Simple Protocols Traffic Analysis

Kevin Scrivnor

COMP 429

Spring 2023

Recall: A network protocol defines how these devices talk to each other

- The design goals behind a protocol are
- Reliability
  - Recover from errors, failures
- Resource allocation
  - Sharing access to common, limited resources
- Evolvability
  - Allow of change over time
- Security
  - Defending the network against various attacks

What does a protocol define?

- Types of messages exchanged
  - (what kind of messages)
  - **Consider what information needs to be transferred.**
- Syntax of the various types of messages
  - (format of messages)
  - **Text based or Binary? Line endings?**
- Semantics of the fields in the messages
  - (meaning of messages)
  - **Define and document the meaning of messages**
- and rules for when messages are exchanged
  - (sending and responding)
  - **Describe actions to be taken, define error codes/messages, think about everything that can go wrong.**

Consider a file transferring networked program

- Types of messages exchanged
  - **What are we transferring?**
- Syntax of the various types of messages
  - **Binary based or text based?**
- Semantics of the fields in the messages
  - **What is the meaning behind our messages?**
- and rules for when messages are exchanged
  - **What is the sender supposed to be doing?**
  - **What is the receiver supposed to be doing?**

Where are standard protocols defined?

- There is a long process to becoming an Internet Standard
- Ultimately ends up in an RFC (Request For Comments)

- For instance, the Trivial File Transfer Protocol (Revision 2) is defined in RFC 1350
  - https://datatracker.ietf.org/doc/html/rfc1350
- Sometimes, features get added later. Such as the blksize option:
  - https://datatracker.ietf.org/doc/html/rfc2348

- Reading an RFC is like reading a very detailed and long lab assignment. Every little sentence can make a huge difference in the implementation.

Wireshark

Solving Networking Problems is Hard

- Do I have an IP?
- Do I have connectivity on the network?
- Is my local firewall blocking traffic?
- Is the LAN firewall blocking traffic?
- Is DNS not resolving domain names?
- Am I routing packets correctly?
- Is the router routing packets correctly?
- Is the problem on my end or your end?
- WHAT THE HELL IS HAPPENING??
- *You can only solve a problem if you can see it*

Packet Analysis – who looks at the packets?

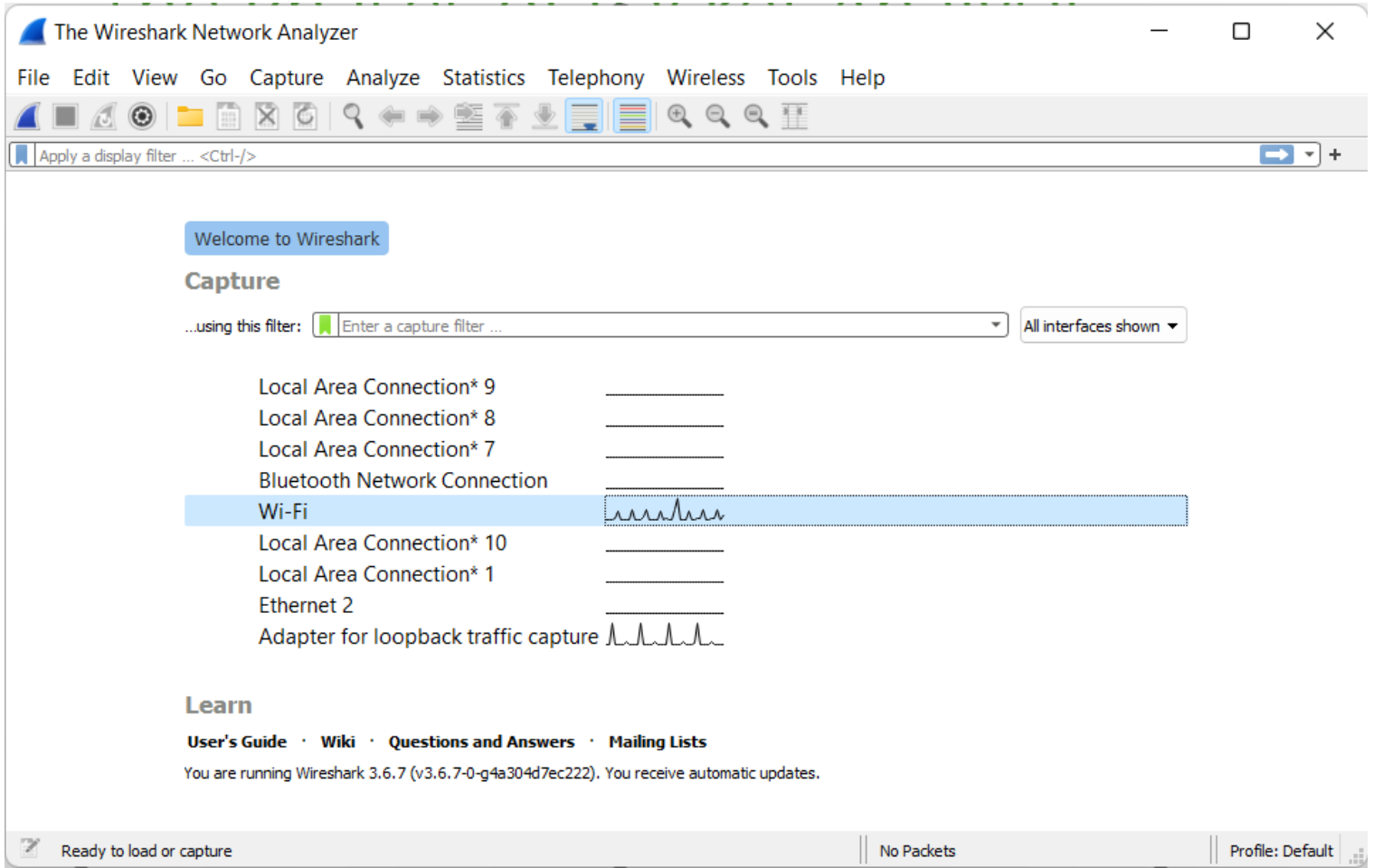
- Packet Capture files, or PCAP files contain data captured from some network interface

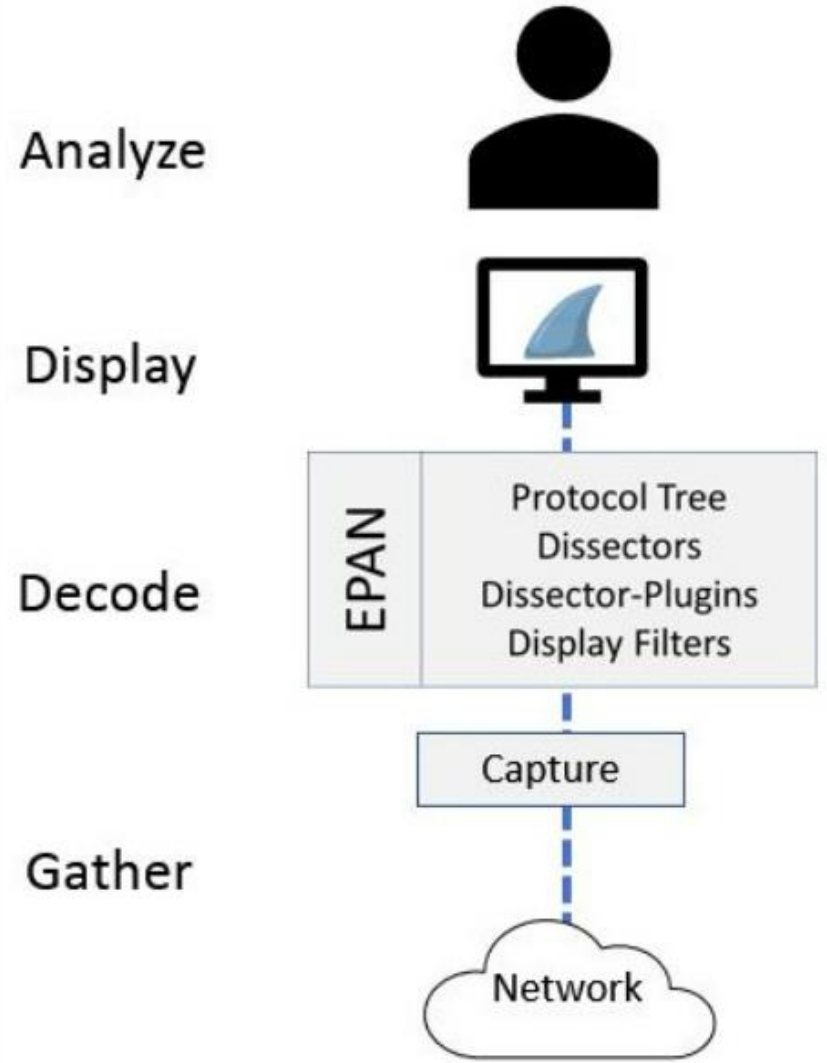
- Network admins – gain information about current conditions
- Security analysts – anything suspicious, forensics
- Students – learning tool to see how networking works
- Hackers – sniffing network traffic for enumeration

