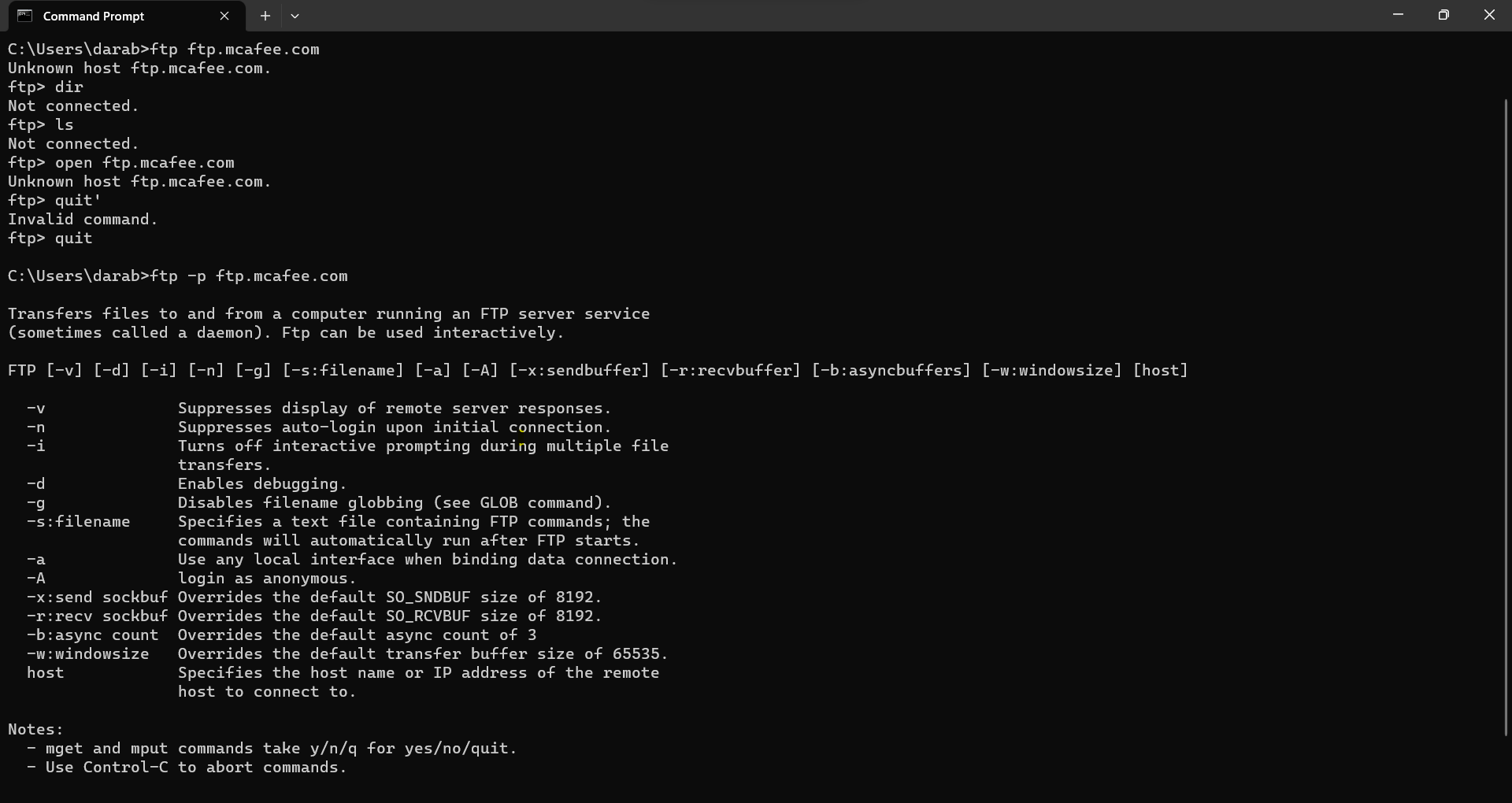
**Roll no 21L-7512 Abdullah Dar BSCS-5G-1**

**Computer Networks**

**Lab no 4**

**Lab Statement 1: Capturing FTP packets using Wireshark.**

Unable to connect to ftp server from my laptop tried several times with different methods but unsuccessful. Adding proof for that too.



