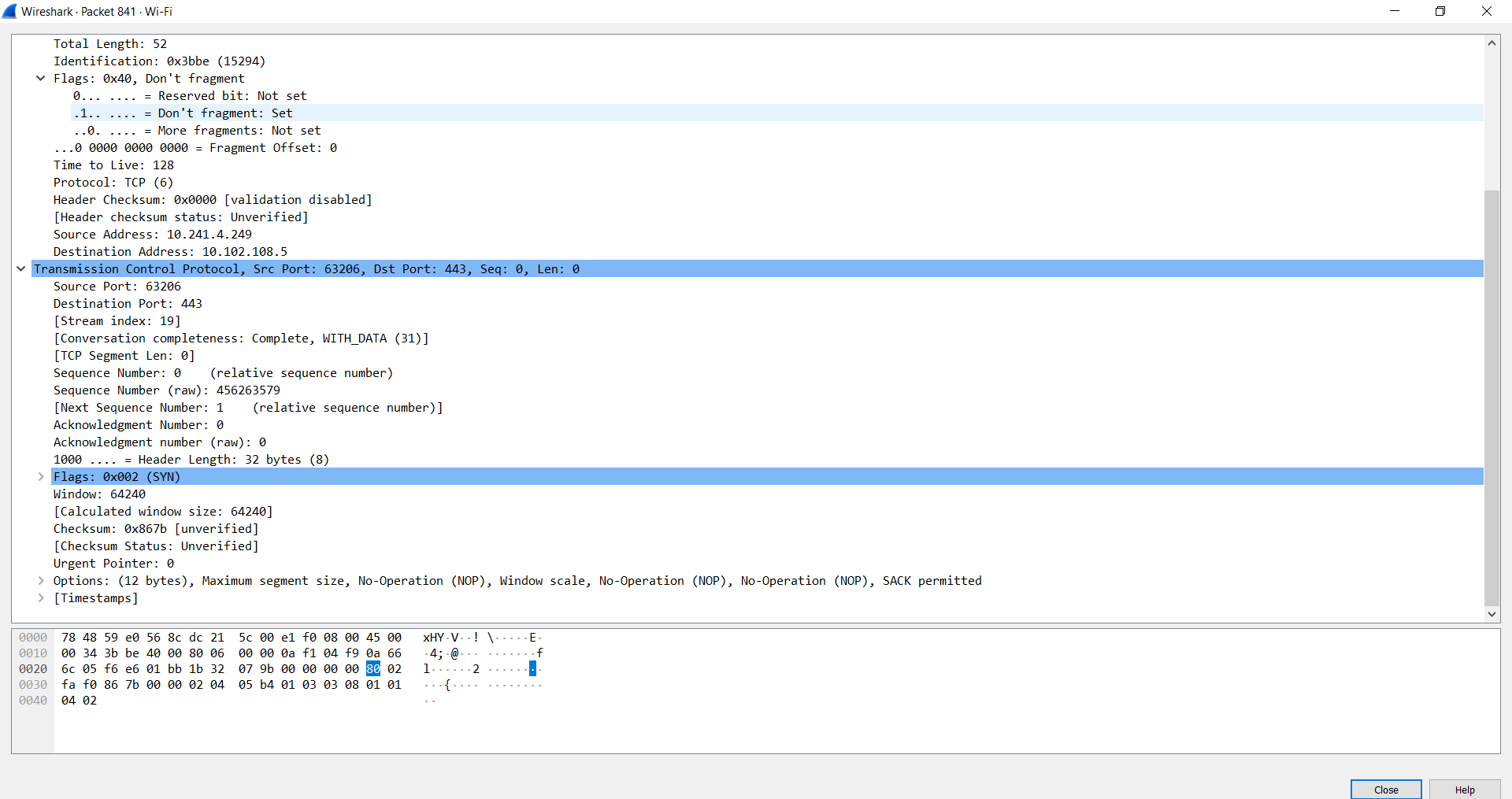
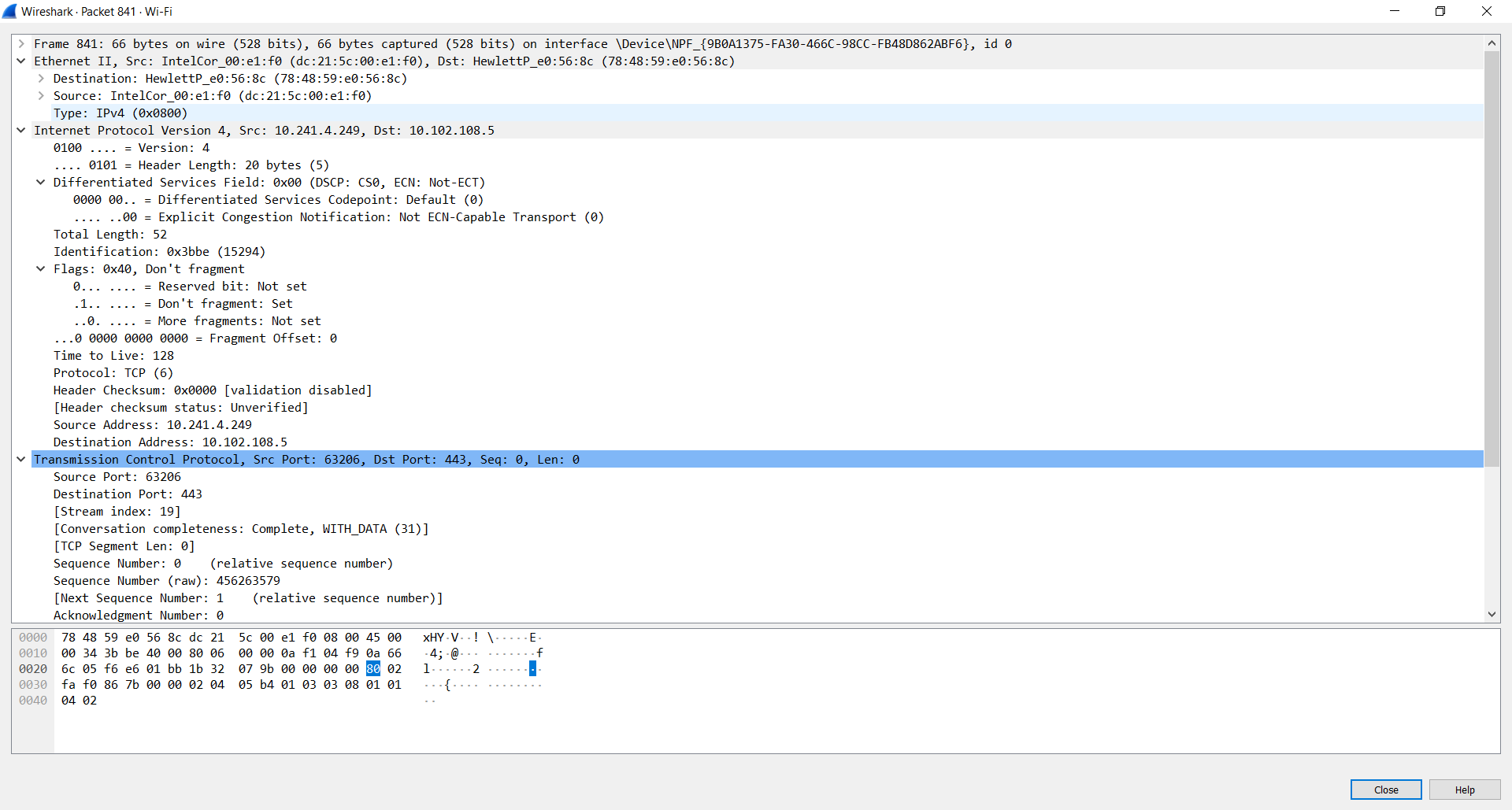
1. Before we capture packets delete the ARP cache. This area of memory keeps a mapping or IP addresses to MAC addresses. We want to delete any previous entry so that the protocol ARP will need to be used in our capture
2. Open a command line windows as administrator and type the following:  
    **netsh interface ip delete arpcache**