- Usually referred to as “sniffing”
  - As early packet tools sniffed the network for packets

The Phases of Packet Analysis: Gather

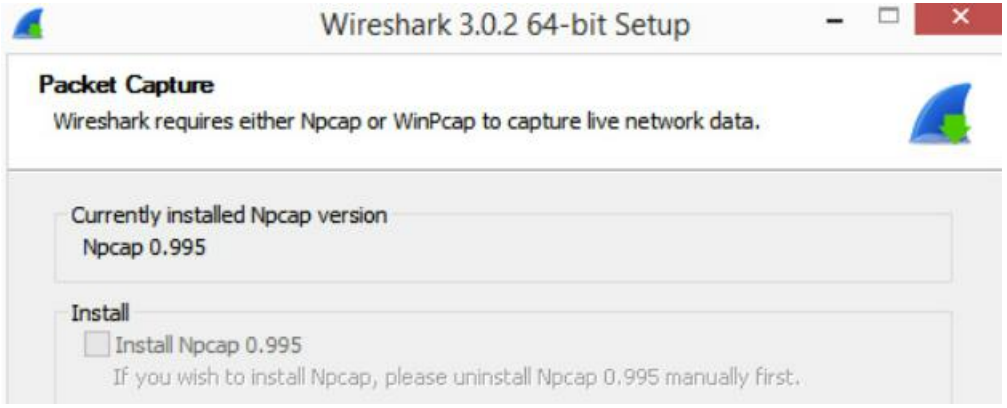
- Gather/capture packets
- Select the network interface you want to capture on

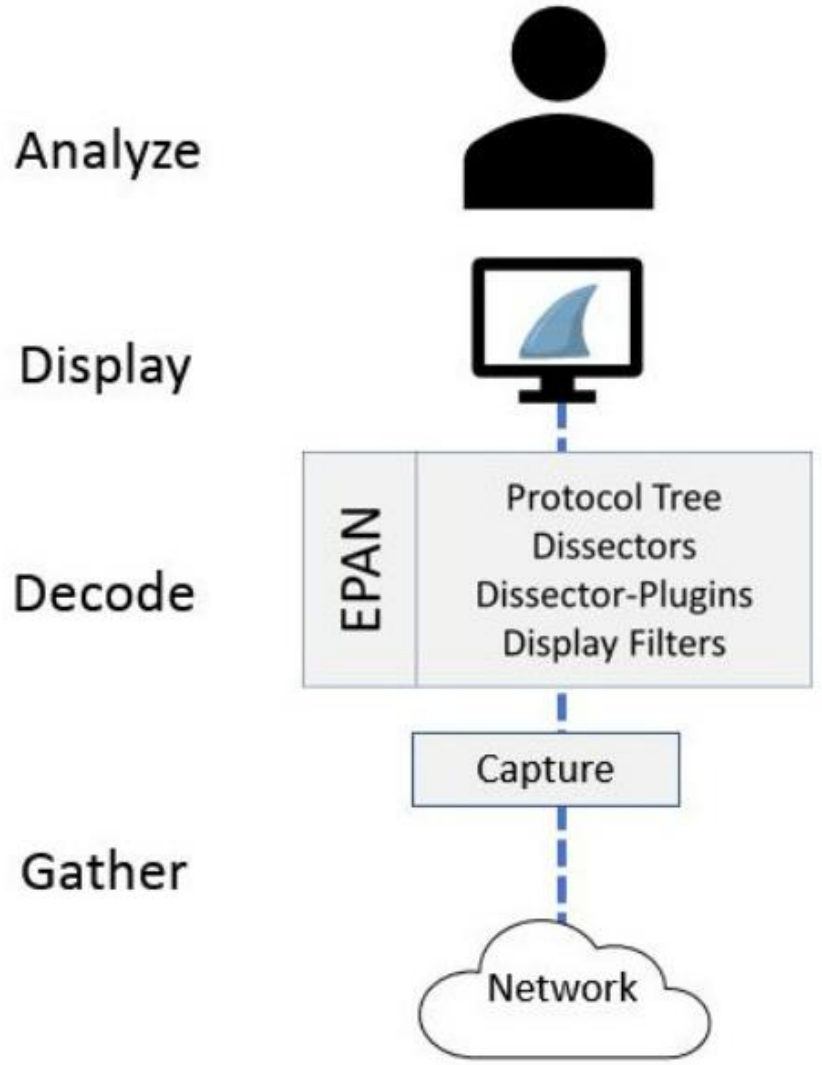




The Phases of Packet Analysis: Capture

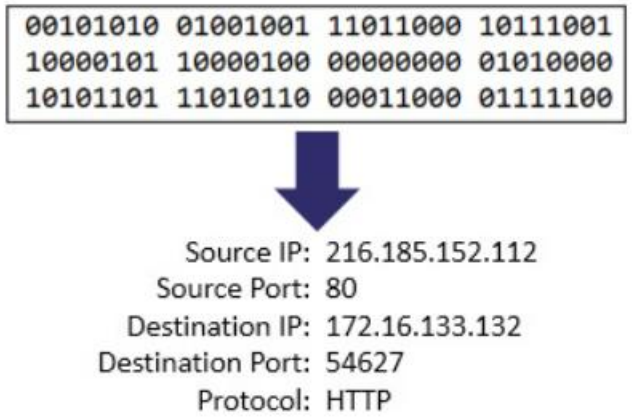
- Wireshark uses a capture engine: libnpcap
- This was installed with Wireshark

