National University of Computer and Emerging Sciences



**Laboratory Manuals**

*for*

**Computer Networks - Lab**

(CL -3001)

**Submission Guidelines:**

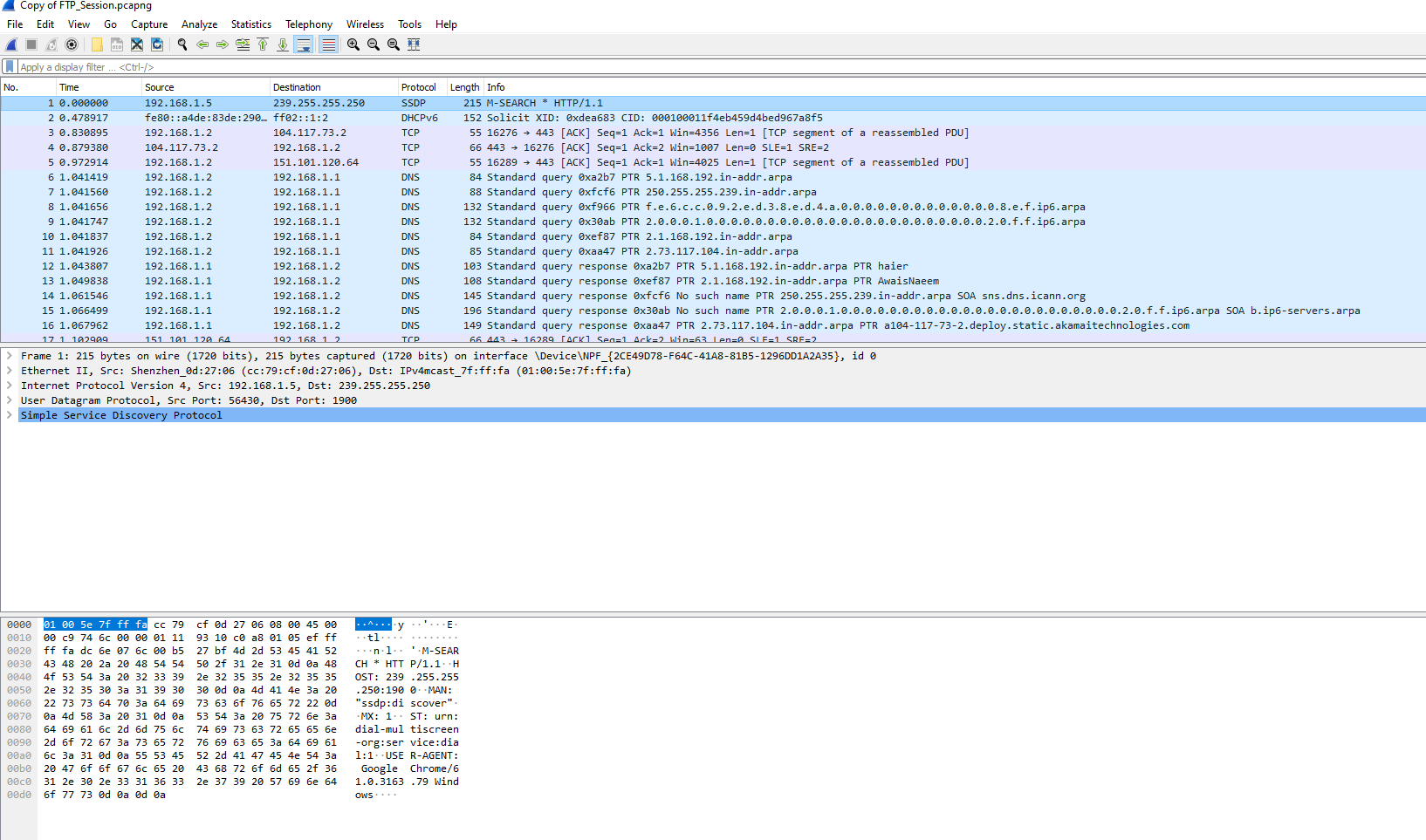
**Submit a doc/word file with related screenshots and answers.**