**Capturing and Examining TCP Packets  
  
TCP Connection Setup: 3-way Handshake**

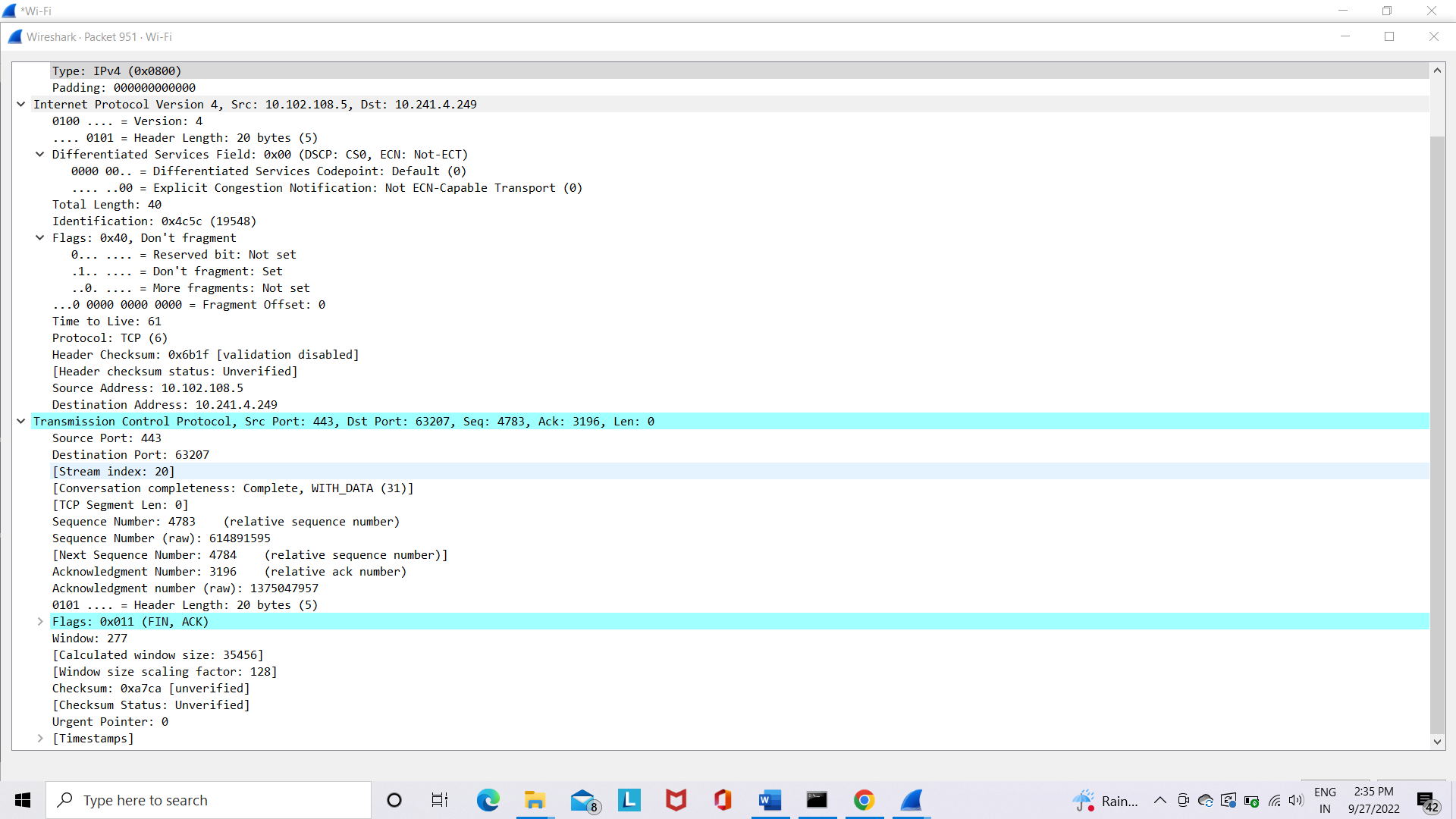
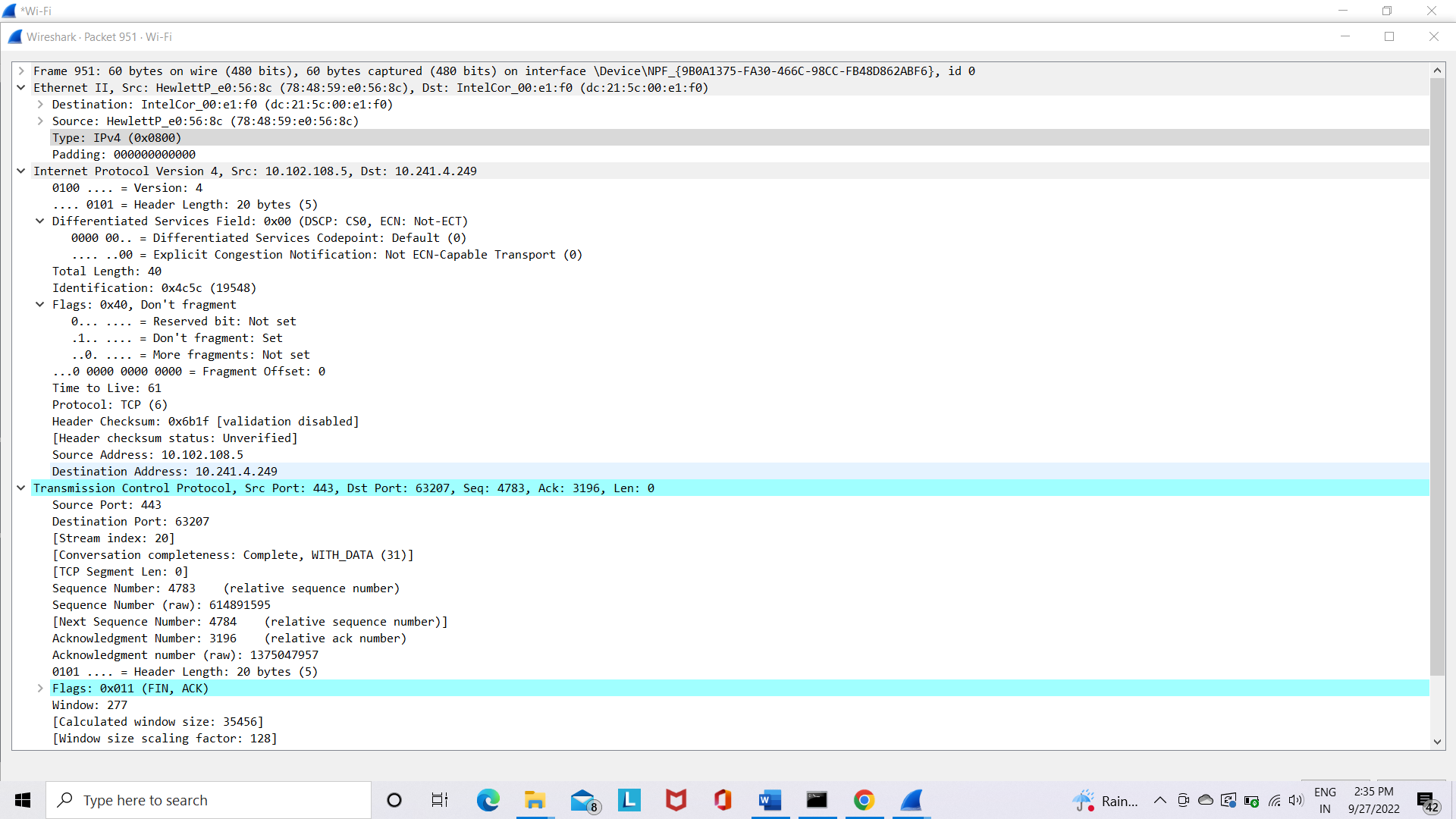
1. **Close all the browser windows** before starting wireshark.
2. Select an Interface to capture called “Ethernet” which shows activity on it. Similar to the screen shot above
3. On wireshark select the interface for packet capturing (ethernet or wifi)
4. On the capture menu click the Start button
5. Open the browser and navigate to **matrix.senecacollege.ca**
6. When the web page loads, close the client window and wait a couple of seconds
7. Return to Wireshark and **Stop** capture.
8. Save the capture as a file called **learnname\_L3\_capture**. This is important, if you need to return to the original file after applying display filters.
9. Type in the Display filter text box **ip.addr==142.204.165.128** (if accessing from outside Seneca network. If this does not work ping matrix.senecacollege.ca and get the ip address from the ping reply message) or **ip.addr==10.102.108.5** (from Seneca network).This will show the beginning of your conversation with the matrix server. Your Wireshark window should look like the screen shot below.



