

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 me anything” search box in Windows.

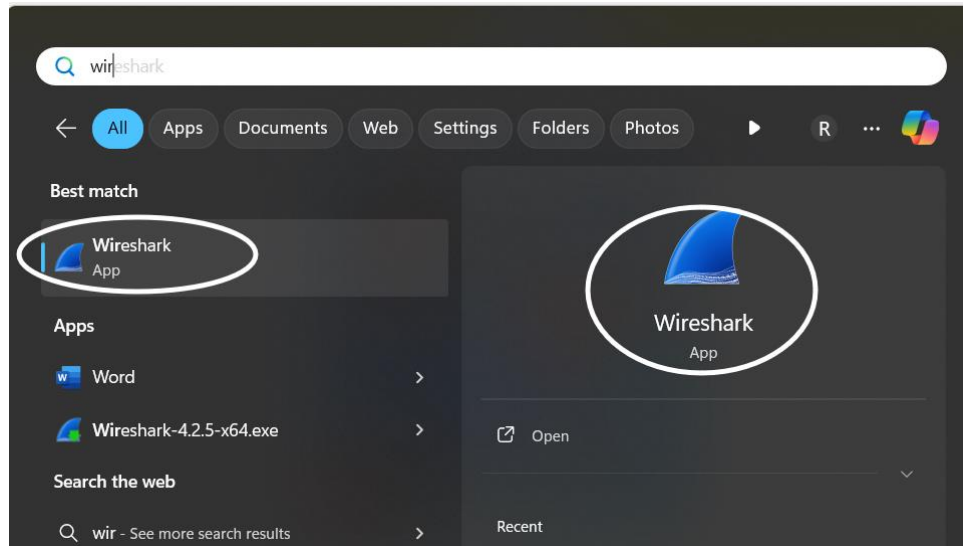


Figure 1: Starting Wireshark

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

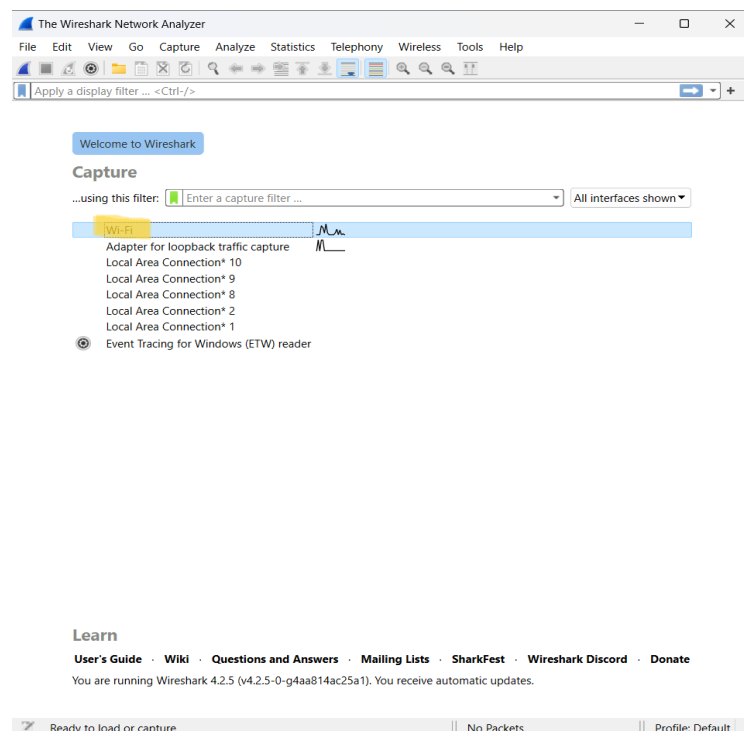


Figure 2: Selecting the WiFi/Ethernet Interface

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

TCP & UDP LAB

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

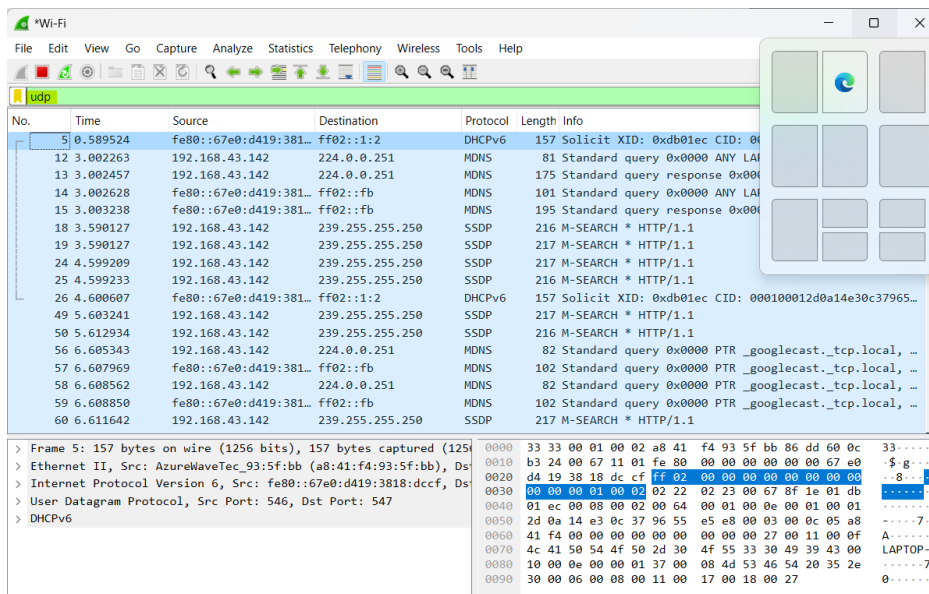


