🦈 Wireshark Packet Analysis Lab

1. Setup & Installation

First, we need to ensure your Mininet VM has the necessary "eyes" to see the traffic.

Step A: Install Wireshark
Open your main terminal and run:

```
sudo apt-get update
sudo apt-get install wireshark
```

*(If prompted to allow non-superusers to capture packets, select **Yes**).*

Step B: Start Mininet
We will use a simple network with two hosts (h1 and h2) connected by a switch.

```
sudo mn
```

2. The Experiment (Capturing the Ping)

We need to capture the conversation between Host 1 and Host 2.

Step 1: Open Host Terminals
Inside the Mininet console (mininet>), open a terminal for h1:

```
xterm h1
```

Step 2: Start Wireshark (The Listener)
In the new h1 terminal window (the black box), type:

```
wireshark &
```

(The & lets it run in the background).

1. When Wireshark opens, look at the interface list.
2. Double-click **h1-eth0** (this is h1's network card).
3. You will see the screen is currently empty or quiet.

Step 3: Generate Traffic (The Talker)
Go back to the h1 terminal (or open h2) and ping the other host.

```
ping 10.0.0.2 -c 4
```

Step 4: Stop and Save

1. Look at Wireshark. You should see colorful lines appearing (Light blue/pink).
2. Click the red square **Stop** button.
3. Go to **File -> Save As** and save it as ping_capture.pcap.

3. The Time Diagram

When you look at the captured file, you are seeing a timeline of events. A PING operation (ICMP) follows a strict request-and-reply structure.

Here is the Time Diagram of what happened in your capture:

```
Time (t)      HOST 1 (10.0.0.1)                    HOST 2 (10.0.0.2)
   |              |                                 |
   |   [ARP Req]     |  "Who has 10.0.0.2? Tell 10.0.0.1"  |
   | ----------------|------------------------------------->|
   |              |                                 |
   |   [ARP Rep]     |  "10.0.0.2 is at AA:BB:CC:DD:EE:FF" |
   |<----------------|-------------------------------------|
   |              |                                 |
   |              |       (Now they can talk)          |
   |              |                                 |
   |   [ICMP Req]    |      Echo (Ping) Request  ---->     |
   | ----------------|------------------------------------->|
   |              |                                 |
   |   [ICMP Rep]    |    <---- Echo (Ping) Reply        |
   |<----------------|-------------------------------------|
   |              |                                 |
   v              v                                 v
```

Note: You might see the ARP (Address Resolution Protocol) packets first if the hosts haven't talked before. This is the computer asking "Who owns this IP?" before sending the actual Ping.

4. Header Analysis (The Anatomy of a Packet)

To understand the data, select one of the "Echo Request" packets in Wireshark. Look at the bottom pane where it creates a tree structure. Here is what you are looking at:

layer 2: Data Link Layer (Ethernet II)

This header handles local delivery between physical hardware.

- **Destination MAC:** The hardware address of the receiver (h2).
- **Source MAC:** The hardware address of the sender (h1).

- **Type:** 0x0800 (Tells the computer: "The payload inside me is IPv4").

Layer 3: Network Layer (IPv4)

This header handles routing across networks.

- **Version:** 4 (IPv4).
- **Time to Live (TTL):** usually 64 (Prevents packets from looping forever).
- **Protocol:** 1 (Tells the computer: "The payload inside me is ICMP").
- **Source IP:** 10.0.0.1
- **Destination IP:** 10.0.0.2

Layer 4 / Payload: ICMP (Internet Control Message Protocol)

While technically Layer 3 helper, in Wireshark it appears as the final payload for Ping.

- **Type:** 8 (Echo Request) or 0 (Echo Reply).
- **Code:** 0 (No specific code).
- **Checksum:** Validates that data wasn't corrupted.
- **Identifier & Sequence:** Helps the computer match the Reply to the specific Request sent.

5. Learning Outcomes

By completing this, you have verified:

1. **Encapsulation:** How ICMP sits inside IP, which sits inside Ethernet.
2. **Handshaking:** How ARP precedes IP communication.
3. **Analysis:** How to verify network events using Packet Capture (pcap) files.