1. Notice the first conversation between your host to the server is a [SYN] packet with an info number of 49912 **(yours will be different)**. The latter is a TCP flag which tells the server to open a connection to the host. Notice SEQ=0. Click on the [SYN] packet and open the drop-down arrow on the Transmission Control Protocol in the Details pane in the middle Wireshark window.
2. In the top Wireshark packet list pane, select the second TCP packet, labeled SYN, ACK with the same info number 49912.
3. Observe the packet details in the middle Wireshark packet details pane. Notice that it is an Ethernet II / Internet Protocol Version 4 / Transmission Control Protocol frame.
4. Expand Ethernet II to view Ethernet details. Answer the following questions in the table below:
5. Observe the Destination and Source fields.
6. Expand Internet Protocol Version 4 to view IP details.
7. Observe the Source and Destination IP addresses.
8. Expand Transmission Control Protocol to view TCP details.
9. Observe the Source and Destination ports.



1. Observe the Sequence number. Notice that it is 0 (relative sequence number). To see the actual sequence number, select Sequence number to highlight the sequence number in the bottom Wireshark bytes pane.
2. Observe the Acknowledgement number. Notice that it is 1 (relative ack number). To see the actual acknowledgement number, select Acknowledgement number to highlight the acknowledgement number in the bottom pane. Notice that the actual acknowledgement number is one greater than the sequence number in the previous segment.
3. Expand Flags to view flag details.
4. Observe the flag settings. Notice that SYN and ACK flags are set, indicating the second segment in the TCP three-way handshake.



1. Complete the FIN ACK packet analysis table. **[0.4 Marks]**

|  |  |
| --- | --- |
| FIN ACK Packet Analysis | |
| What is the source MAC address of this packet? (should be the default gateway physical address) | Source: HewlettP\_e0:56:8c (78:48:59:e0:56:8c) |
| What is the destination address of this packet? (should be the host physical address) | Destination: IntelCor\_00:e1:f0 (dc:21:5c:00:e1:f0) |
| What is the source IP address of this packet? (should be the matrix server IP address) | Source Address: 10.102.108.5 |
| What is the destination IP address of this packet? (should be the host IP address) | Destination Address: 10.241.4.249 |
| What is the destination port of this packet? (should be a local dynamic port created for this connection) | Destination Port: 63207 |
| What is the source port of this packet? (should be port 80) | Source Port: 443 |

