TCP & UDP LAB

1. Lab Exercise - UDP

Capture a UDP Trace

There are many ways to cause your computer to send and receive UDP messages since UDP is widely used as a transport protocol.

1. Launch Wireshark by entering Wireshark in the “ask my anything” search box in Windows.

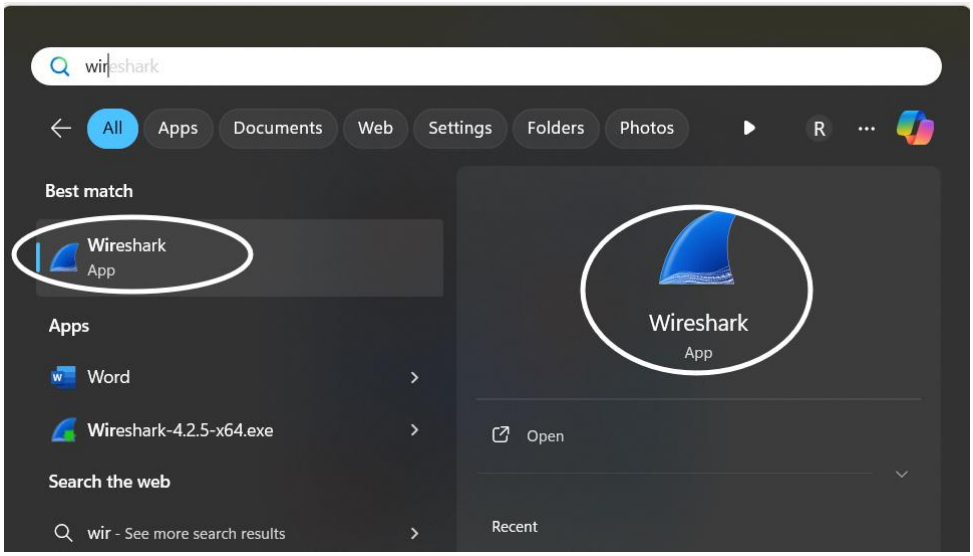


Figure 1: Starting Wireshark

2. Once Wireshark starts, select the WiFi/Ethernet interface.

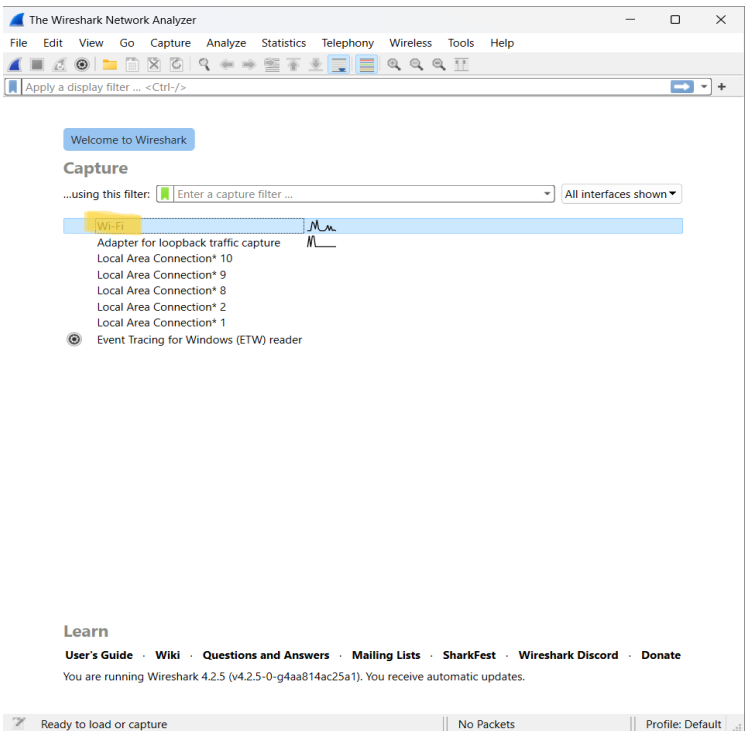


Figure 2: Selecting the WiFi/Ethernet Interface

3. Wireshark will automatically start capturing packets on the network.

Now, enter a filter of **udp**. (This is shown below).