**Step 5: Analyse 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:**

FTP (File Transfer Protocol) indeed uses two port numbers: 20 and 21. These ports serve different purposes in the FTP communication process:

1. **Port 21 (Control Port)**:
   * Port 21 is known as the control port in FTP.
   * It is used for establishing and managing the control connection between the FTP client and server.
   * The control connection is responsible for sending commands from the client to the server (e.g., login, change directory, list directory contents) and receiving responses from the server.
   * Port 21 typically uses the FTP command channel to transmit control information and does not transfer actual file data.
2. **Port 20 (Data Port)**:
   * Port 20 is known as the data port in FTP.
   * It is used for the actual data transfer between the FTP client and server.
   * When a file transfer is initiated (e.g., when the client requests to download or upload a file), the data connection is established through port 20.
   * Data can be transferred in two modes: active and passive. In active mode, the server initiates the data connection from port 20 to the client. In passive mode, the client initiates the data connection to a port specified by the server. Regardless of the mode, port 20 is typically involved in data transfer.
3. **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:**

1. **Packet 89**: This is a response packet from the FTP server. It indicates that the server is ready to communicate, with a message "220 spftp/1.0.0000 Server [195.89.6.167]."
2. **Packet 94**: This packet is a request from the FTP client. It sends a "USER" command with the username "anonymous."
3. **Packet 96**: The server responds to the "USER" command with "331 Password required for USER," indicating that a password is needed.
4. **Packet 99**: The client sends a "PASS" command, which is used to provide the password.
5. **Packet 100**: The server responds with "230-" to confirm that the login was successful. Additional information may follow this response.
6. **Packet 104**: The client sends a "PORT" command, specifying the IP and port for data transfer (e.g., "PORT 192,168,1,2,63,213").
7. **Packet 105**: The server acknowledges the successful receipt of the "PORT" command with "200 PORT command successful."
8. **Packet 106**: The client sends an "NLST" command, which requests a list of files in the current directory.
9. **Packet 107**: The server responds with "150 Opening ASCII mode data connection for /," indicating that it's preparing to send the directory listing.
10. **Packet 125**: The server confirms the successful transfer of the directory listing with "226 Transfer Complete."
11. **Packet 151**: The client sends another "PORT" command, possibly for the upcoming data transfer (e.g., "PORT 192,168,1,2,63,214").
12. **Packet 152**: The server acknowledges the successful receipt of the second "PORT" command with "200 PORT command successful."
13. **Packet 153**: The client requests to retrieve a file named "legal.txt" using the "RETR" command.
14. **Packet 155**: The server responds with "150 Opening ASCII mode data connection for legal.txt (1415 bytes)," indicating that it's preparing to send the file "legal.txt."
15. **Packet 160**: The server confirms the successful transfer of the "legal.txt" file with "226 Transfer Complete."
16. **Packet 173**: The client sends a "QUIT" command, signaling the end of the FTP session.
17. **Packet 175**: The server responds with "221 Goodbye," indicating that the session is ending.

These packets collectively represent the steps of an FTP session where the client logs in, retrieves a file ("legal.txt"), and then gracefully terminates the session.

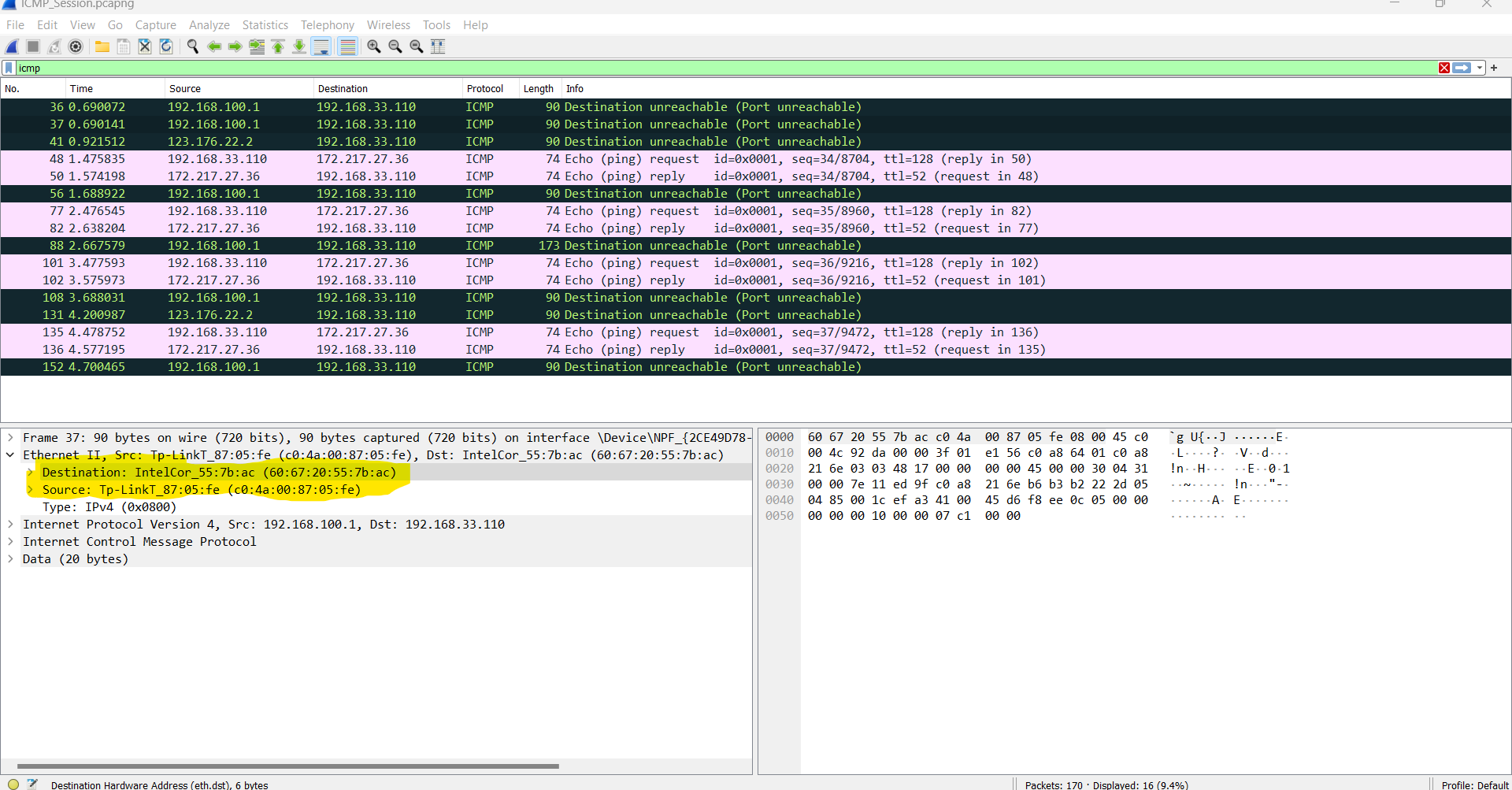
**Lab Statement 2: Analyzing ICMP Packets using Wireshark.**