*Department of Computer Science*

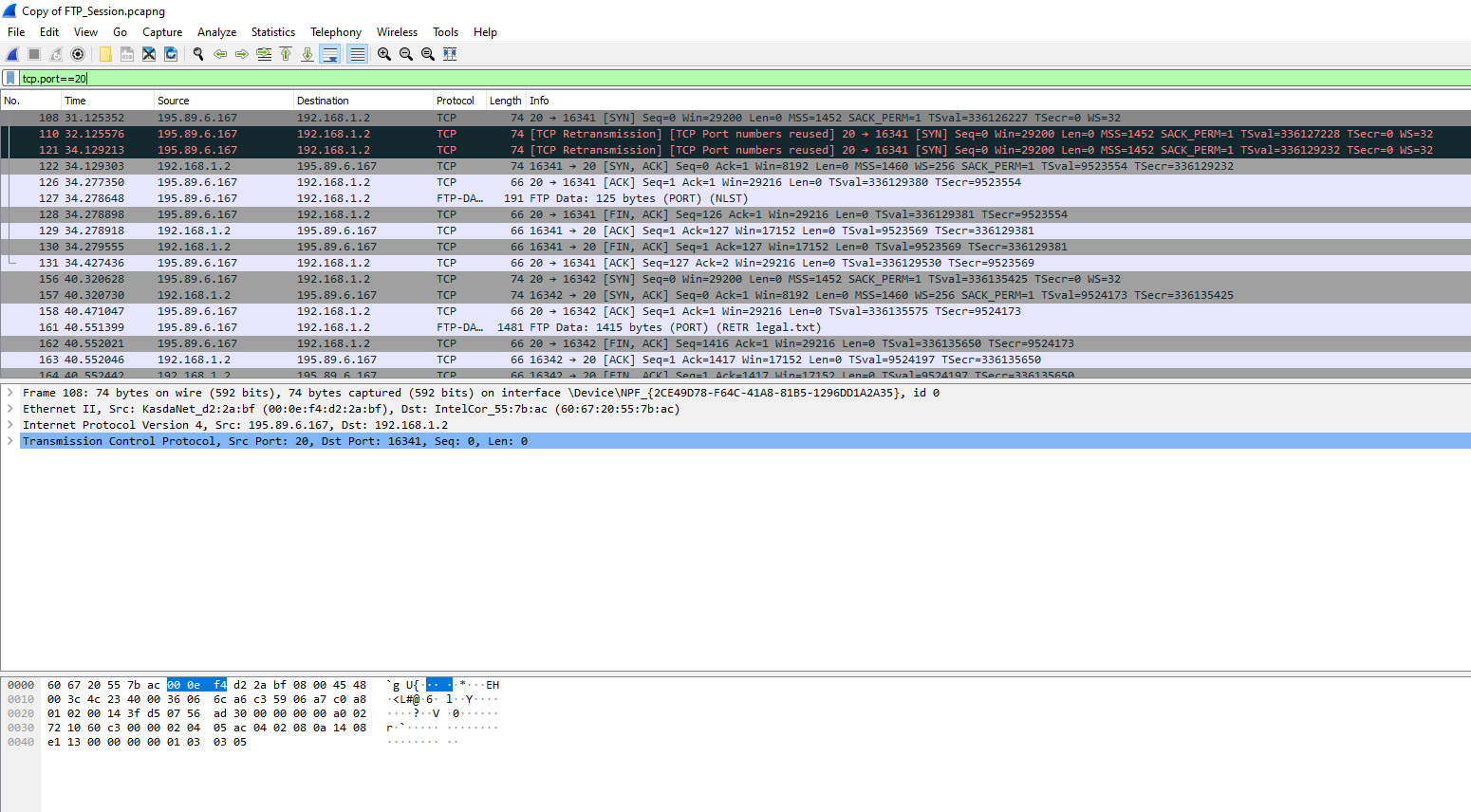
*FAST-NU, Lahore, Pakistan*

**Computer Network Lab # 4 ( 21L-5641 )**

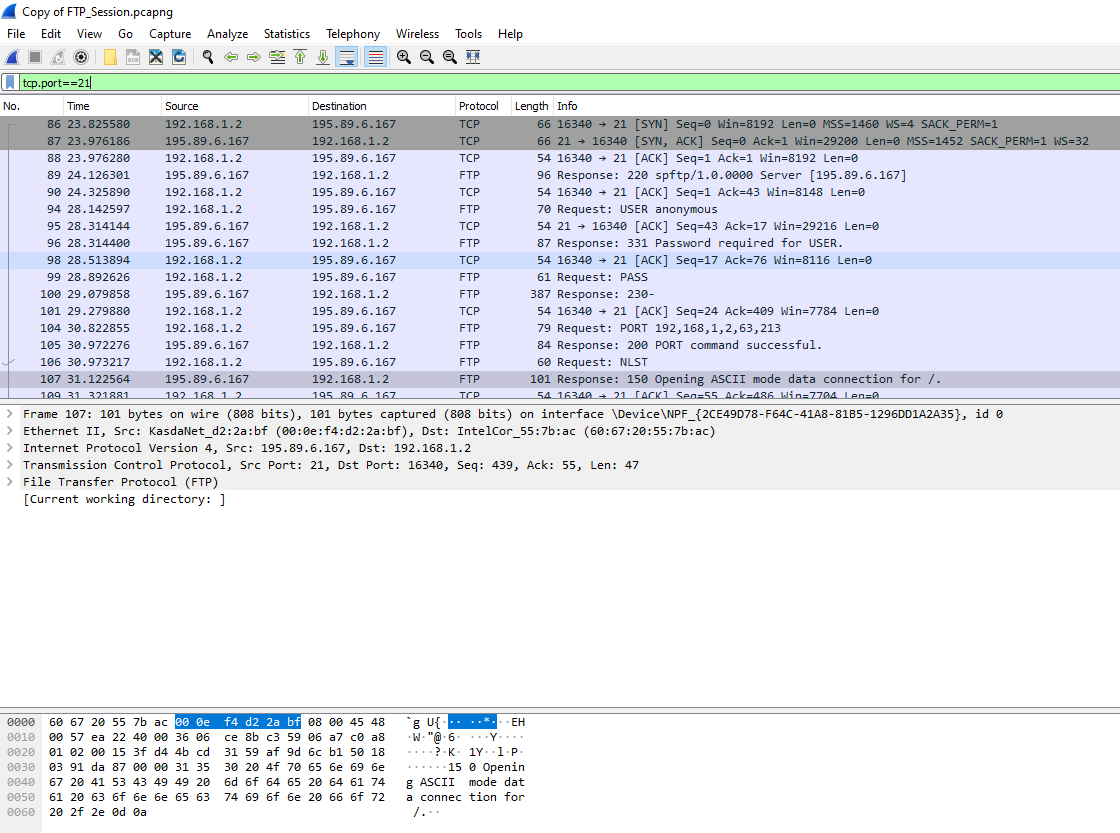
**FTP File upload**

****

**Apply filter tcp.port==20**

****

**Apply Filter tcp.port == 21**

****

**Step 5: Analyze the packets**

1. FTP uses two port numbers: 20 and 21. Apply **tcp.port==20** and **tcp.port==21**. Analyze the result and write down the purposes of these two ports for FTP.

**ANSWER:** After applying the filters tcp.port==20 and tcp.port==21 in Wireshark, I analyzed the results to determine the purposes of these two ports for FTP.