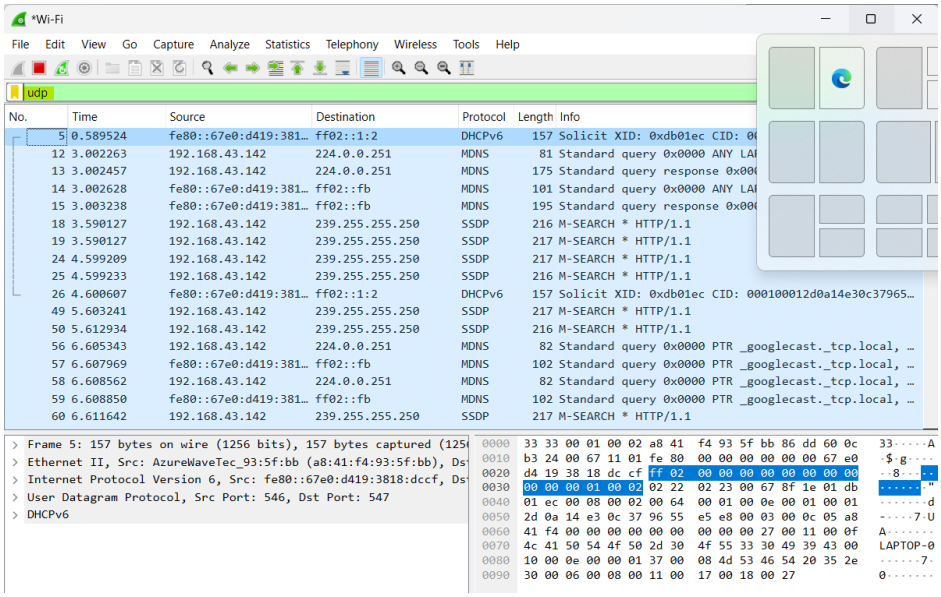


Figure 3: Setting up the capture options

1. When the capture is started, it will collect UDP traffic automatically.
2. Use the Wireshark menus or buttons to stop the capture.

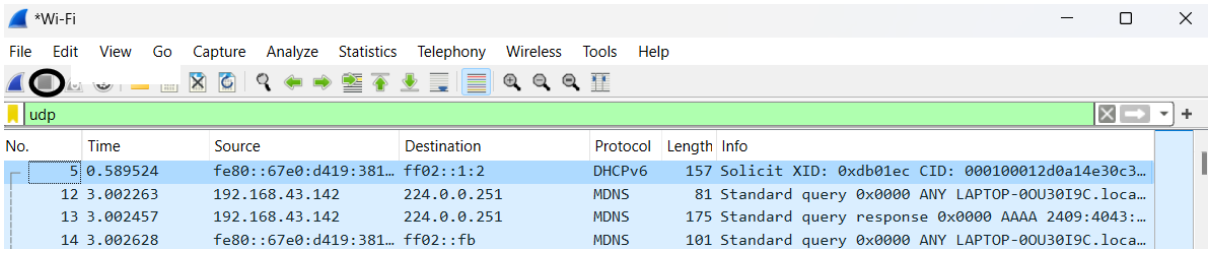


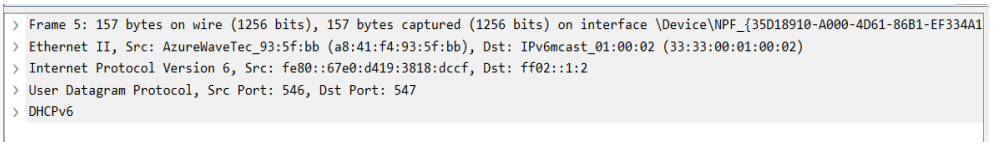
Figure 4: Stopping the capture

3. You should now have a trace with many UDP packets.

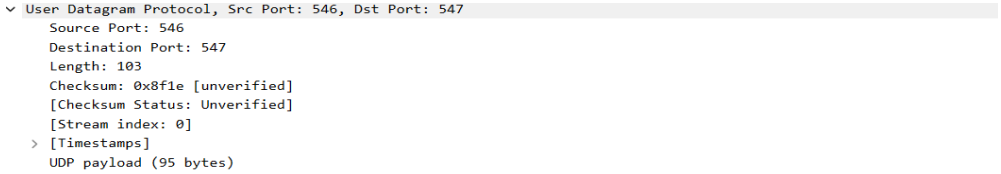
Inspect the Trace of UDP packets

Select different packets in the trace (in the top panel) and browse the expanded UDP header (in the middle panel). it contains the following fields:

- Source Port, Destination Port, Checksum and length of specific packet:



- Length and Checksum



UDP Message Structure-

The figure below shows the UDP message structure as you observed. It shows the position of the IP header, UDP header, and UDP payload.