**1- Are ICMP messages sent over UDP or TCP?**

**Answer:** ICMP messages are sent independently using their own ICMP protocol and are not encapsulated within UDP or TCP packets.

**2- What is the link-layer (e.g., Ethernet) address of the host?**

Answer: The link-layer address (Ethernet address) of the host in Frame 37 is "**Tp-LinkT\_87:05:fe**" and its MAC address is "**c0:4a:00:87:05:fe**"



**3- Which kind of request is sent through these ICMP packets?**

Answer: The ICMP packets in the provided data are "Echo (ping) requests."

A screenshot of a computer

Description automatically generated

**4- How many requests are sent through the host?**

Answer: There are 4 requests sent through the host.

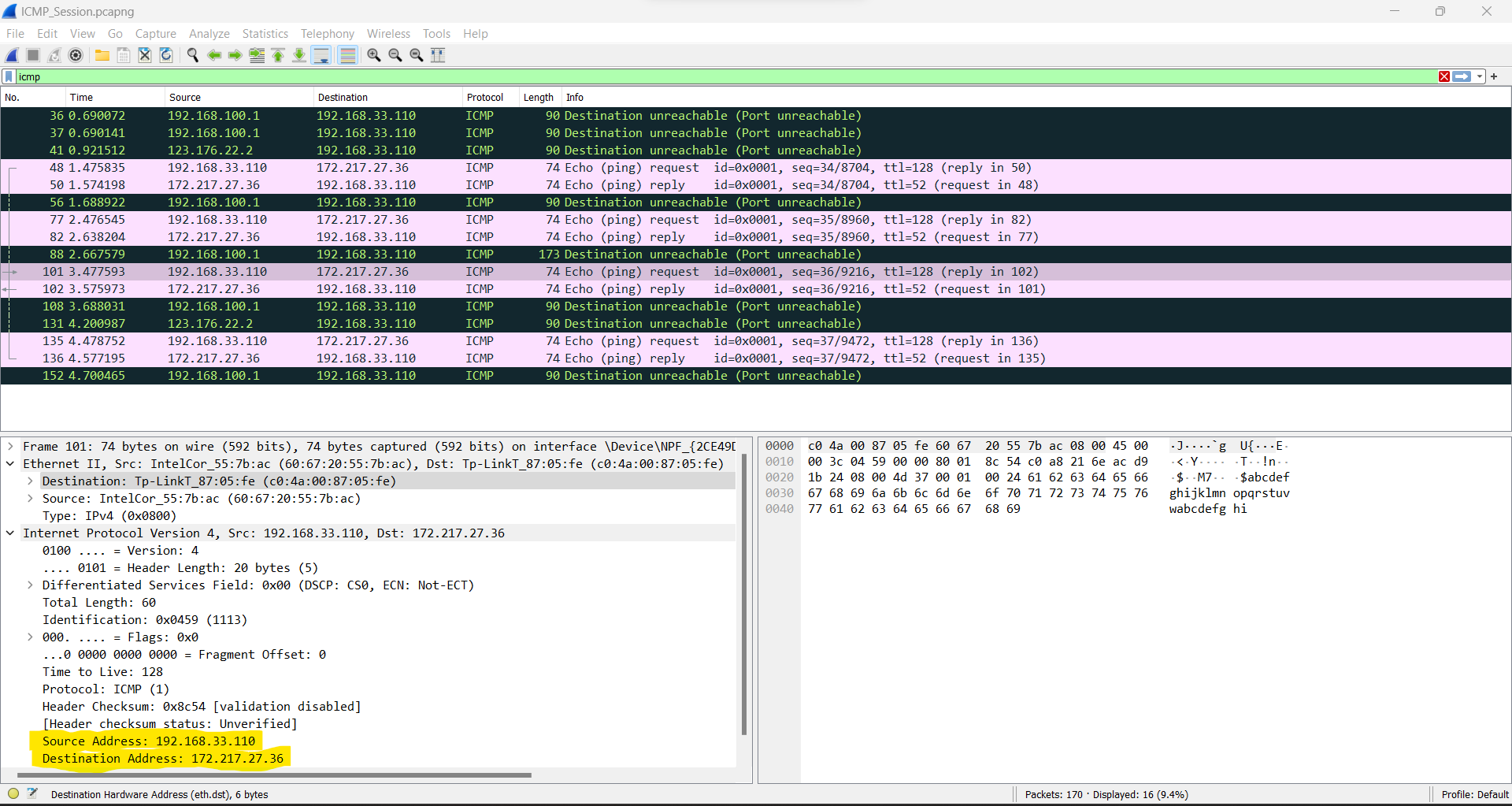
A screenshot of a computer

Description automatically generated