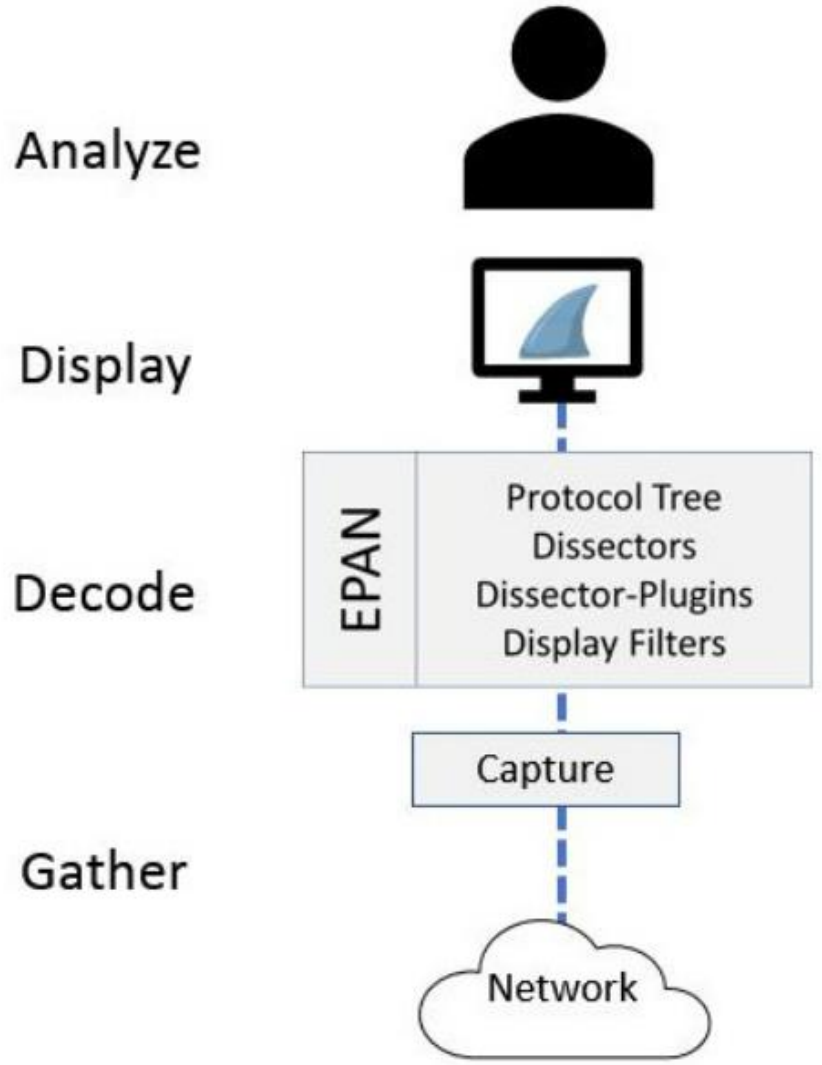




The Phases of Packet Analysis: Decode

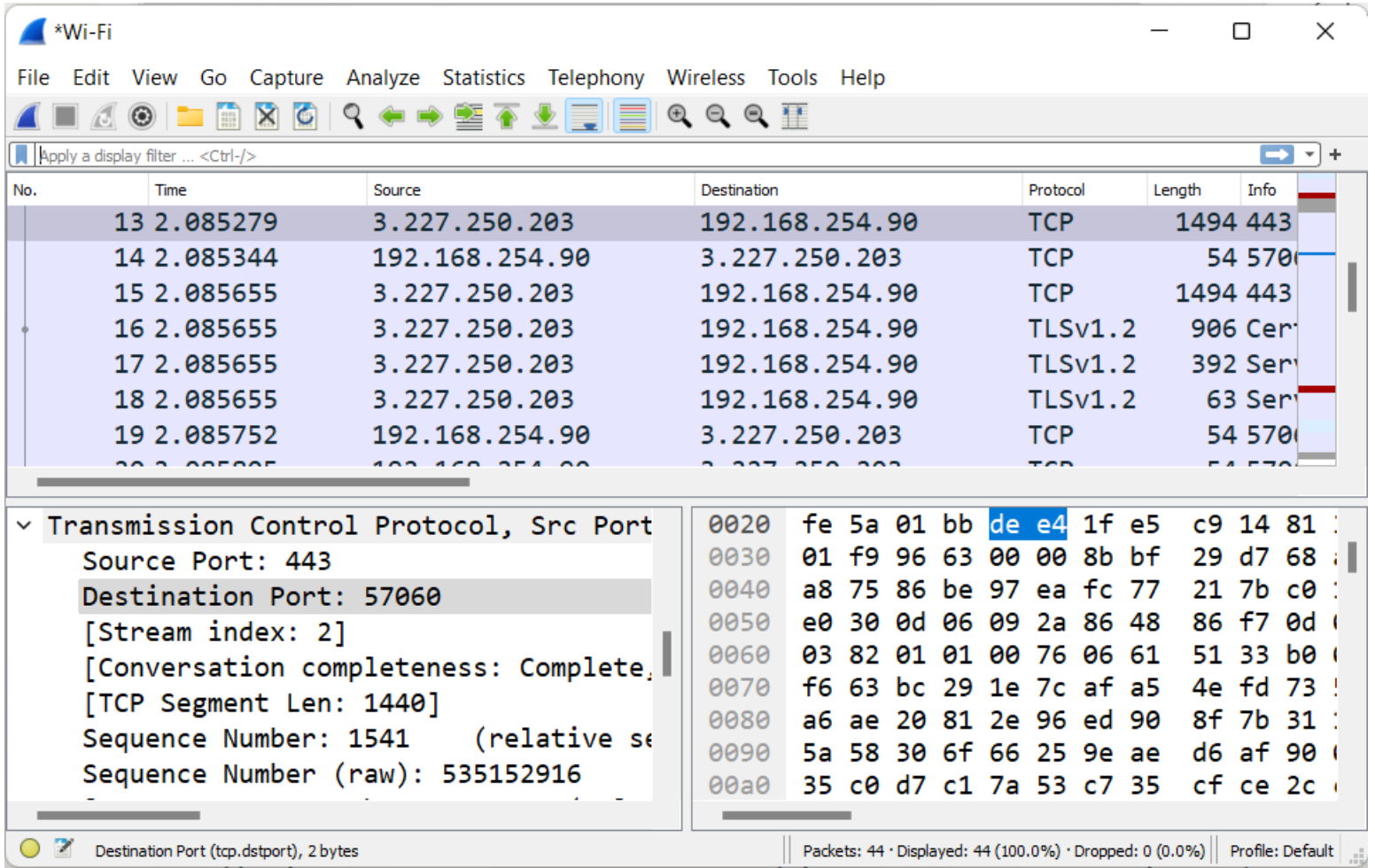
- The capture engine captures the raw bits of traffic going through the selected interface
- Wireshark then decodes the bits into human readable information

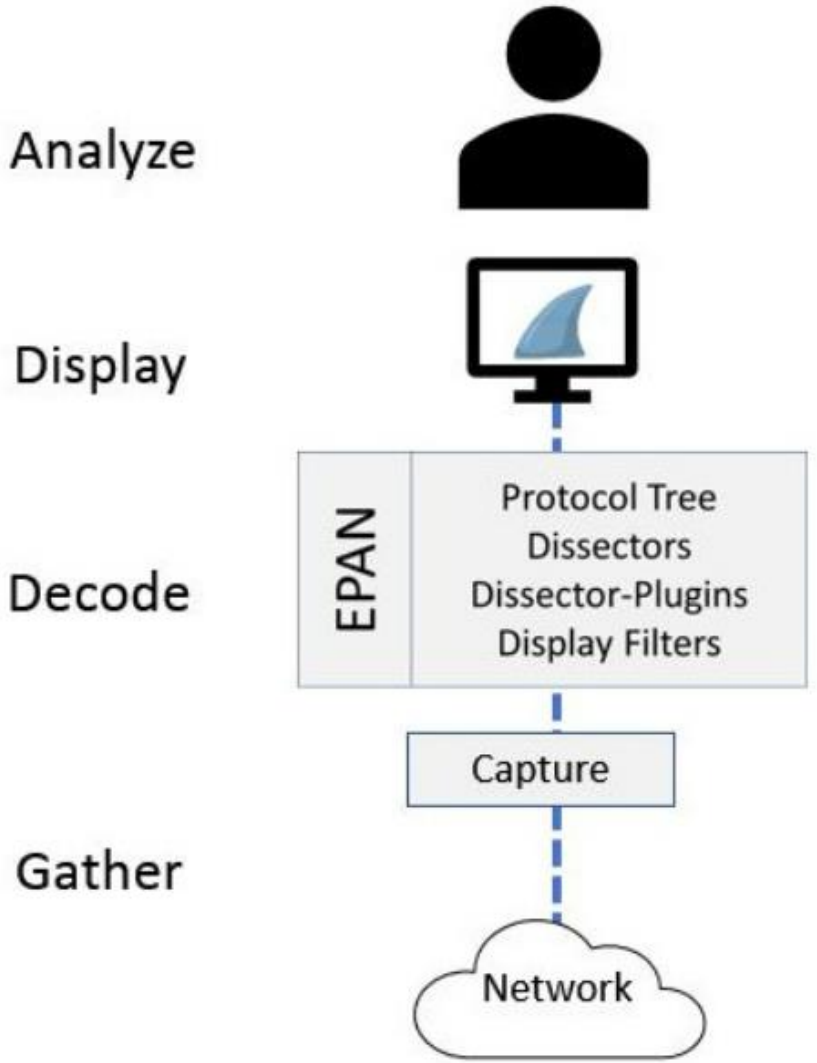




The Phases of Packet Analysis: Display

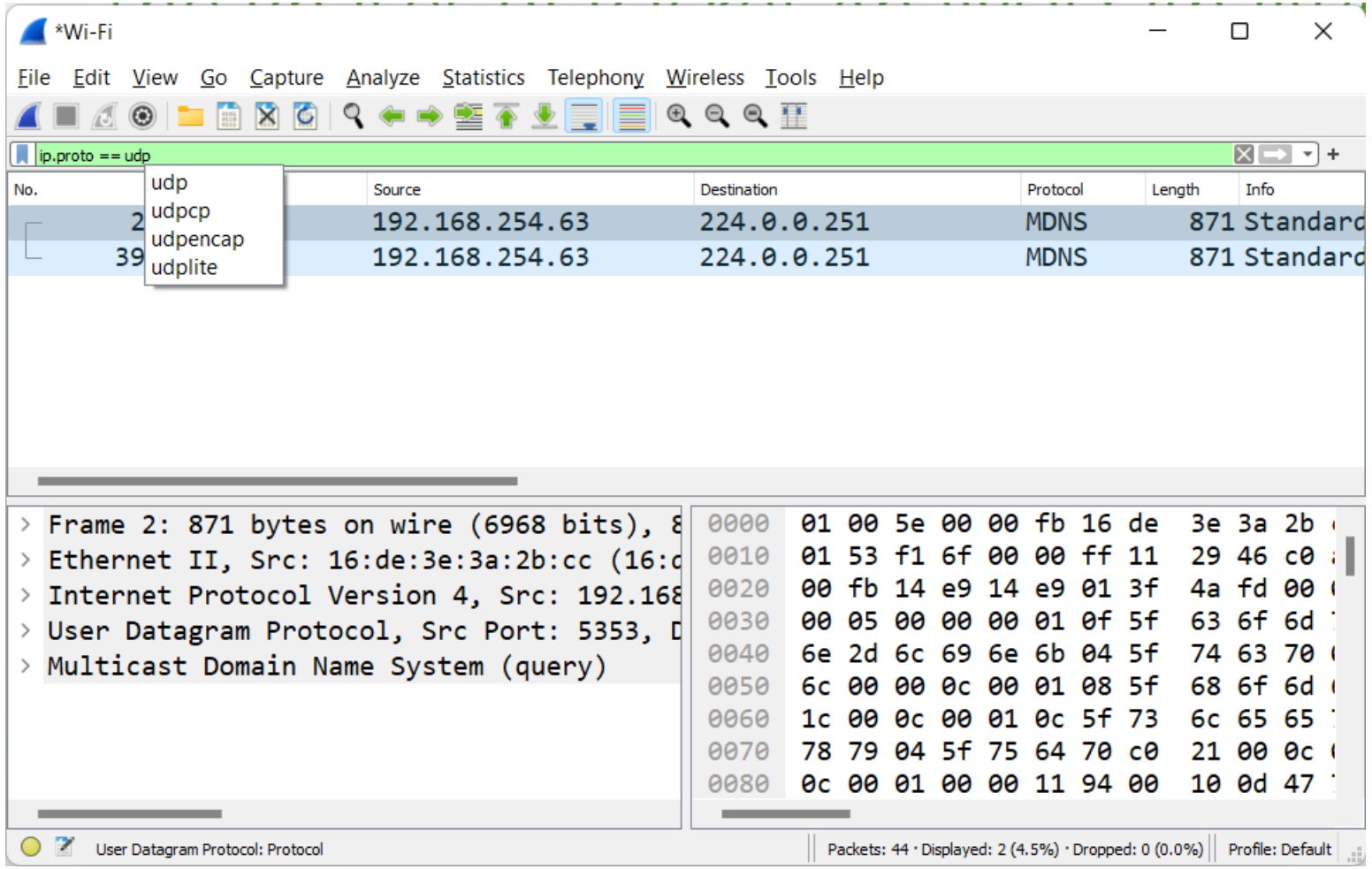
- Packet List, top
- Packet Details, left Packet Bytes, right