Figure 3: Setting up the capture options

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

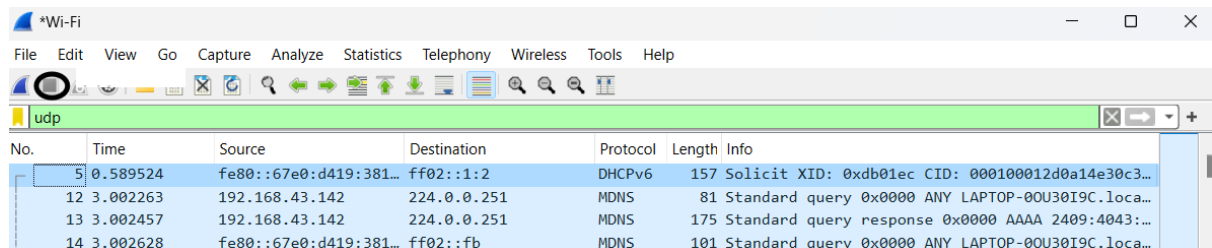


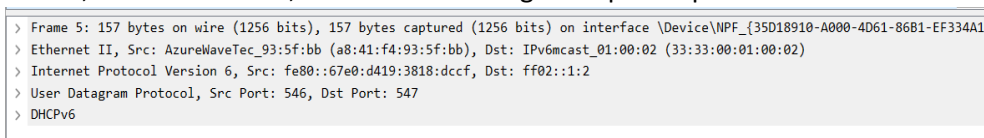
Figure 4: Stopping the capture

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

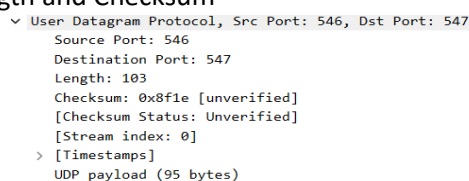
Inspect the Trace of UDP packets

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



TCP & UDP LAB

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.