Client

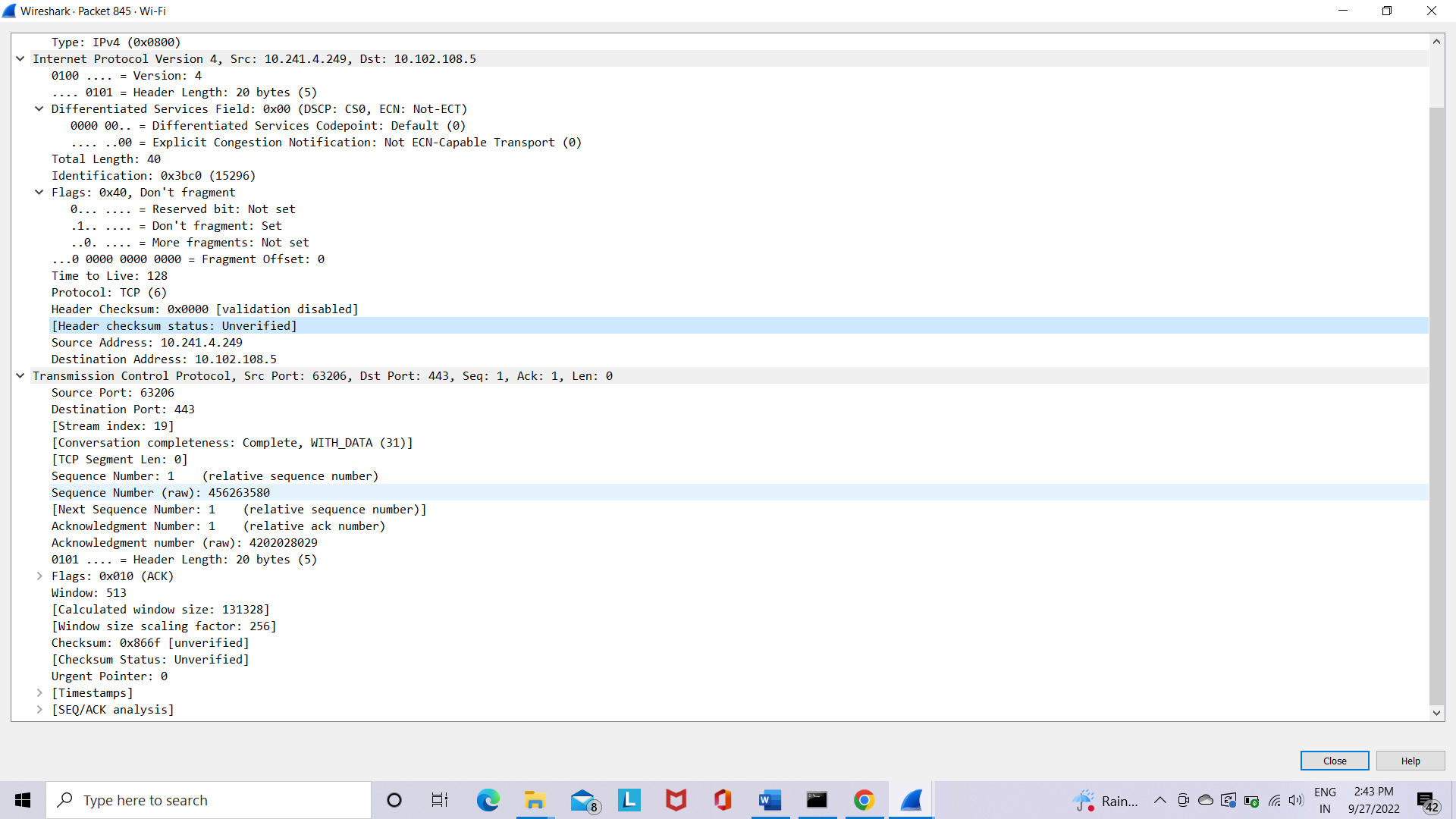
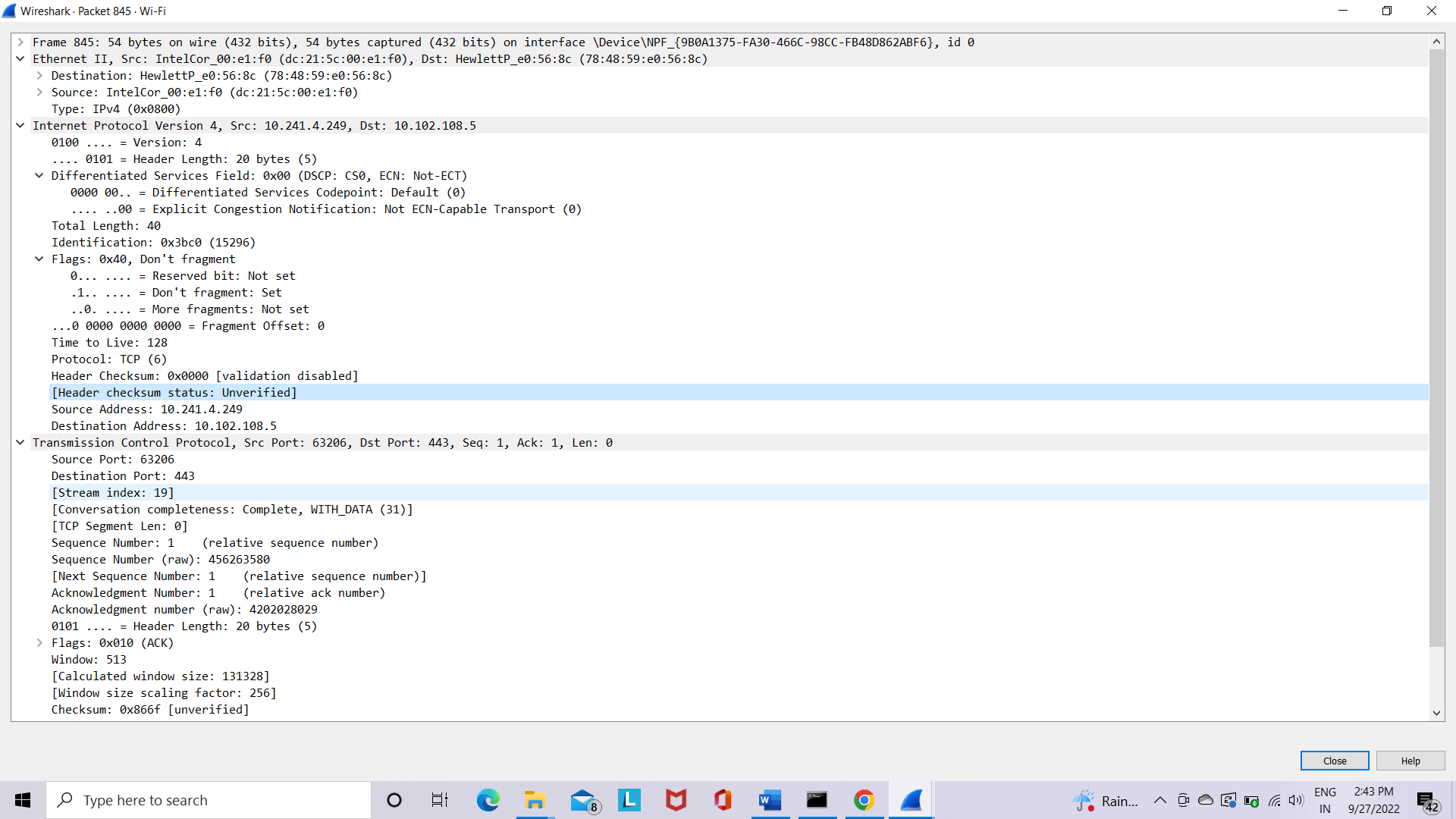
Server