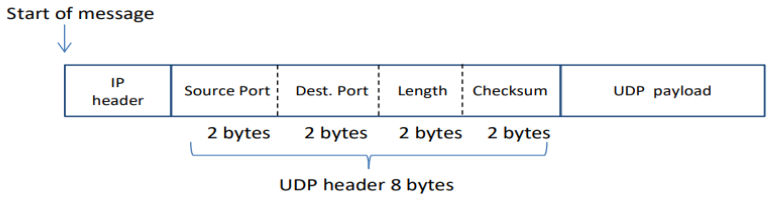


Figure 5: Structure of a UDP message

---

2. Lab Exercise - TCP

Open the TCP Trace

Open the trace file here: https://kevincurran.org/com320/labs/wireshark/trace-tcp.pcap

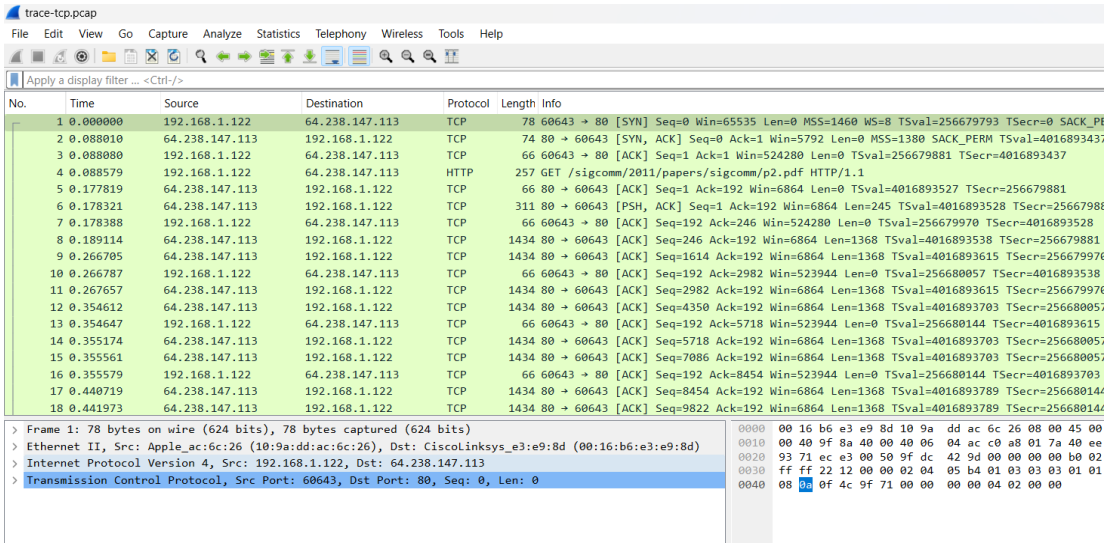


Figure 6: Selecting the Ethernet Interface

Inspect the Trace of TCP Packets

- **Frame length**

```
Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul 12, 2012 11:34:41.439558000 India Standard Time
    UTC Arrival Time: Jul 12, 2012 06:04:41.439558000 UTC
    Epoch Arrival Time: 1342073081.439558000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 78 bytes (624 bits)
```

TCP & UDP LAB

- **TCP Port:**

```
Transmission Control Protocol, Src Port: 60643, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 60643
    Destination Port: 80
    [Stream index: 0]
  > [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 2682012317
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
```