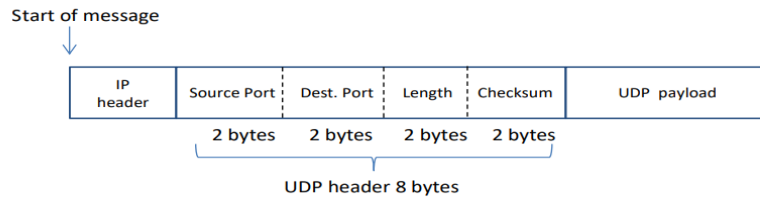


Figure 5: Structure of a UDP message

2. Lab Exercise - TCP

Open the TCP Trace

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

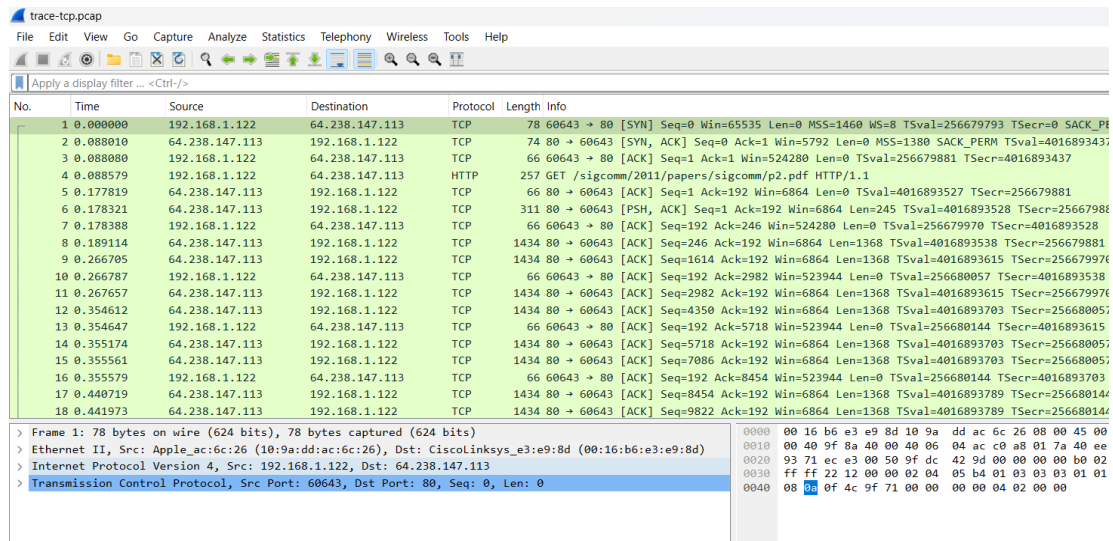


Figure 6: Selecting the Ethernet Interface

Inspect the Trace of TCP Packets

- Frame length

```
> Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on 0
  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

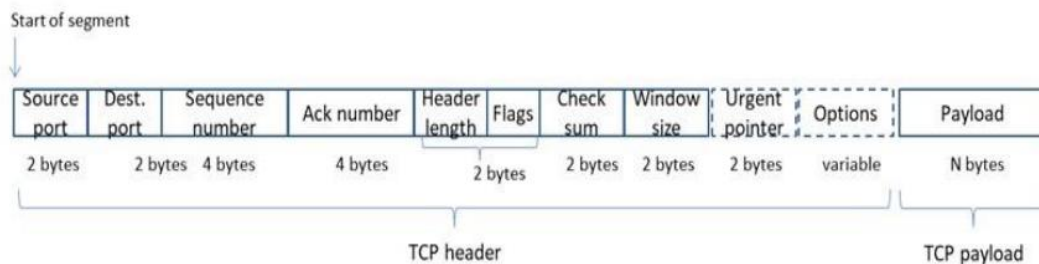


Figure 7: Structure of a TCP segment

Examining the size of segments

trace-tcp.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.122	64.238.147.113	TCP	78	60643 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=256679793 TSecr=0 SACK_PERM
2	0.088010	64.238.147.113	192.168.1.122	TCP	74	80 → 60643 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1380 SACK_PERM TSval=4016893437 TSe
3	0.088080	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=256679881 TSecr=4016893437
4	0.088579	192.168.1.122	64.238.147.113	HTTP	257	GET /sigcomm/2011/papers/sigcomm/p2.pdf HTTP/1.1
5	0.177819	64.238.147.113	192.168.1.122	TCP	66	80 → 60643 [ACK] Seq=1 Ack=192 Win=6864 Len=0 TSval=4016893527 TSecr=256679881
6	0.178321	64.238.147.113	192.168.1.122	TCP	311	80 → 60643 [PSH, ACK] Seq=1 Ack=192 Win=6864 Len=245 TSval=4016893528 TSecr=256679881 [T
7	0.178388	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=192 Ack=246 Win=524280 Len=0 TSval=256679970 TSecr=4016893528
8	0.189114	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=246 Ack=192 Win=6864 Len=1368 TSval=4016893538 TSecr=256679881 [TCP