**5- What is the IP address of your host? What is the IP address of the destination host?**

Answer: The IP address of your host (**source**) is **192.168.33.110**

The IP address of the destination host (**destination**) is **172.217.27.36**



**6- Why is it that an ICMP packet does not have source and destination port numbers?**

Answer: ICMP packets lack source and destination port numbers because they're intended for network-layer communication between hosts and routers, not application-layer processes, focusing on network diagnostics and error reporting.

**7- What values in the ICMP request message differentiate this message from the ICMP reply message?**

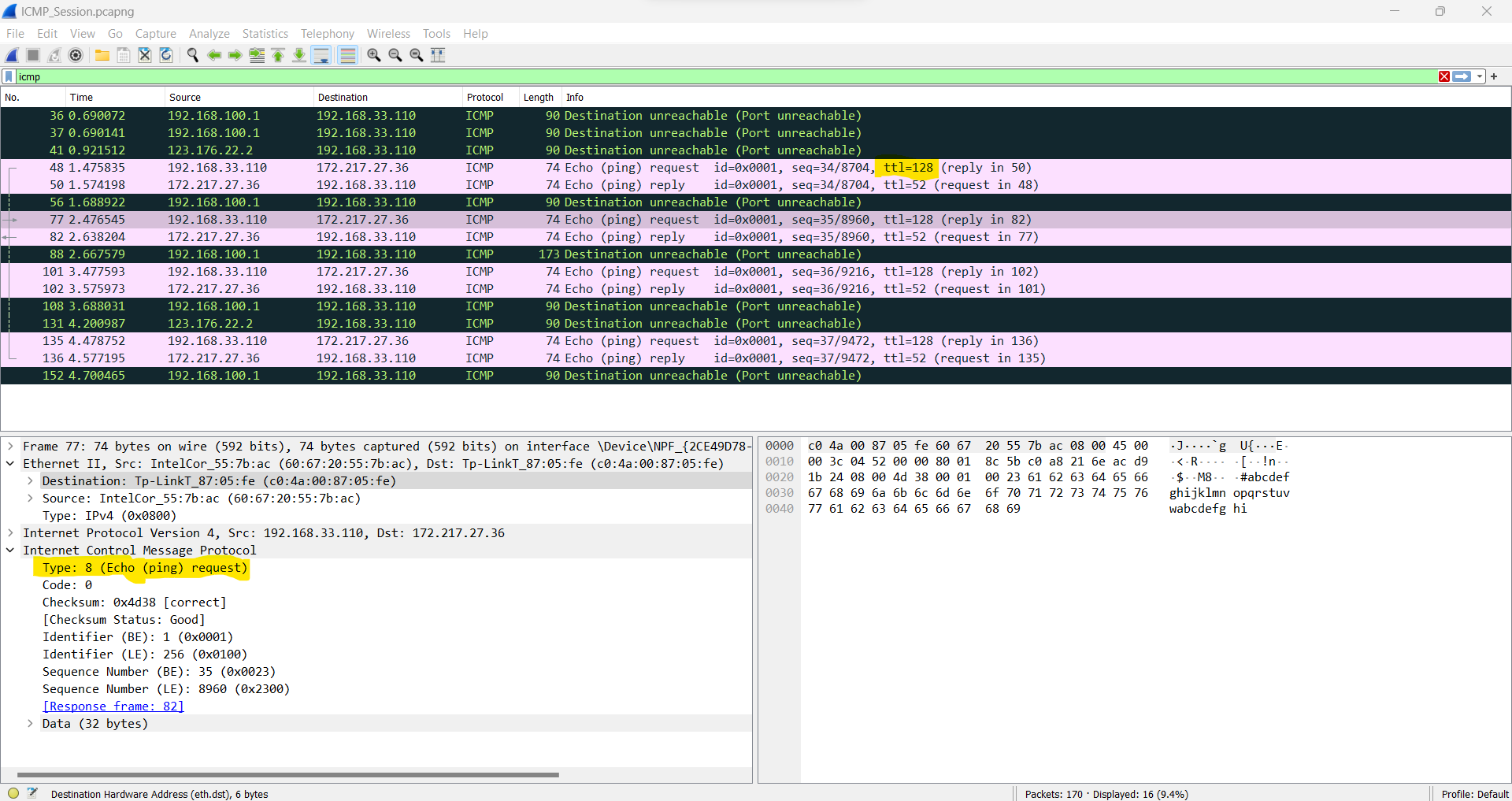
Answer: ICMP request and reply message differentiate this message from the as follows