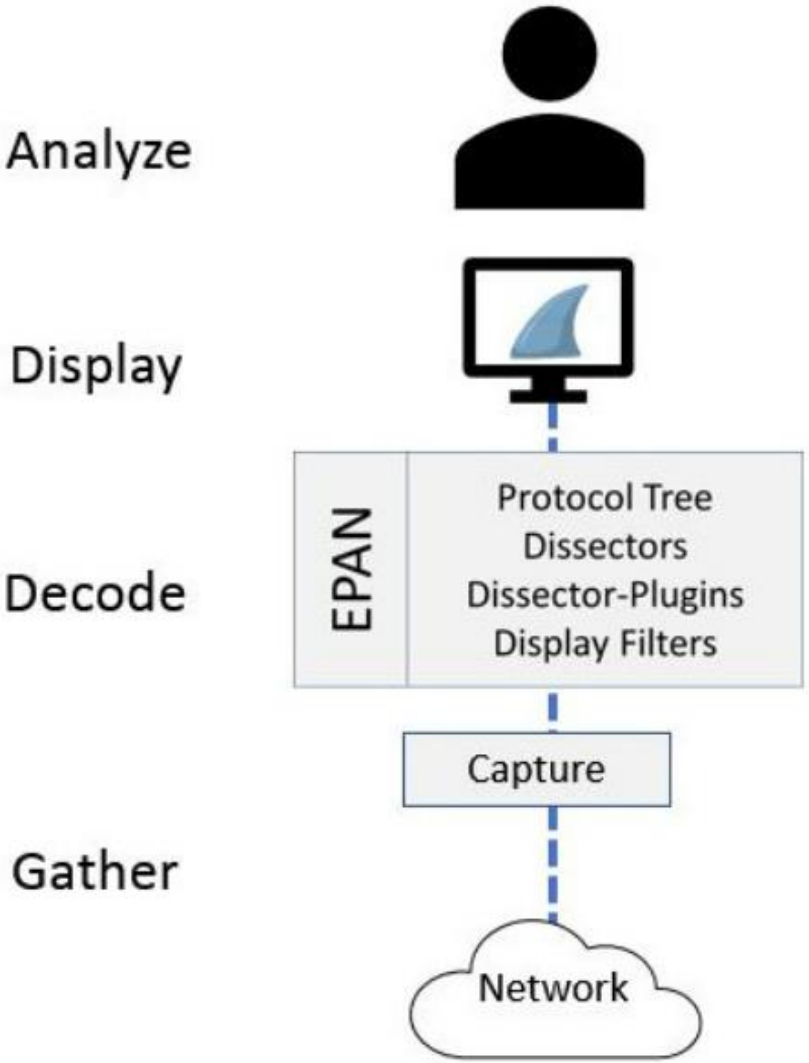




The Phases of Packet Analysis: Analyze

- Apply a display filter to look for specific packets



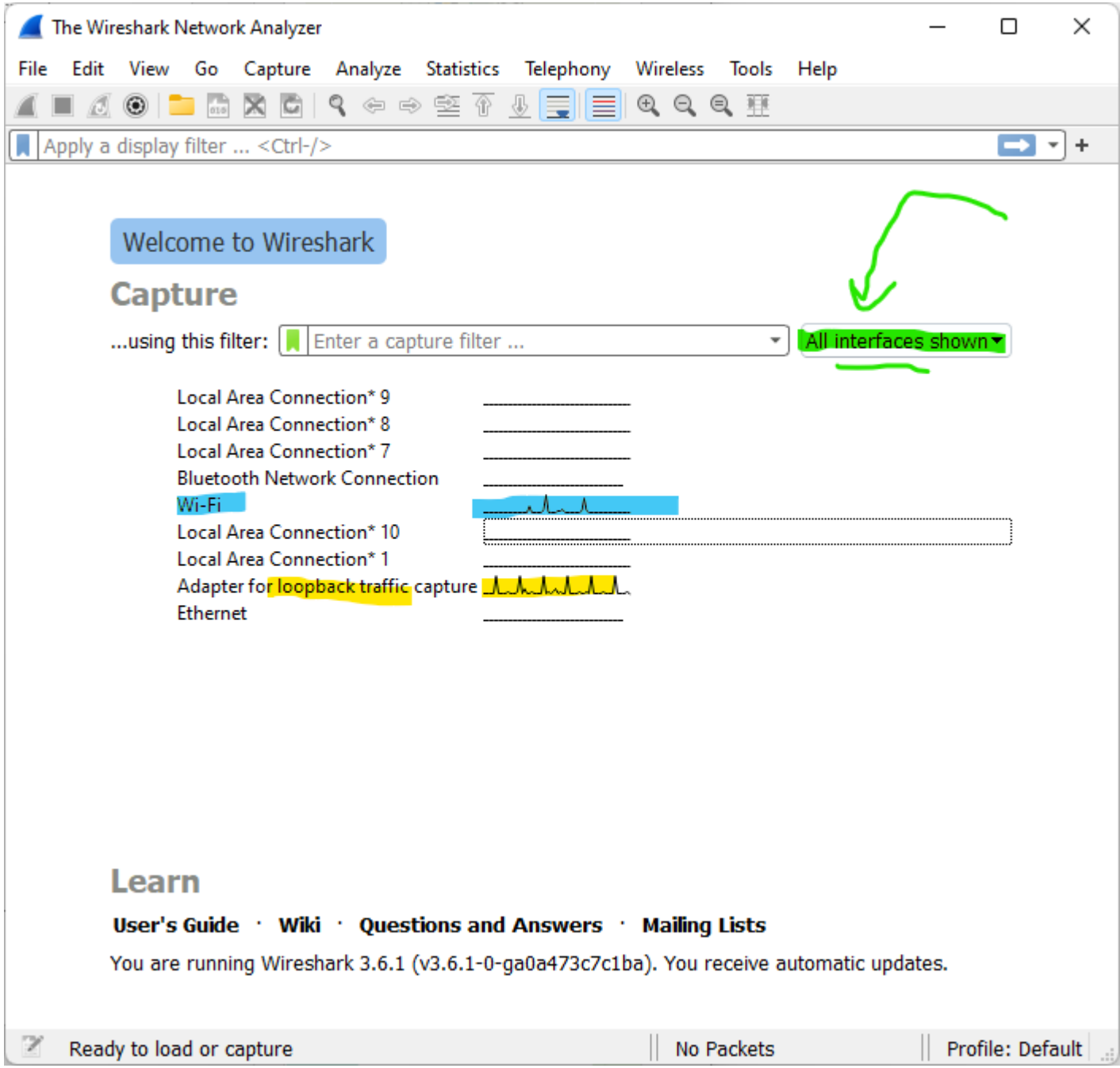


First View of Wireshark

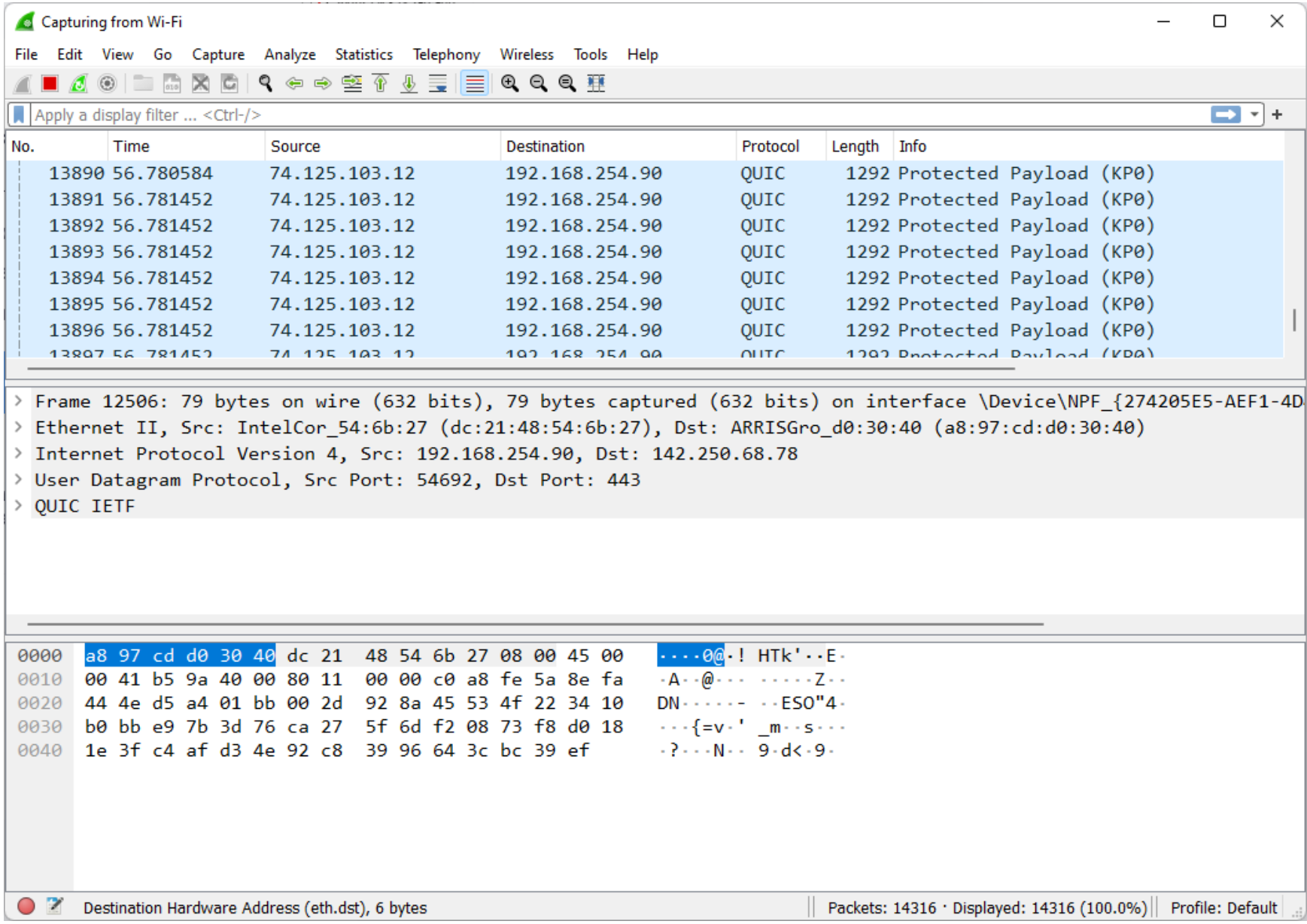
- You can limit the interfaces with choices of wired/wireless/Bluetooth as you see fit.