9	0.266705	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=1614 Ack=192 Win=6864 Len=1368 TSval=4016893615 TSecr=256679970 [TC
10	0.266787	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=192 Ack=2982 Win=523944 Len=0 TSval=256680057 TSecr=4016893538
11	0.267657	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=2982 Ack=192 Win=6864 Len=1368 TSval=4016893615 TSecr=256679970 [TC
12	0.354612	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=4350 Ack=192 Win=6864 Len=1368 TSval=4016893703 TSecr=256680057 [TC
13	0.354647	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=192 Ack=5718 Win=523944 Len=0 TSval=256680144 TSecr=4016893615
14	0.355174	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=5718 Ack=192 Win=6864 Len=1368 TSval=4016893703 TSecr=256680057 [TC
15	0.355561	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=7086 Ack=192 Win=6864 Len=1368 TSval=4016893703 TSecr=256680057 [TC
16	0.355579	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=192 Ack=8454 Win=523944 Len=0 TSval=256680144 TSecr=4016893703
17	0.440719	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=8454 Ack=192 Win=6864 Len=1368 TSval=4016893789 TSecr=256680144 [TC
18	0.441973	64.238.147.113	192.168.1.122	TCP	1434	80 → 60643 [ACK] Seq=9822 Ack=192 Win=6864 Len=1368 TSval=4016893789 TSecr=256680144 [TC

> Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

> Ethernet II, Src: Apple_ac:6c:26 (10:9a:dd:ac:6c:26), Dst: CiscoLinksys_e3:e9:8d (00:16:b6:e3:e9:8d)

> Internet Protocol Version 4, Src: 192.168.1.122, Dst: 64.238.147.113

> Transmission Control Protocol, Src Port: 60643, Dst Port: 80, Seq: 192, Ack: 2982, Len: 0