**Type Field**:

* In the ICMP request message the Type field is set to 8, which indicates an Echo Request.
* In the ICMP reply message the Type field is set to 0, which indicates an Echo Reply.

**Time to Live (TTL) Field:**

* In the ICMP request message the TTL field is set to 128.
* In the ICMP reply message the TTL field is set to 52.



A screenshot of a computer

Description automatically generated

**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?**

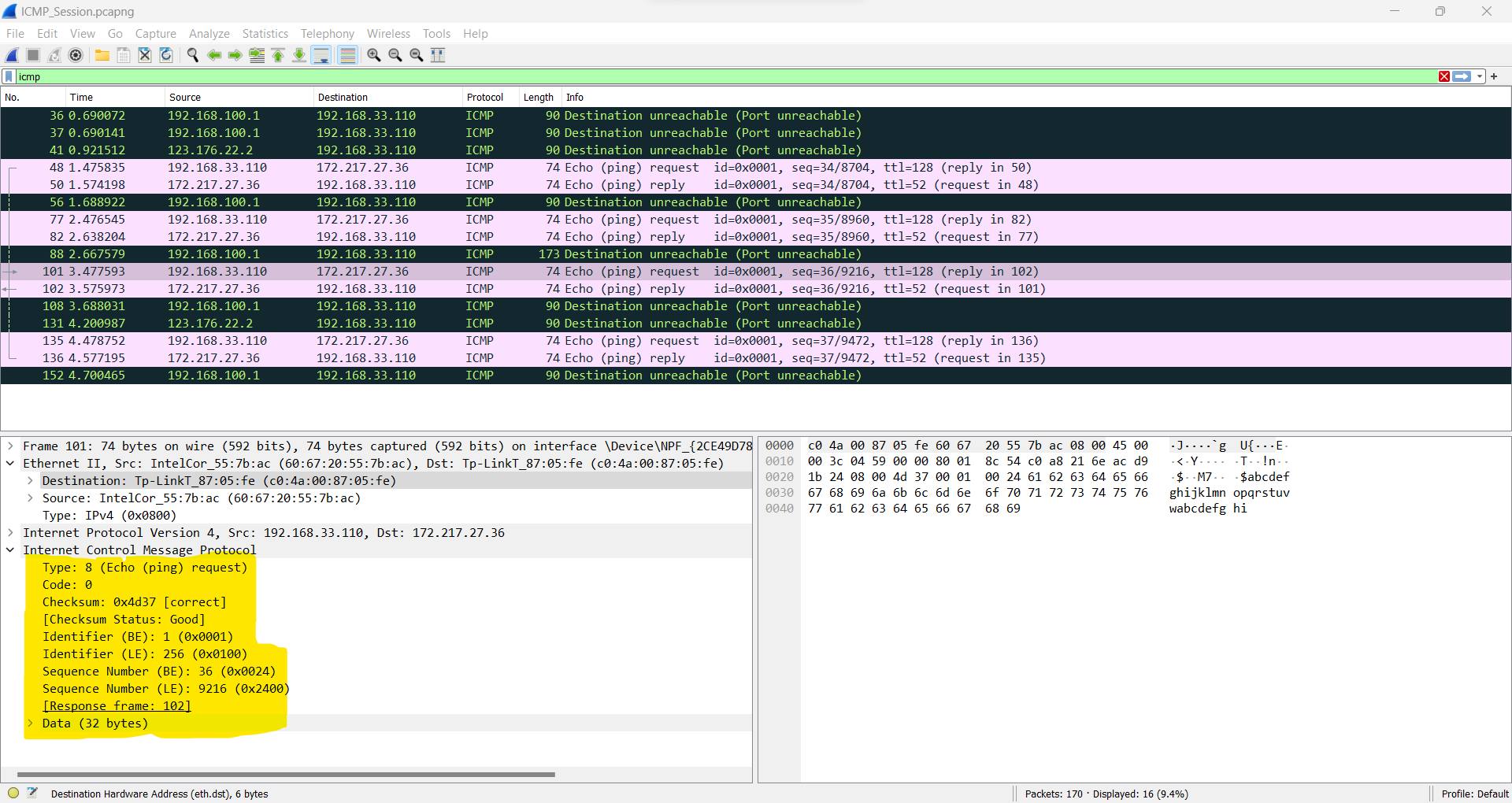
Answer:

**Request:** “101 3.477593 192.168.33.110 172.217.27.36 ICMP 74 Echo (ping) request id=0x0001, seq=36/9216, ttl=128 (reply in 102)”

Packet 101 is an ICMP Echo Request (ping) packet sent by your host to 172.217.27.36

Its Details are as follows:

* ICMP Type: 8
* ICMP Code: 0
* Identifier (ID) Field: 0x0001
* Sequence Number Field: 36/9216
* Checksum: 2 bytes

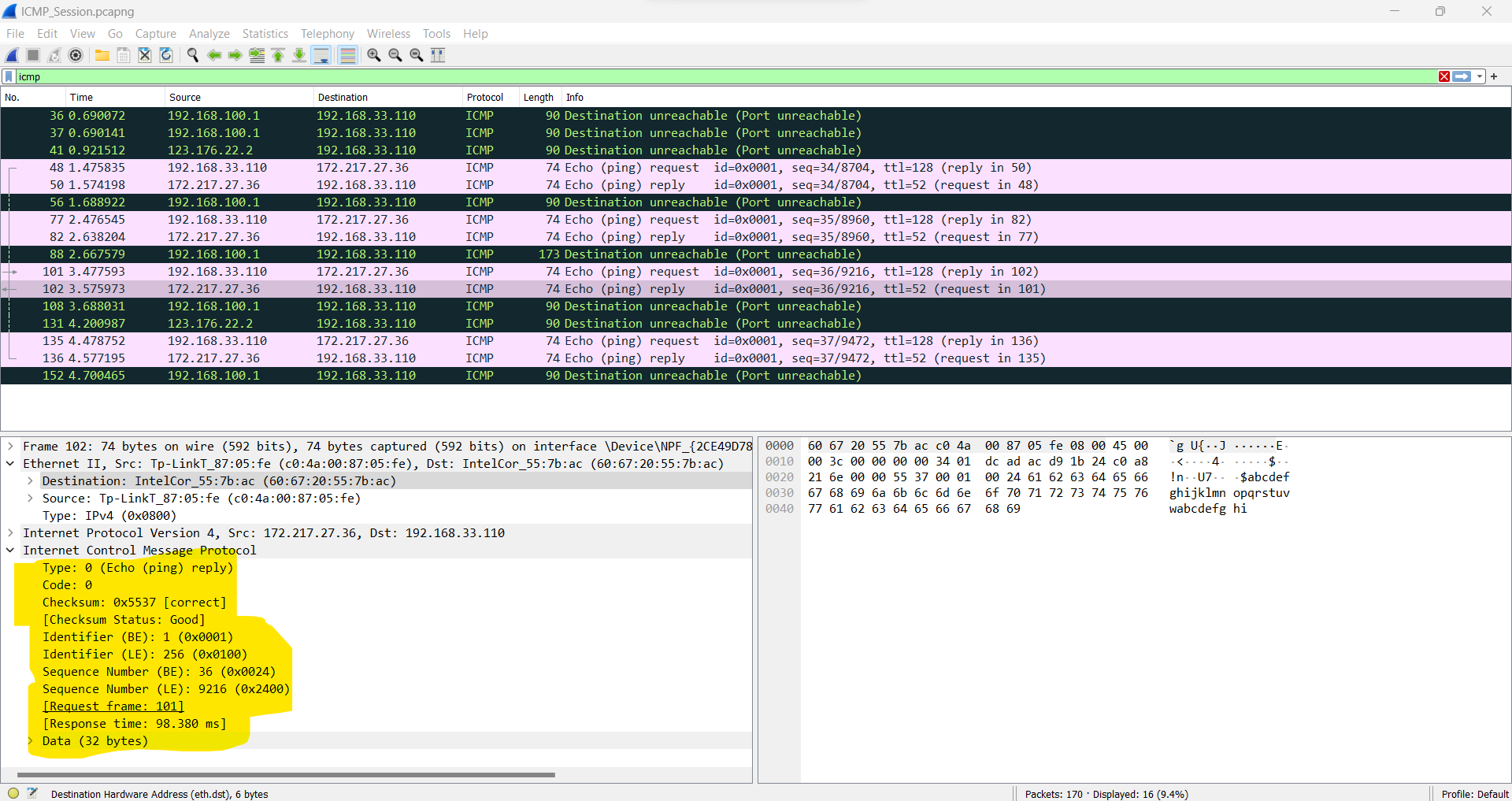


**9- Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?**

Answer:

**Request:** “102 3.575973 172.217.27.36 192.168.33.110 ICMP 74 Echo (ping) reply id=0x0001, seq=36/9216, ttl=52 (request