- My computer is on wifi, so I'll be capturing network traffic from this interface.

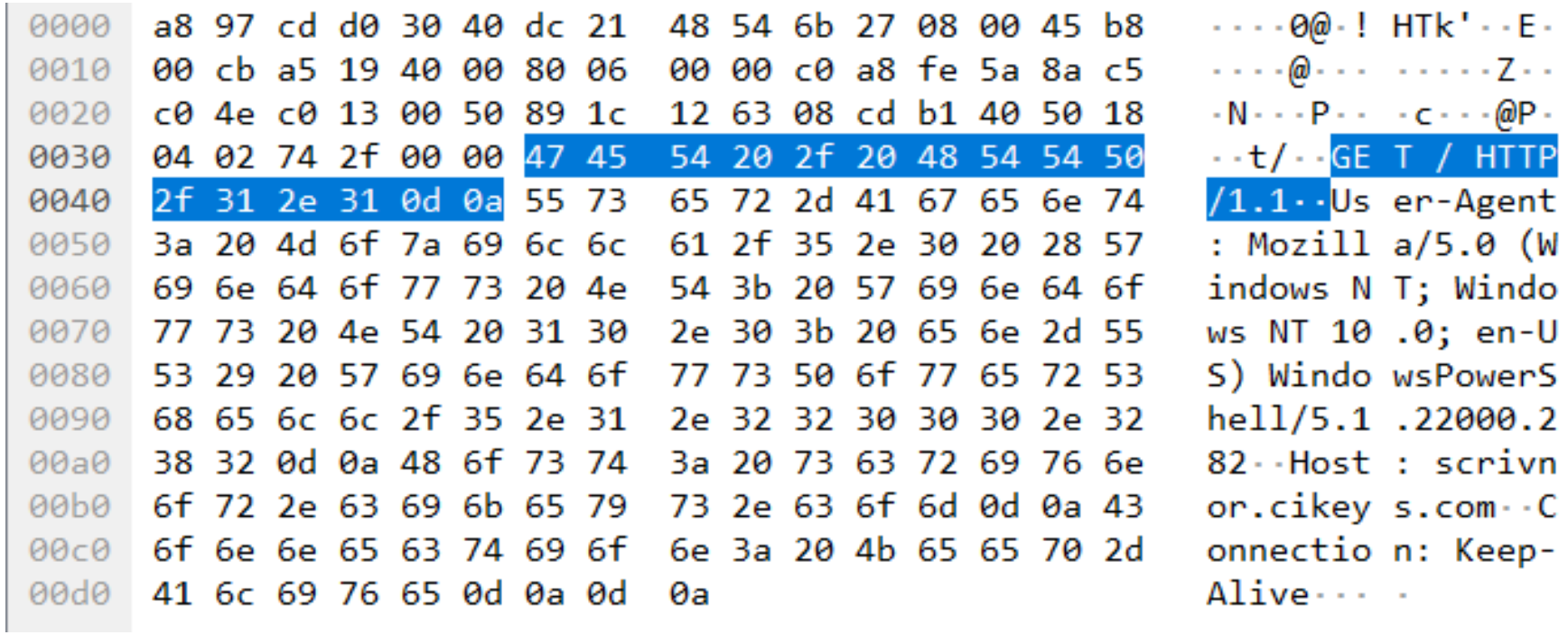
- The loopback is traffic from my computer to my computer. I'm not sure what Windows 11 is doing with all this local traffic…