0000 00 16 b6 e3 e9 8d 10 9a dd ac 6c 26 08 00 45 00 ..

0010 00 34 76 e5 40 00 06 2d 5d c0 a8 01 7a 40 ee ..

0020 93 71 ec e3 00 50 9f dc 43 5d 14 d4 ce 46 80 10 ..

0030 ff d5 9f 47 00 00 01 01 08 0a 0f 4c a0 79 ef 6c ..

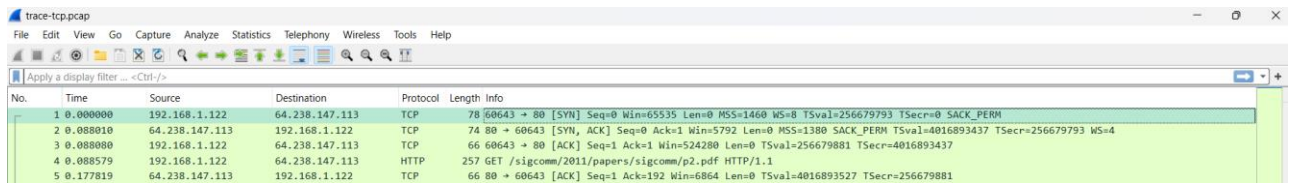
0040 ee 62 ..

TCP Connection Setup/Teardown

Three-Way Handshake

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

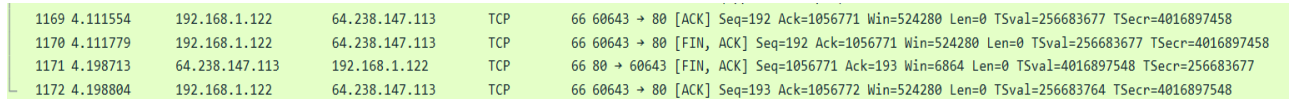
TCP & UDP LAB



trace-tcp.pcap

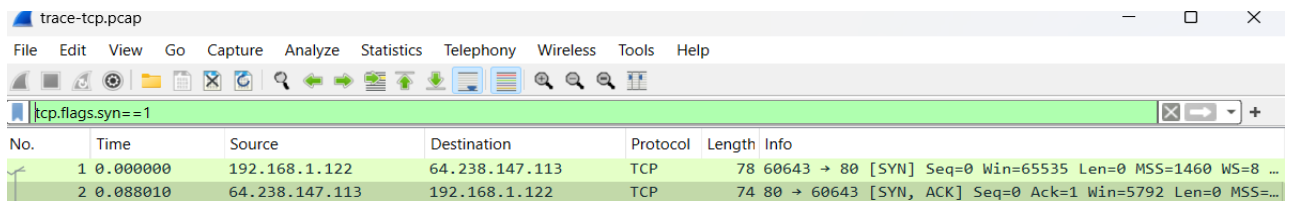
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.122	64.238.147.113	TCP	78	60643 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=256679793 TSecr=0 SACK_PERM
2	0.008010	64.238.147.113	192.168.1.122	TCP	74	80 → 60643 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1380 SACK_PERM TSval=4016893437 TSecr=256679793 WS=4
3	0.008080	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=256679881 TSecr=4016893437
4	0.008579	192.168.1.122	64.238.147.113	HTTP	257	GET /sigcomm/2011/papers/sigcomm/p2.pdf HTTP/1.1
5	0.177819	64.238.147.113	192.168.1.122	TCP	66	80 → 60643 [ACK] Seq=1 Ack=192 Win=6864 Len=0 TSval=4016893527 TSecr=256679881

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



No.	Time	Source	Destination	Protocol	Length	Info
1169	4.111554	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=192 Ack=1056771 Win=524280 Len=0 TSval=256683677 TSecr=4016897458
1170	4.111779	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [FIN, ACK] Seq=192 Ack=1056771 Win=524280 Len=0 TSval=256683677 TSecr=4016897458