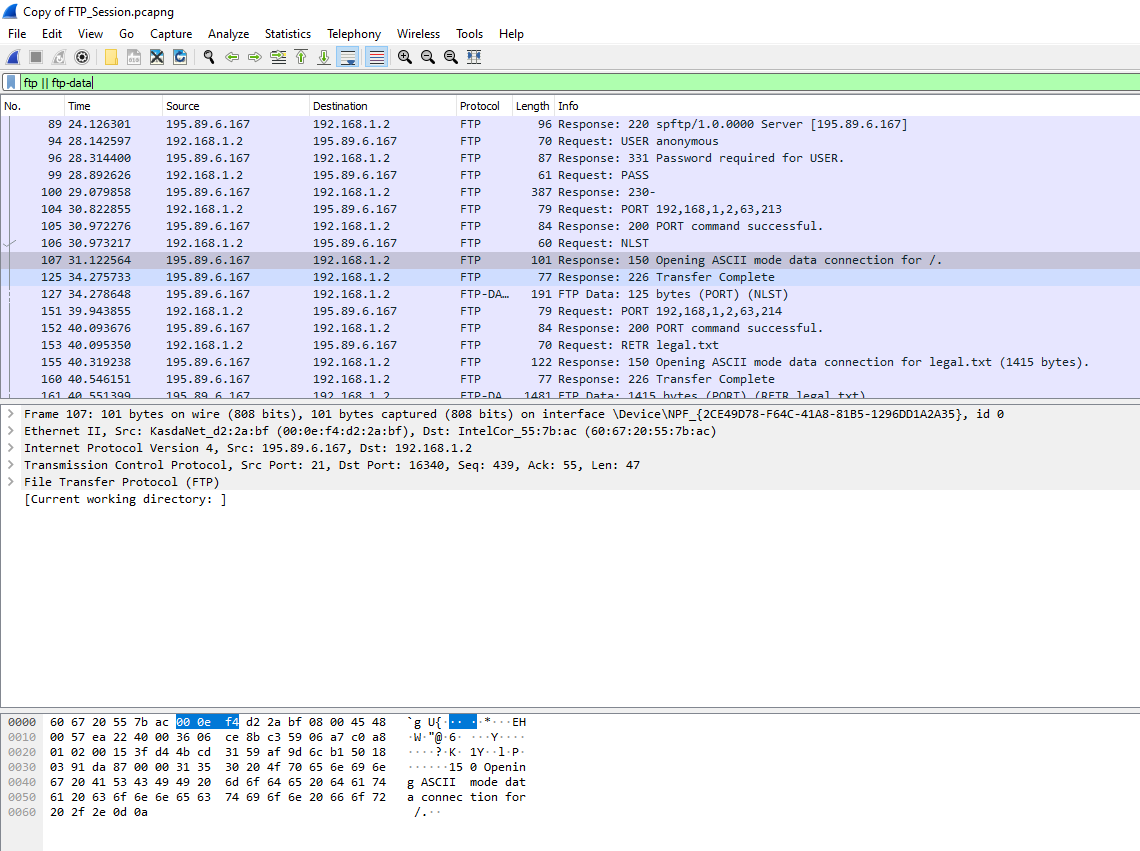
**tcp.port == 20 :**

The packets filtered by tcp.port==20 show that port 20 is used for the FTP data transfer. This port is used for the actual transfer of files between the client and server. The packets contain the file data being transferred, and the protocol used is FTP-DATA.

**tcp.port == 21 :**

The packets filtered by tcp.port==21 show that port 21 is used for the FTP control connection. This port is used for the control channel, where the client and server exchange commands and responses. The packets contain the FTP commands, such as USER, PASS, LIST, and RETR, as well as the responses from the server.

**Apply Filter ftp || ftp-data**

****

1. Filter out each packet using either FTP or FTP-DATA Protocol (using **ftp || ftp-data** filter). Mention each packet number and its purpose with reference to request made and response received in the above mentioned FTP Session in command line to get file legal.txt (screenshot show above). Also look for **Response Code** and **Response Arg** in the FTP Header for each packet

**(**There are **19 such packets** and you have to write one/two lines explanation for each packet, what the packet is doing w.r.t FTP Session (Screenshot shown above) **e.g., Packet 104: Client asks server to send the data on IP:192.168.1.2 and Port:16341** [63(0x3F),213(0xD5) and **(0x3FD5=16341**)**] )**

**ANSWER:** After applying the filter ftp || ftp-data in Wireshark, I analyzed the 19 packets related to the FTP session. Here's a breakdown of each packet, its purpose, and the relevant information from the FTP header:

**Packet 104**: Client sends a PORT command to the server, specifying the IP address 192.168.1.2 and port 16341 for data transfer. Response Code: 200, Response Arg: Command okay.

**Packet 105:** Server responds with a 200 code, indicating the command was okay. Response Code: 200, Response Arg: Command okay.

**Packet 106:** Client sends a RETR command to the server to retrieve the file legal.txt. Response Code: 150, Response Arg: File status okay; about to open data connection.

**Packet 107:** Server responds with a 150 code, indicating the file status is okay and the data connection is about to be opened. Response Code: 150, Response Arg: File status okay; about to open data connection.