in 101)”

Packet 102 is an ICMP Echo Reply (ping reply) packet sent in response to the Echo Request (ping) packet with sequence number 36/9216. Let's break down the relevant fields in this ICMP packet:

* ICMP Type: 0
* ICMP Code: 0
* Identifier (ID) Field: 0x0001 (same as in Echo Request)
* Sequence Number Field: 36/9216 (same as in Echo Request)
* Checksum: 2 bytes
* TTL: 52
* 

**10-Examine packet no 56. What are the ICMP type and code numbers? Why is the IP and TCP Header included in the ICMP Header? What do these headers depict?**

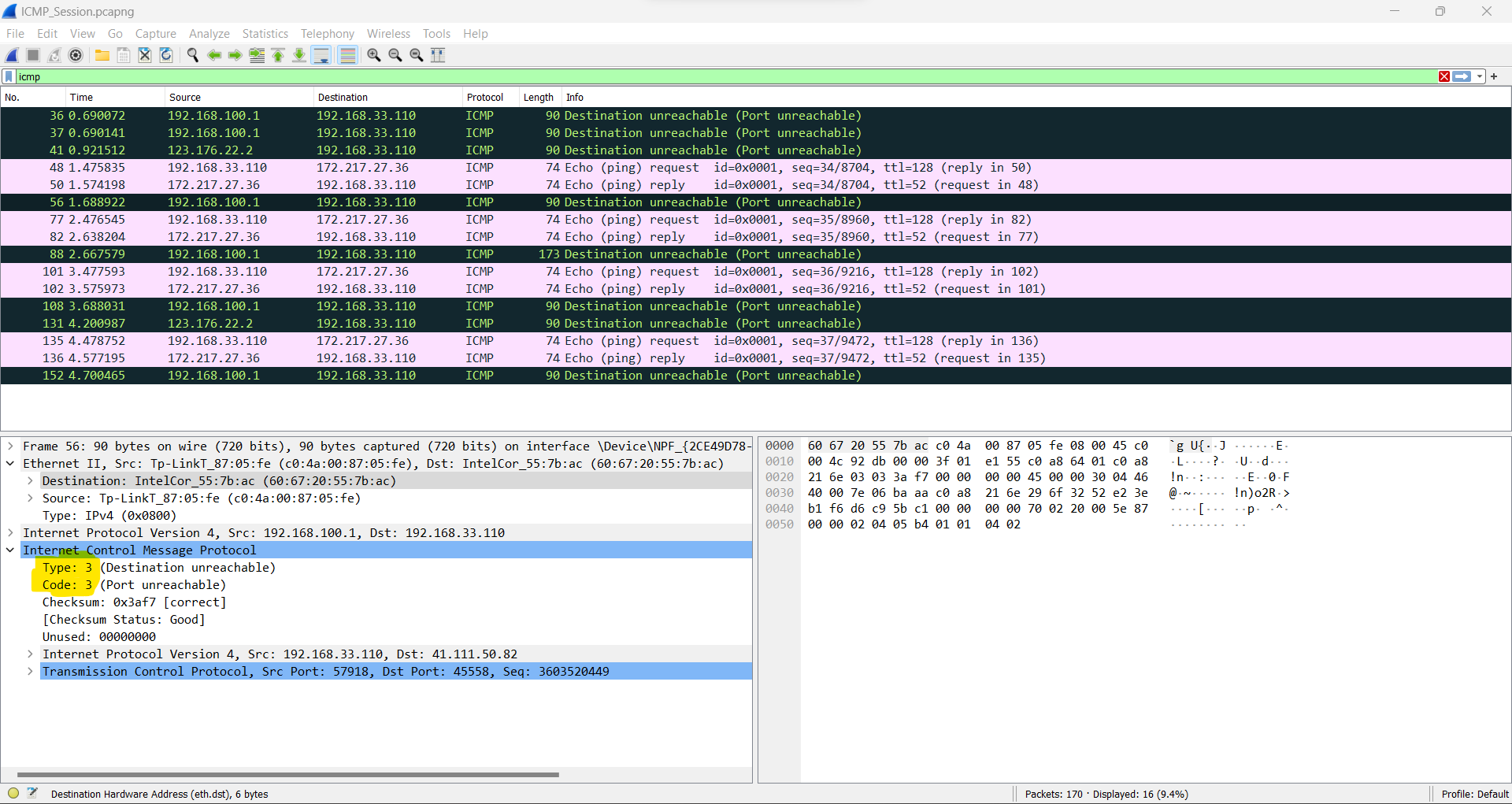
Answer:

**Request:** “56 1.688922 192.168.100.1 192.168.33.110 ICMP 90 Destination unreachable (Port unreachable)”

Packet 56 is an ICMP "Destination unreachable (Port unreachable)" message.

* **ICMP Type**: 3
* **ICMP Code**: 3

The ICMP message includes a portion of the original IP and TCP headers to provide context about why the destination is unreachable. It indicates that the destination port (TCP) on the target host is unreachable, likely due to no service listening on that port or a firewall blocking access.



**Lab Statement 3**

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

**Answer:** A packet sniffer is a network tool that intercepts and analyses data packets traveling across a computer network. It works by capturing packets from the network, extracting information from them, and displaying or logging the data for analysis. Packet sniffers are used for troubleshooting, monitoring network traffic, and sometimes for malicious purposes like eavesdropping on sensitive information.

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

Answer: Packet sniffers like Wireshark are essential for network troubleshooting and monitoring because they provide detailed insights into network traffic. Their primary purposes include:

1. **Troubleshooting**: They help identify and diagnose network issues, such as packet loss, latency problems, or misconfigured devices. By analysing captured packets, network administrators can pinpoint the source of problems and take corrective actions.
2. **Monitoring**: Packet sniffers allow continuous monitoring of network traffic. This helps in detecting unusual or unauthorized activities, optimizing network performance, and ensuring compliance with security policies, making networks more robust and secure.

Overall, packet sniffers are indispensable tools for maintaining reliable and secure computer networks.

**c. In Wireshark, what is the significance of a “capture filter” and when might**

**you use it during packet capture?**

Answer: In Wireshark, a "capture filter" is a set of rules that define specific criteria for capturing network packets. Unlike a display filter, which filters packets for viewing after capture, a capture filter selects which packets are recorded during packet capture. Capture filters are useful when you want to limit the amount of data collected, especially in high-traffic environments. They help conserve storage space and reduce the processing overhead during packet capture. You might use a capture filter to focus on capturing packets from a particular source IP address, port, or protocol, making analysis more efficient and relevant to your specific needs.

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

Answer: An ethical concern associated with the use of packet sniffers in network security and privacy is the potential for invasive surveillance and privacy breaches. When used without proper authorization or for malicious purposes, packet sniffers can capture sensitive and private information, such as login credentials, personal communications, and confidential data, without the knowledge or consent of individuals. This can lead to privacy violations, identity theft, and other forms of cybercrime. Therefore, it is crucial to use packet sniffers responsibly, following legal and ethical guidelines, and only in legitimate network management or security contexts to avoid infringing on individuals' privacy rights.

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

Answer: Wireshark can analyse and decode a wide range of common network protocols and technologies, including:

1. **HTTP**: Analyses web traffic, allowing inspection of websites and HTTP requests.
2. **TCP/IP**: Decodes Transmission Control Protocol/Internet Protocol, crucial for internet communication.
3. **DNS**: Reveals domain name resolution requests and responses.
4. **SMTP/POP3/IMAP**: Analyses email protocols, aiding in email troubleshooting.
5. **SSL/TLS**: Decrypts secure connections, enabling inspection of encrypted traffic.
6. **VoIP (e.g., SIP and RTP)**: Assists in troubleshooting voice and video calls.
7. **FTP/SSH**: Helps monitor file transfers and secure shell sessions.
8. **ARP**: Resolves IP addresses to MAC addresses on local networks.

These examples demonstrate Wireshark's versatility in dissecting various network protocols.