1171	4.198713	64.238.147.113	192.168.1.122	TCP	66	80 → 60643 [FIN, ACK] Seq=1056771 Ack=193 Win=6864 Len=0 TSval=4016897548 TSecr=256683677
1172	4.198804	192.168.1.122	64.238.147.113	TCP	66	60643 → 80 [ACK] Seq=193 Ack=1056772 Win=524280 Len=0 TSval=256683764 TSecr=4016897548

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

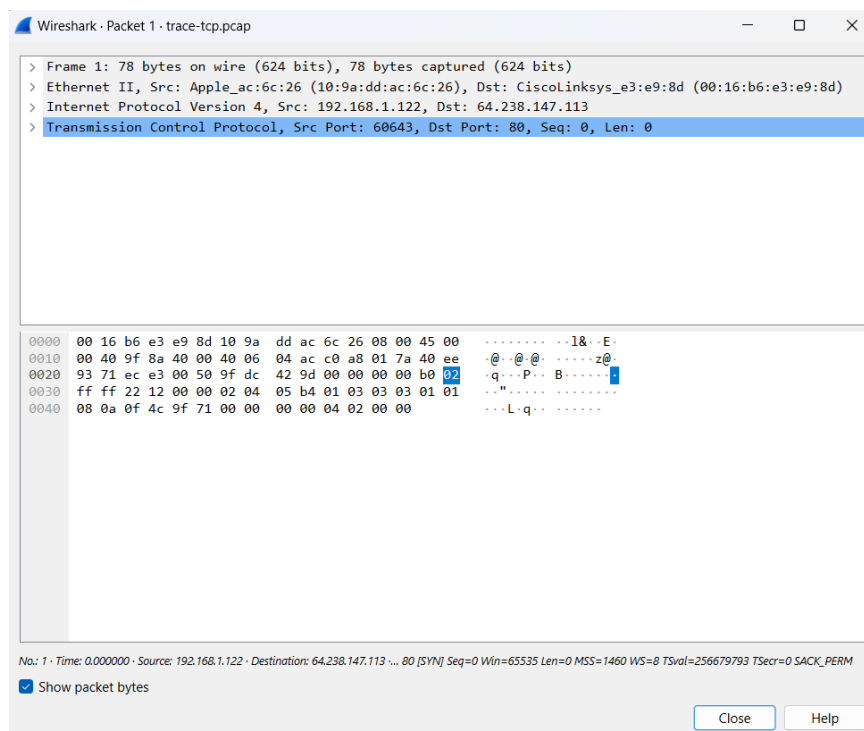


trace-tcp.pcap

Filter: tcp.flags.syn==1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.122	64.238.147.113	TCP	78	60643 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 ...
2	0.008010	64.238.147.113	192.168.1.122	TCP	74	80 → 60643 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=...

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



Wireshark · Packet 1 · trace-tcp.pcap

> Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)

> Ethernet II, Src: Apple_ac:6c:26 (10:9a:dd:ac:6c:26), Dst: CiscoLinksys_e3:e9:8d (00:16:b6:e3:e9:8d)

> Internet Protocol Version 4, Src: 192.168.1.122, Dst: 64.238.147.113

> Transmission Control Protocol, Src Port: 60643, Dst Port: 80, Seq: 0, Len: 0

0000 00 16 b6 e3 e9 8d 10 9a dd ac 6c 26 08 00 45 00 16.E.
0010 00 40 9f 8a 40 00 40 06 04 ac c0 a8 01 7a 40 ee .@..@..z@
0020 93 71 ec e3 00 50 9f dc 42 9d 00 00 00 b0 02 .q..P..B..
0030 ff ff 22 12 00 00 02 04 05 b4 01 03 03 01 01 ..".....
0040 08 0a 0f 4c 9f 71 00 00 00 00 04 02 00 00 ...L.q..

No.: 1 · Time: 0.000000 · Source: 192.168.1.122 · Destination: 64.238.147.113 ... 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=256679793 TSecr=0 SACK_PERM

☒ Show packet bytes

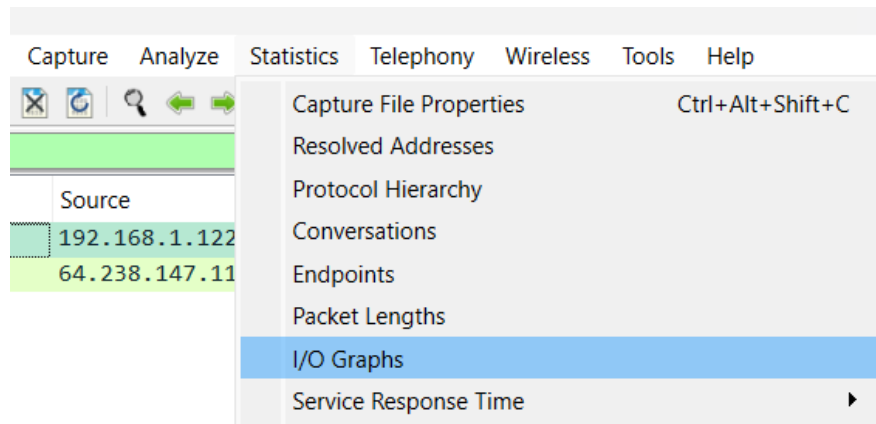
Close Help

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.