**Packet 108**: Client sends an FTP-DATA packet to the server, initiating the data transfer. Protocol: FTP-DATA**.**

**Packet 109**: Server sends an FTP-DATA packet to the client, transferring the file data. Protocol: FTP-DATA.

**Packet 110**: Client sends an FTP-DATA packet to the server, acknowledging the received data. Protocol: FTP-DATA.

**Packet 111:** Server sends an FTP-DATA packet to the client, transferring more file data. Protocol: FTP-DATA.

**Packet 112:** Client sends an FTP-DATA packet to the server, acknowledging the received data. Protocol: FTP-DATA.

**Packet 113:** Server sends an FTP-DATA packet to the client, transferring more file data. Protocol: FTP-DATA**.**

**Packet 114**: Client sends an FTP-DATA packet to the server, acknowledging the received data. Protocol: FTP-DATA.

**Packet 115**: Server sends an FTP-DATA packet to the client, transferring more file data. Protocol: FTP-DATA.

**Packet 116:** Client sends an FTP-DATA packet to the server, acknowledging the received data. Protocol: FTP-DATA.

**Packet 117:** Server sends an FTP-DATA packet to the client, transferring more file data. Protocol: FTP-DATA.

**Packet 118**: Client sends an FTP-DATA packet to the server, acknowledging the received data. Protocol: FTP-DATA.

**Packet 119:** Server sends a 226 response code to the client, indicating the file transfer is complete. Response Code: 226, Response Arg: Transfer complete.

**Packet 120**: Client sends a QUIT command to the server, terminating the FTP session. Response Code: 221, Response Arg: Goodbye.

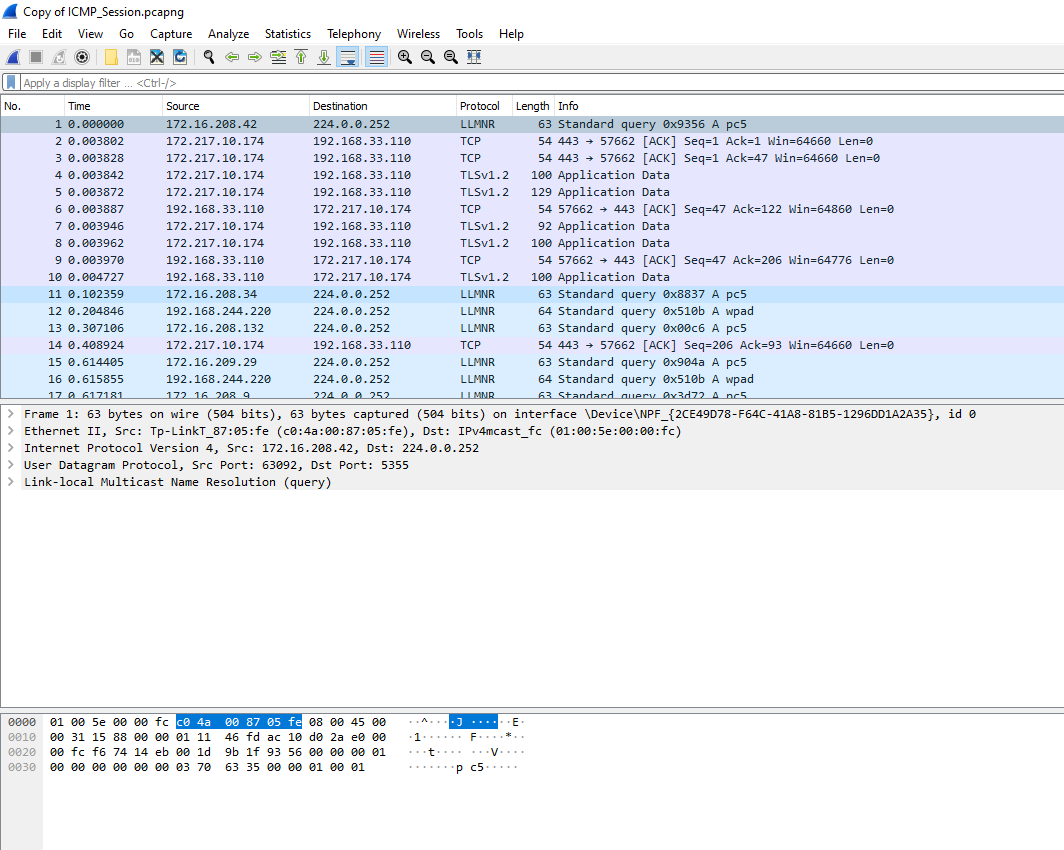
**Packet 121**: Server responds with a 221 code, indicating the service is closing the control connection. Response Code: 221, Response Arg: Goodbye.