https://www.wireshark.org/docs/wsug_html_chunked/ChUseMainWindowSection.html

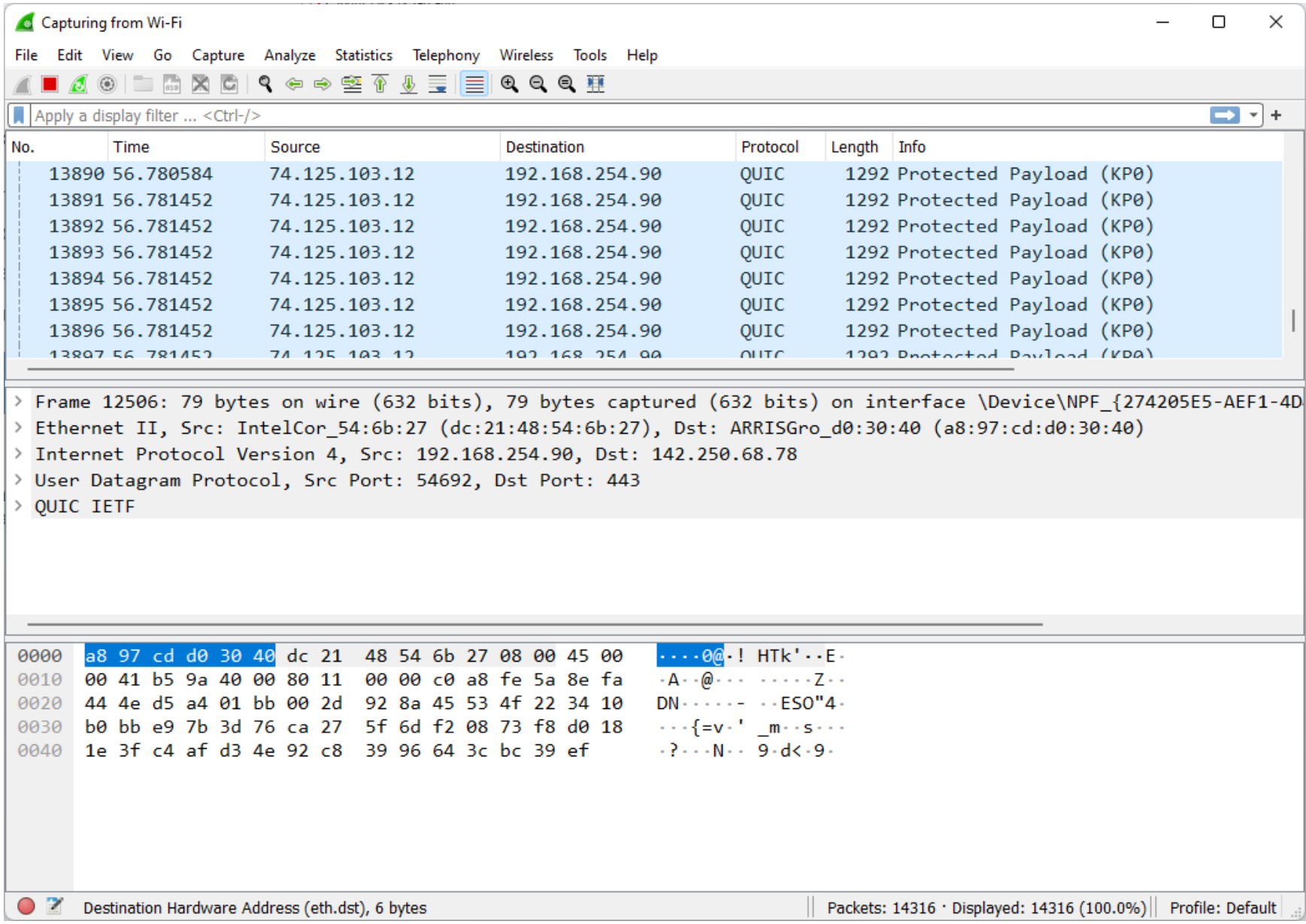


Reading Hex

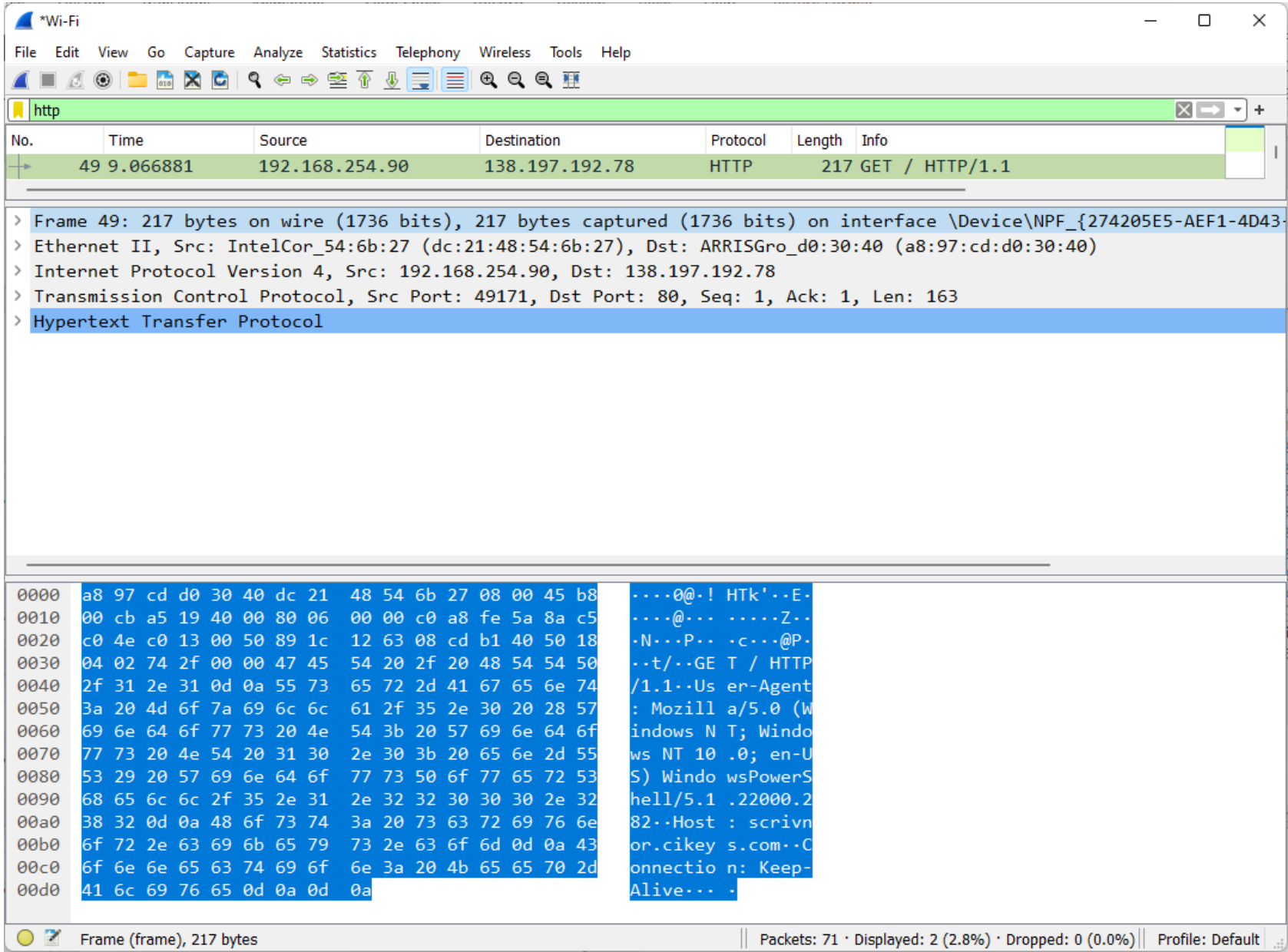


```
0000  a8 97 cd d0 30 40 dc 21  48 54 6b 27 08 00 45 b8   ····0@·! HTk'··E·
0010  00 cb a5 19 40 00 80 06  00 00 c0 a8 fe 5a 8a c5   ····@··· ·····Z··
0020  c0 4e c0 13 00 50 89 1c  12 63 08 cd b1 40 50 18   ·N···P·· ·c···@P·
0030  04 02 74 2f 00 00 47 45  54 20 2f 20 48 54 54 50   ··t/··GE T / HTTP
0040  2f 31 2e 31 0d 0a 55 73  65 72 2d 41 67 65 6e 74   /1.1··Us er-Agent
0050  3a 20 4d 6f 7a 69 6c 6c  61 2f 35 2e 30 20 28 57   : Mozill a/5.0 (W
0060  69 6e 64 6f 77 73 20 4e  54 3b 20 57 69 6e 64 6f   indows N T; Windo
0070  77 73 20 4e 54 20 31 30  2e 30 3b 20 65 6e 2d 55   ws NT 10 .0; en-U
0080  53 29 20 57 69 6e 64 6f  77 73 50 6f 77 65 72 53   S) Windo wsPowerS
0090  68 65 6c 6c 2f 35 2e 31  2e 32 32 30 30 30 2e 32   hell/5.1 .22000.2
00a0  38 32 0d 0a 48 6f 73 74  3a 20 73 63 72 69 76 6e   82··Host : scrivn
00b0  6f 72 2e 63 69 6b 65 79  73 2e 63 6f 6d 0d 0a 43   or.cikey s.com··C
00c0  6f 6e 6e 65 63 74 69 6f  6e 3a 20 4b 65 65 70 2d   onnectio n: Keep-
00d0  41 6c 69 76 65 0d 0a 0d  0a                       Alive··· ·
```