TCP Segment Structure

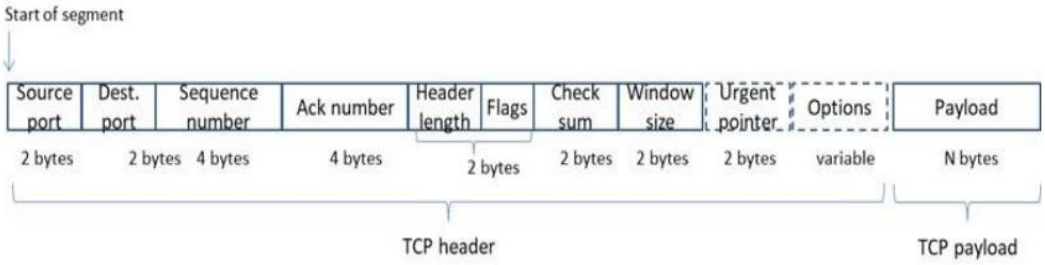
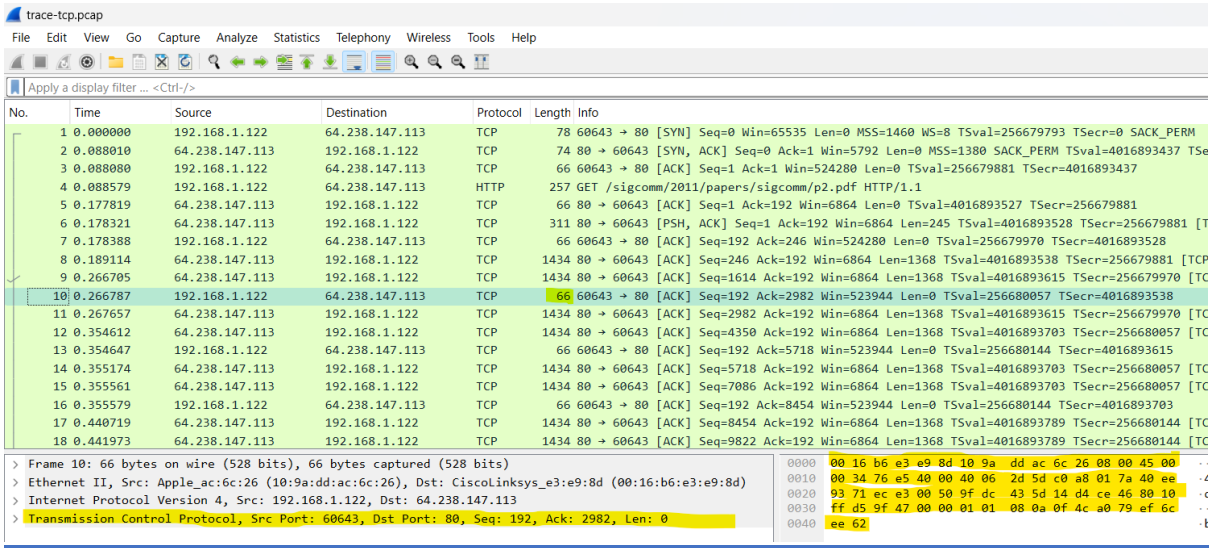


Figure 7: Structure of a TCP segment

Examining the size of segments

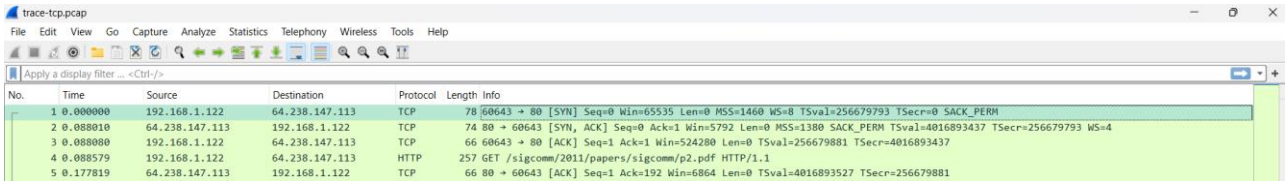


TCP Connection Setup/Teardown

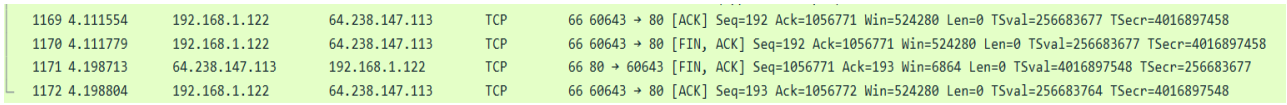
Three-Way Handshake

1.) Sending SYN and Receiving ACK for starting the connection:

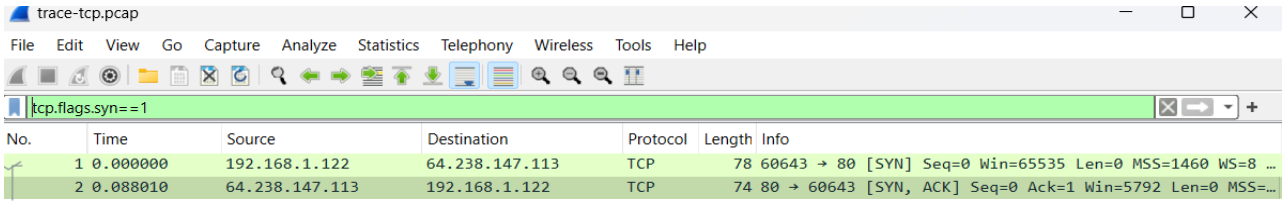
TCP & UDP LAB



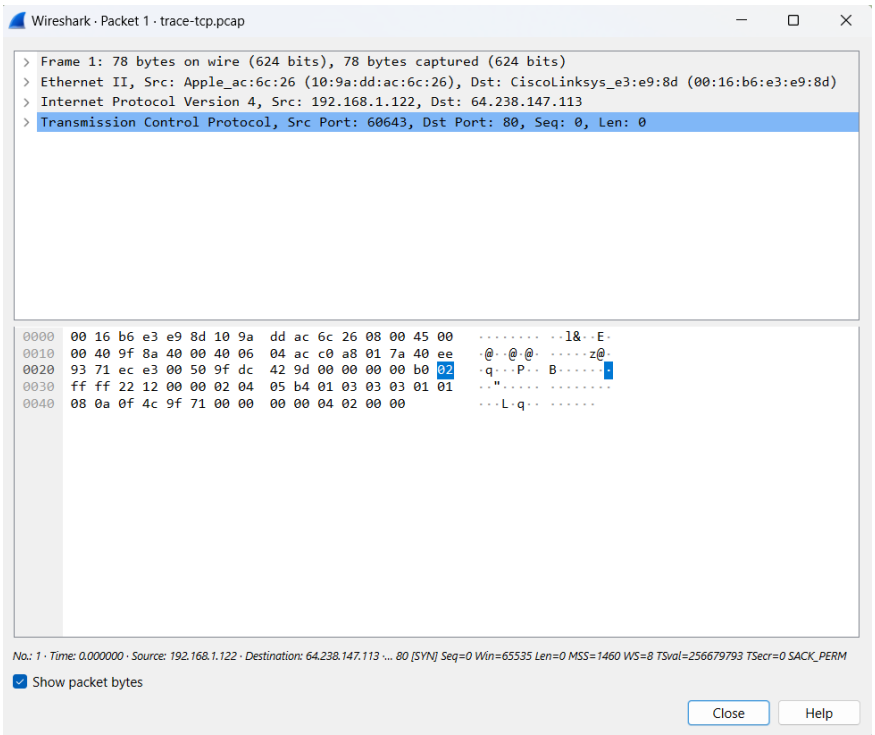
2.) Closing the connection with SYN and Acknowledge it:



3.) We can search packets with the SYN flag on using the filter expression “tcp.flags.syn==1”. (See below).



4.) On clicking on syn request, we get:

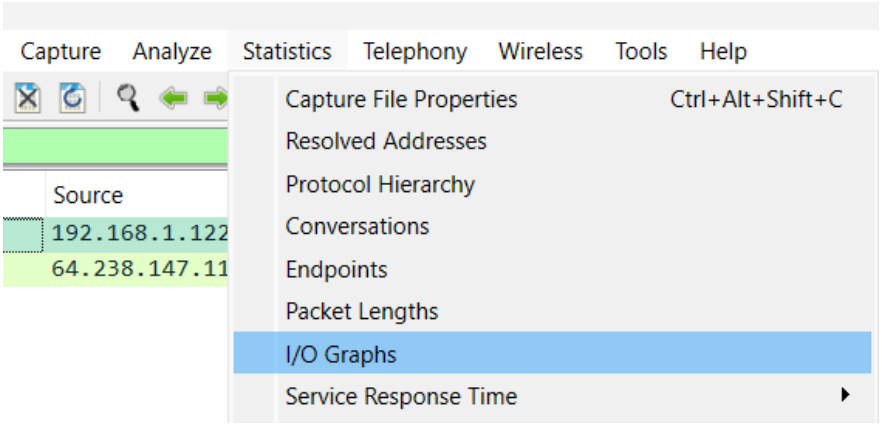


TCP Data Transfer:

IO Graph

Under the Statistics menu of wireshark select an “IO Graph” (as shown below).

TCP & UDP LAB



You should end up with a graph like below.