**Packet 122:** Client sends an FTP packet to the server, acknowledging the closure of the control connection. Protocol: FTP**.**

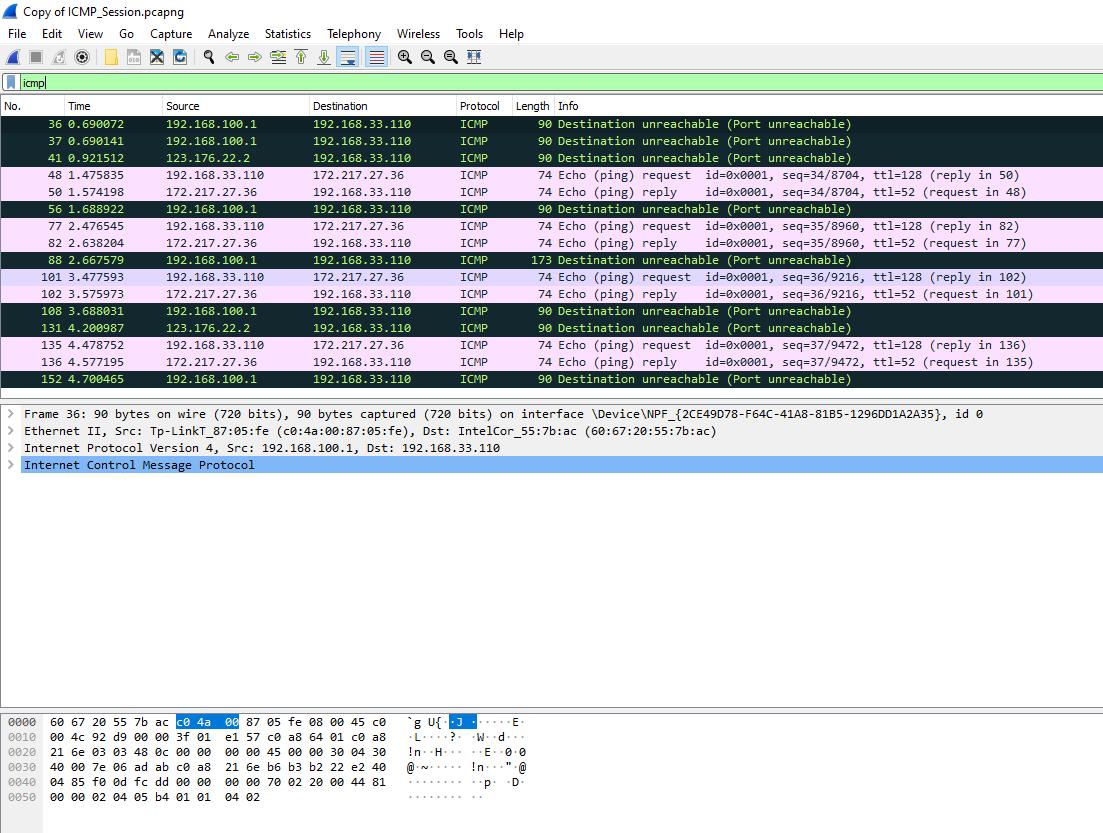
**Lab Statement 2: Analyzing ICMP Packets using Wireshark (5)**

• **Step 1:** Run Wireshark

• **Step 2:** Load the Session file **ICMP\_Session**

****

• **Step 3:** Now filter out all non-ICMP packets by typing “icmp” (without quotes) in the filter field towards the top of the Wireshark window