SYN

SYN ACK

ACK

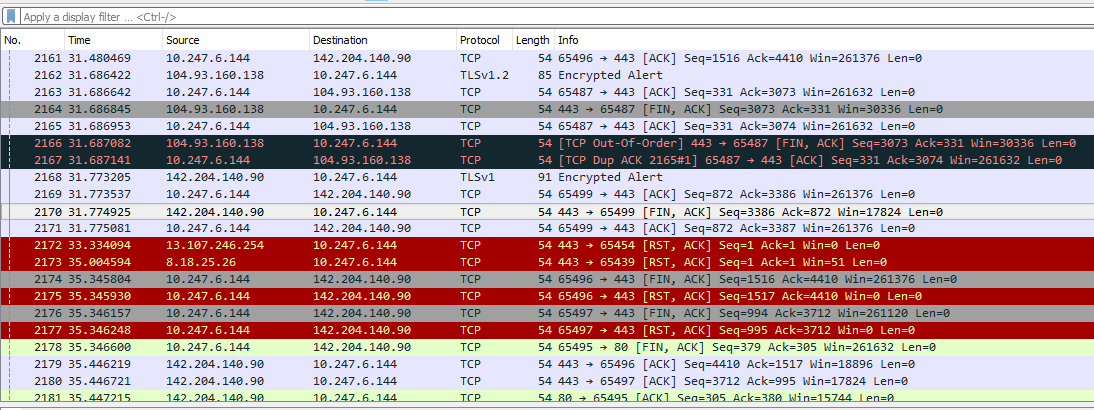
1. The last step of the 3-way handshake is the host sends an acknowledgement to the server’s acknowledgement with the SEQ=1, which tells the server that this packet is an acknowledgement of its previous packet and with the ACK=1, which tells the server the communication channel is now fully open and able to send data. Answer the following questions and complete the table below:
2. Observe the Destination and Source fields.
3. Expand Internet Protocol Version 4 to view IP details.
4. Observe the Source and Destination IP addresses.
5. Expand Transmission Control Protocol to view TCP details.
6. Observe the Source and Destination ports.
7. Observe the Sequence number. Notice that it is 1 (relative sequence number). To see the actual sequence number, select Sequence number to highlight the sequence number in the bottom Wireshark bytes pane.
8. Observe the Acknowledgement number. Notice that it is 1 (relative ack number). To see the actual acknowledgement number, select Acknowledgement number to highlight the acknowledgement number in the bottom pane.
9. Expand Flags to view flag details.
10. Observe the flag settings. Notice that ACK is set, indicating the third segment in the TCP three-way handshake. The client has established a TCP connection with the server.
11. Complete the ACK packet analysis table. **[0.4 Marks]**



|  |  |
| --- | --- |
| ACK Packet Analysis | |
| What is the source MAC address of this packet? (should be the host physical address) | Source: IntelCor\_00:e1:f0 (dc:21:5c:00:e1:f0) |
| What is the destination address of this packet? (should be the default gateway physical address) | Destination: HewlettP\_e0:56:8c (78:48:59:e0:56:8c) |
| What is the source IP address of this packet? (should be the host IP address) | Source Address: 10.241.4.249 |
| What is the destination IP address of this packet? (should be matrix server IP address) | Destination Address: 10.102.108.5 |
| What is the destination port of this packet? (should be a port 80) | Destination Port: 443 |
| What is the source port of this packet? (should be the local dynamic port) | Source Port: 63206 |

**TCP Connection Tear Down: 3-way handshake**

1. The ending of a TCP connection also uses a 3-way handshake (there are multiple ways to end a TCP connection, we will only discuss one way)
2. Scroll down to after the application data has been sent and look for a TCP packet with the FIN and ACK flags set. This is the beginning step of a tear down. After the server has sent the http response message to the client, it sends a packet with the FIN ACK flags set which tells the client there is nothing more. The client responses with an ACK packet back to the server. Notice the ACK packet copies the SEQ number of the FIN ACK packet as the ACK number of the ACK packet and the SEQ number of the FIN ACK packet is used as the ACK number + 1. Thus, the server knows to which TCP connection the client is referring to and closes the connection.



1. A TCP connection is full duplex. At this state, the connection is half closed. The next step is for the client to send a FIN ACK packet to the server and the server responds with an ACK packet which closes the client side of the connection. Notice again how TCP copies the SEQ number of the FIN ACK packet as the SEQ number of the ACK packet and the ACK number is the SEQ number of the FIN ACK packet + 1.