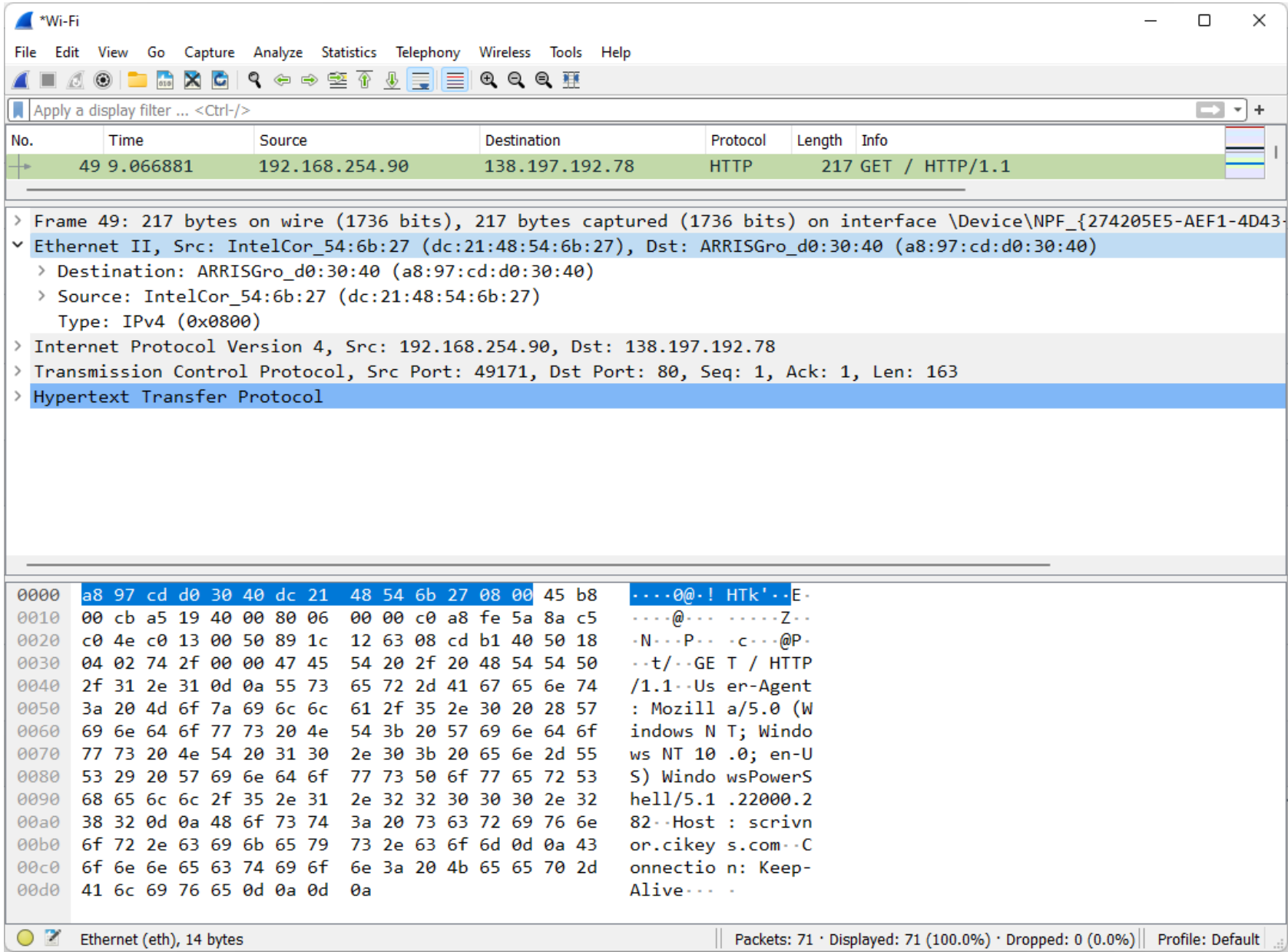
The 5 Layer Model in Action



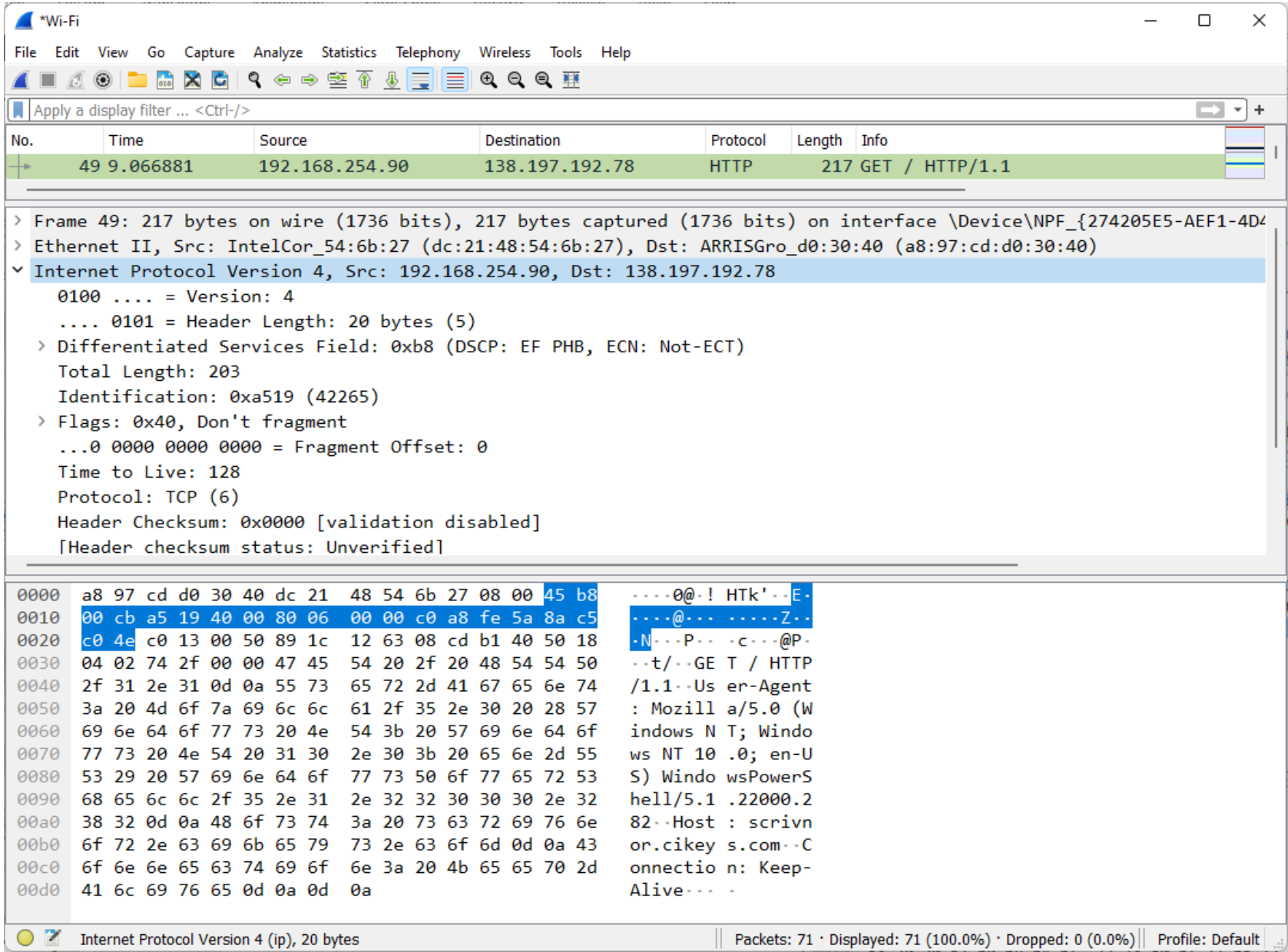
Physical Layer – actual bits on the wire