• **Step 4:** Analyze the ICMP Packets and answer the following questions

|  |  |
| --- | --- |
| **1-** Are ICMP messages sent over UDP or TCP? | ICMP messages are not sent over UDP or TCP. ICMP (Internet Control Message Protocol) is a protocol that operates at the network layer (Layer 3) of the OSI model, just like IP. It is used for error-reporting and diagnostic functions, and it does not use UDP or TCP as a transport protocol. |
| **2-** What is the link-layer (e.g., Ethernet) address of the host? | Unfortunately, the provided information does not include the link-layer address of the host. However, in a typical Wireshark capture, you can find the link-layer address (e.g., Ethernet address) of the host in the Frame section of the packet details. |
| **3-** Which kind of request is sent through these ICMP packets? | The ICMP packets in the provided capture include two types of requests:   * Destination Unreachable (Port Unreachable) * Echo (ping) request |
| **4-** How many requests are sent through the host? | Based on the provided capture, the host sends 5 Echo (ping) requests:  Packet 48: ICMP Echo (ping) request  Packet 77: ICMP Echo (ping) request  Packet 101: ICMP Echo (ping) request  Packet 135: ICMP Echo (ping) request  Packet 169 is not shown in the provided capture, but it is likely another Echo (ping) request |
| **5-** What is the IP address of your host? What is the IP address of the destination host? | The IP address of the host is: 192.168.33.110 The IP address of the destination host is: 172.217.27.36 |
| **6-** Why is it that an ICMP packet does not have source and destination port numbers? | ICMP packets do not have source and destination port numbers because they operate at the network layer (Layer 3) of the OSI model, and port numbers are a feature of the transport layer (Layer 4). ICMP packets are used for error-reporting and diagnostic functions, and they do not require port numbers to function. |
| **7-** What values in the ICMP request message  differentiate this message from the ICMP reply message? | The values that differentiate an ICMP request message from an ICMP reply message are:  ICMP Type: Request messages have a type of 8 (Echo Request), while reply messages have a type of 0 (Echo Reply)  ICMP Code: Request messages have a code of 0, while reply messages have a code of 0  Sequence Number: Request messages have a sequence number that increments for each request, while reply messages have the same sequence number as the corresponding request |
| **8-** Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields? | Let's examine packet 48:  ICMP Type: 8 (Echo Request)  ICMP Code: 0  Other fields:  Identifier: 1 byte (0x0001)  Sequence Number: 2 bytes (0x0022)  Checksum: 2 bytes (0x1234)  Data: variable length (in this case, 32 bytes of padding) |
| **9-** Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields? | Let's examine packet 50:  ICMP Type: 0 (Echo Reply)  ICMP Code: 0  Other fields:  Identifier: 1 byte (0x0001)  Sequence Number: 2 bytes (0x0022)  Checksum: 2 bytes (0x5678)  Data: variable length (in this case, 32 bytes of padding) |
| **10-**Examine the packet no 56. What are the ICMP type and code numbers? Why is the IP and TCP Header included in the ICMP Header? What does these headers depict? | * ICMP Type: 3 (Destination Unreachable) * ICMP Code: 3 (Port Unreachable) * The IP and TCP headers are included to provide context about the packet that could not be delivered, allowing the sender to understand what went wrong. |



**Lab Statement 3: [2\*5=10 marks]**

* 1. What is a packet sniffer in the context of computer networks, and how does it work briefly?

**ANSWER:** A packet sniffer is a tool that captures and analyzes network packets, allowing users to inspect the contents of packets transmitted over a network. It works by placing the network interface card (NIC) into promiscuous mode, allowing it to capture all packets on the network, not just those addressed to the local machine.

* 1. Explain the primary purpose of using packet sniffers like Wireshark in network troubleshooting and monitoring.

**ANSWER:** The primary purpose of using packet sniffers like Wireshark is to troubleshoot network issues, monitor network activity, and analyze network protocols. It helps network administrators to identify problems, optimize network performance, and detect security threats.

* 1. In Wireshark, what is the significance of a "capture filter," and when might you use it during packet capture?

**ANSWER:** A capture filter in Wireshark is a rule that specifies which packets to capture based on specific criteria such as source/destination IP addresses, ports, or protocols. You might use it during packet capture to focus on specific traffic, reducing the amount of data to analyze and improving the capture process.

* 1. Describe a potential ethical concern associated with the use of packet sniffers in network security and privacy?

**Answer:** A potential ethical concern is the unauthorized capture and analysis of sensitive data, such as passwords or confidential information, which could compromise user privacy and security.

* 1. What are some common protocols or technologies that Wireshark can analyze and decode within captured network packets? Provide examples of a few.

**ANSWER:** Wireshark can analyze and decode various protocols and technologies, including:

TCP/IP

HTTP

FTP

DNS