Client

FIN ACK

ACK

FIN ACK

ACK

Half closed

Fully closed

Server

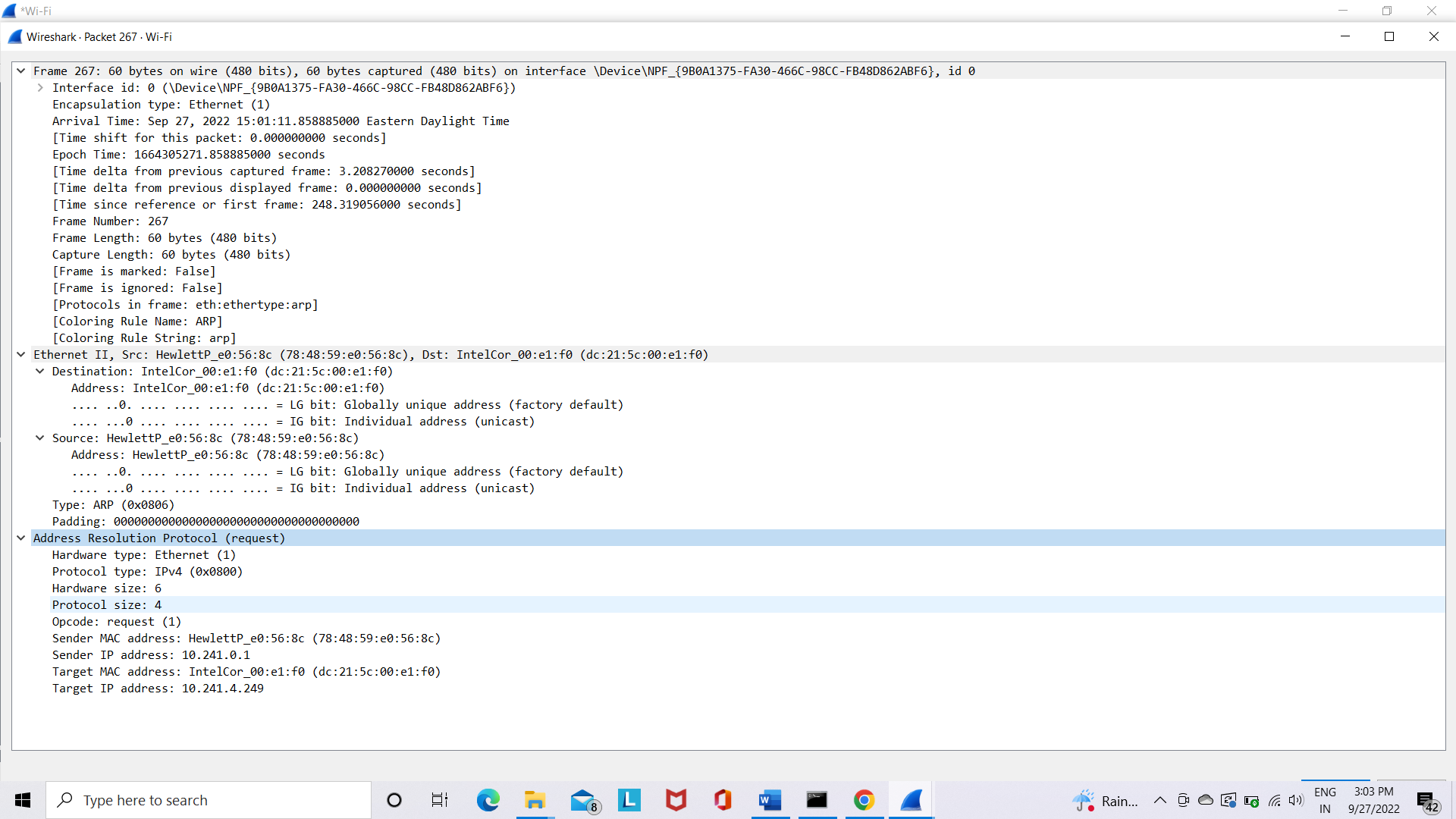
**ARP Protocol**

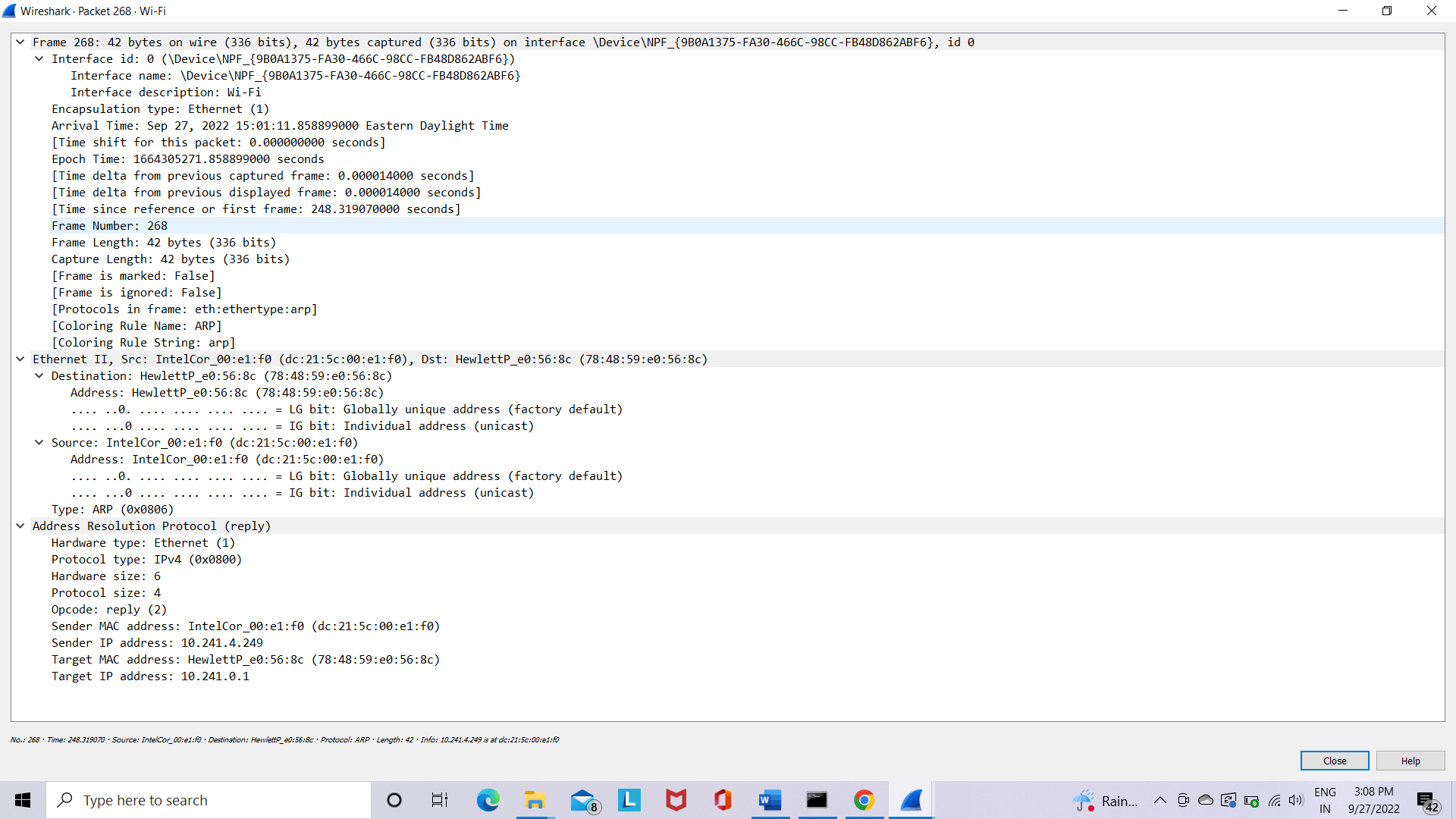
1. The ARP protocol is used to map IP addresses to Ethernet addresses which is used on a LAN. Since we deleted the ARP cache the default gateway physical address is not in the ARP cache, so ARP broadcasts to all hosts on the network, “whoever owns this IP address, please forward to me your MAC address” so I can send you some data. Notice it is the Internet layer that gets the physical or MAC address, which is added to the Data Link header. ARP proves that messages are forwarded link-by-link from source to destination.
2. Click on the first ARP packet.
3. Observe the details pane. Notice that the packet is encapsulated inside an Ethernet frame. Answer the following questions and complete the table below. **[0.4 Marks]**

|  |  |
| --- | --- |
| ARP Request Packet | |
| What type of ARP packet is this (Request/Reply)? | Address Resolution Protocol (request) |
| What is the destination physical address of the ARP packet? | Destination: IntelCor\_00:e1:f0 (dc:21:5c:00:e1:f0) |
| What type of casting is this address? (Unicast, broadcast,multicast) | .... ...0 .... .... .... .... = IG bit: Individual address (unicast) |
| What is the source physical address of the ARP packet? | Address: IntelCor\_00:e1:f0 (dc:21:5c:00:e1:f0) |
| What is the sender’s IP address? | Sender IP address: 10.241.0.1 |
| What is the target’s IP address? | Target IP address: 10.241.4.249 |

1. Scroll down to the next ARP packet.Click on the packet and observer the details pane. Answer the following questions: **[0.4 Marks]**

|  |  |
| --- | --- |
| ARP Reply Packet | |