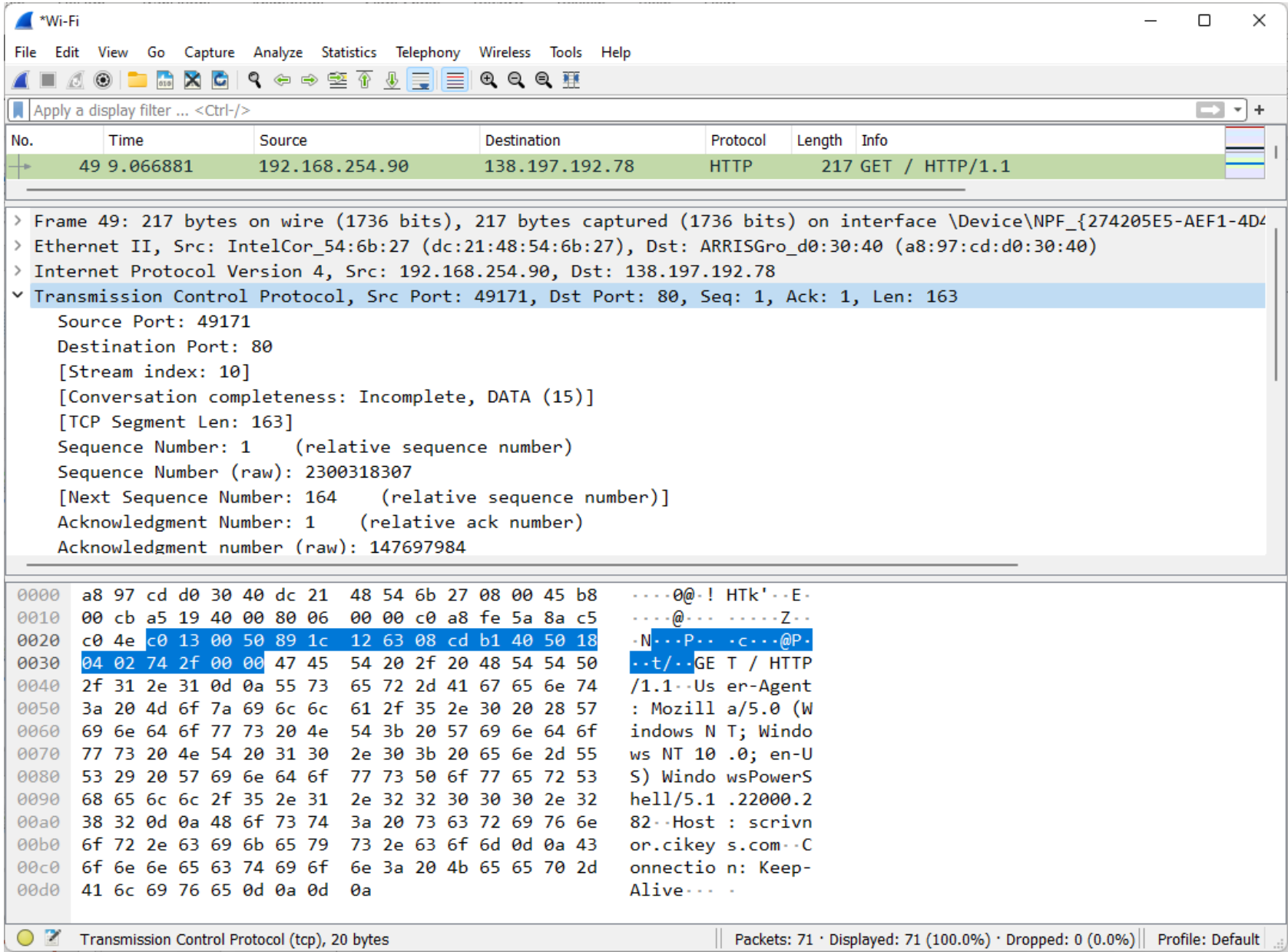
Data Link Layer – wifi in this case



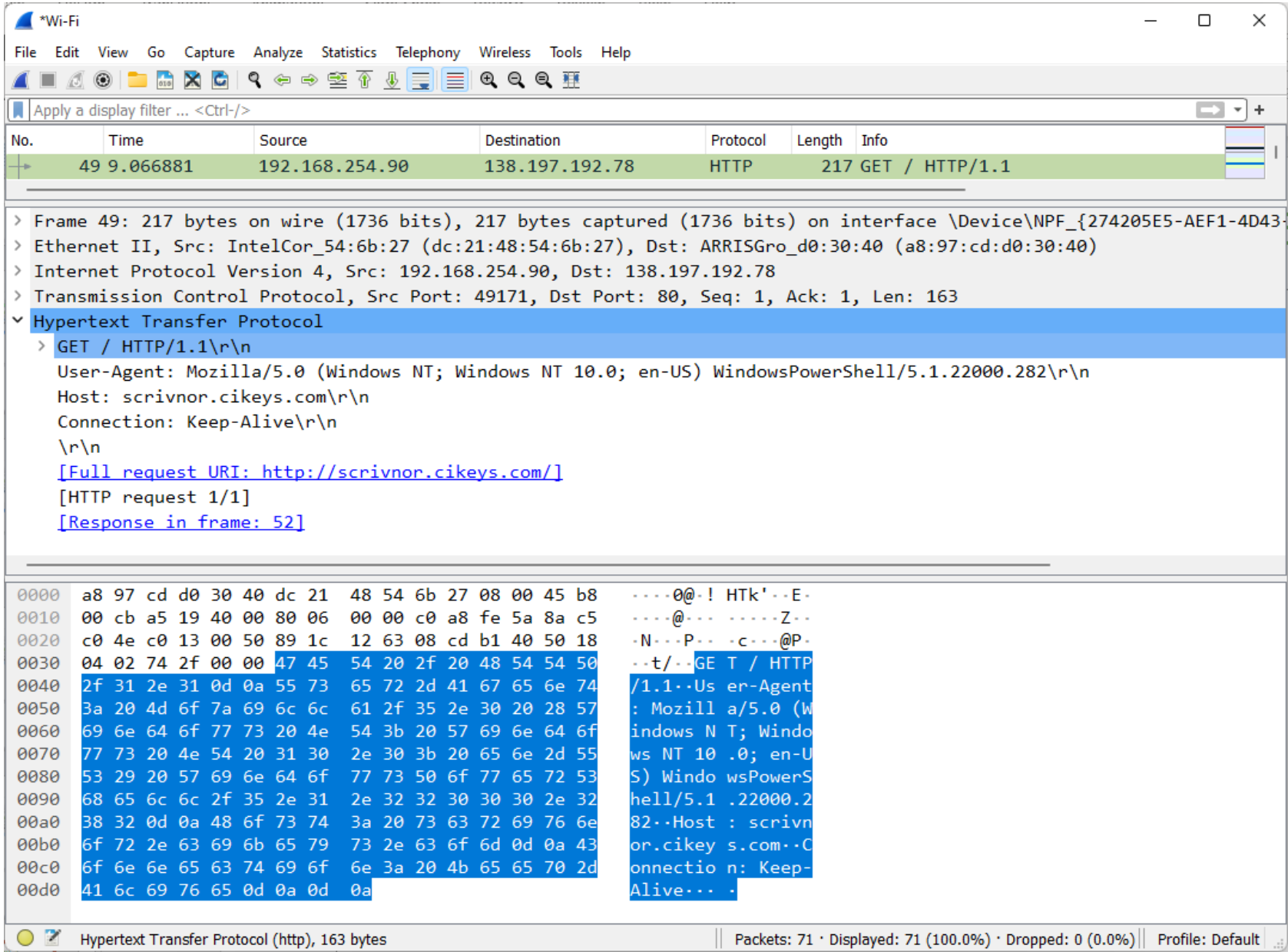
Network Layer – IPv4



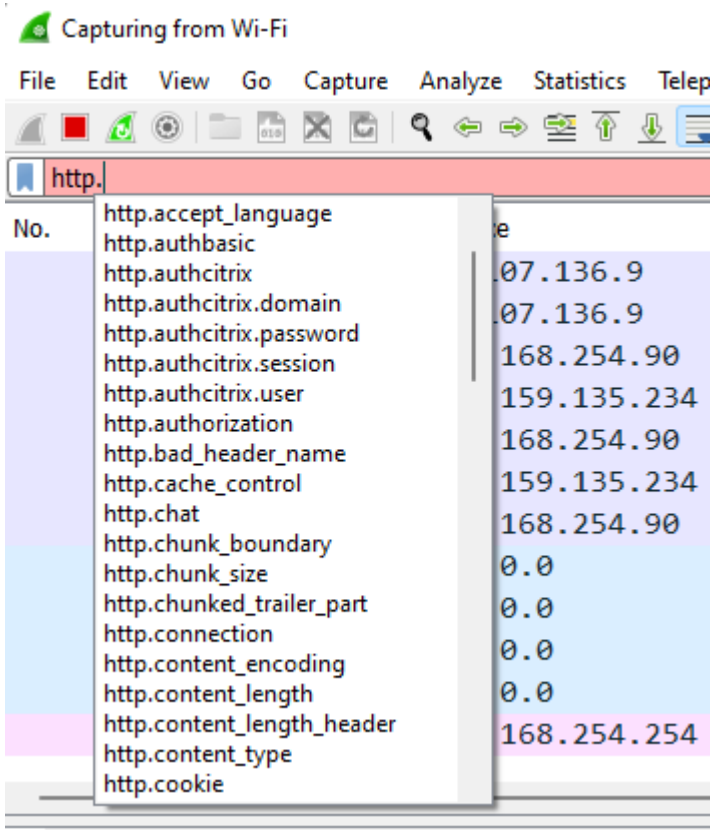
Transport Layer - TCP



Application Layer – HTTP (client request for file)



Filtering



- Filtering is a very powerful and useful feature in Wireshark.
- You filter by protocol, say http
  - Valid filter: http
  - Result: shows you all the packets related to http
- You filter more specifically
  - Valid filter: http.request.method == “GET”
  - Result: shows you all the GET requests made.
- You have to know what the protocol headers are, so that you can filter down.
  - IDEA: type “http.” into the filter bar and look at the results. You can drill down until you find what you need.

Practice

Download and Analyze TFTP Wireshark Capture

- Wireshark Capture: http://429.scrivnor.cikeys.com/captures/week03-tftp.pcapng

1. How many unique message types do we see in this transaction?
2. What is the default block size for TFTP?
3. What is the block size used in this transaction?
4. Which frame numbers contain read requests, if any?
5. Which frame numbers contain write requests, if any?
6. Is a file being transferred from the server or being sent to the server? How do you know?
7. Given your answer to number 1, which IP address represents the server?
8. How does the client/server know the file transfer is complete?