| What type of ARP packet is this (Request/Reply)? | Address Resolution Protocol (reply) |
| What is the destination physical address of the ARP packet? | Destination: HewlettP\_e0:56:8c (78:48:59:e0:56:8c) |
| What type of casting is this address? (Unicast, broadcast,multicast) | .... ...0 .... .... .... .... = IG bit: Individual address (unicast) |
| What is the source physical address of the ARP packet? | Address: HewlettP\_e0:56:8c (78:48:59:e0:56:8c) |
| What is the sender’s IP address? | Sender IP address: 10.241.4.249 |
| What is the target’s IP address? | Target IP address: 10.241.0.1 |



****

**Other Common Protocols**

Scroll through the list of protocols listed in Wireshark and identify 4 additional protocols. Provide a one sentence description of the protocol’s purpose. (use your own words, do not copy and paste) **[0.25 Marks]**

|  |  |
| --- | --- |
| **Protocol** | **Description** |
| TSLv1.2 | is the newest SSL protocol version supported by OpenSSH on z/VSE |
| DNS | Domain Name System, translates human readable domain names (for example, www.amazon.com) to machine readable IP addresses |
| SSDP | Simple Service Discovery Protocol is a network protocol used in small networks, including home networks, to advertise and discover network services primarily supported by the Universal Plug-and-Play (UPnP) architecture. |
| QUIC | (Quick UDP Internet Connections) is a new generation Internet protocol that speeds online web applications that are susceptible to delay, such as searching, video streaming etc., by reducing the round-trip time (RTT) needed to connect to a server. |

**Grading:**

* DCF255\_L3\_packetcapture.docx – with completed tables
* LearnName\_L3\_packetcapture.pcap

Upload files using link on MySeneca\Graded Work