HW3 Part 1 – Wireshark

# Part 1 – Getting Familiar with Wireshark

## Q.1.1)

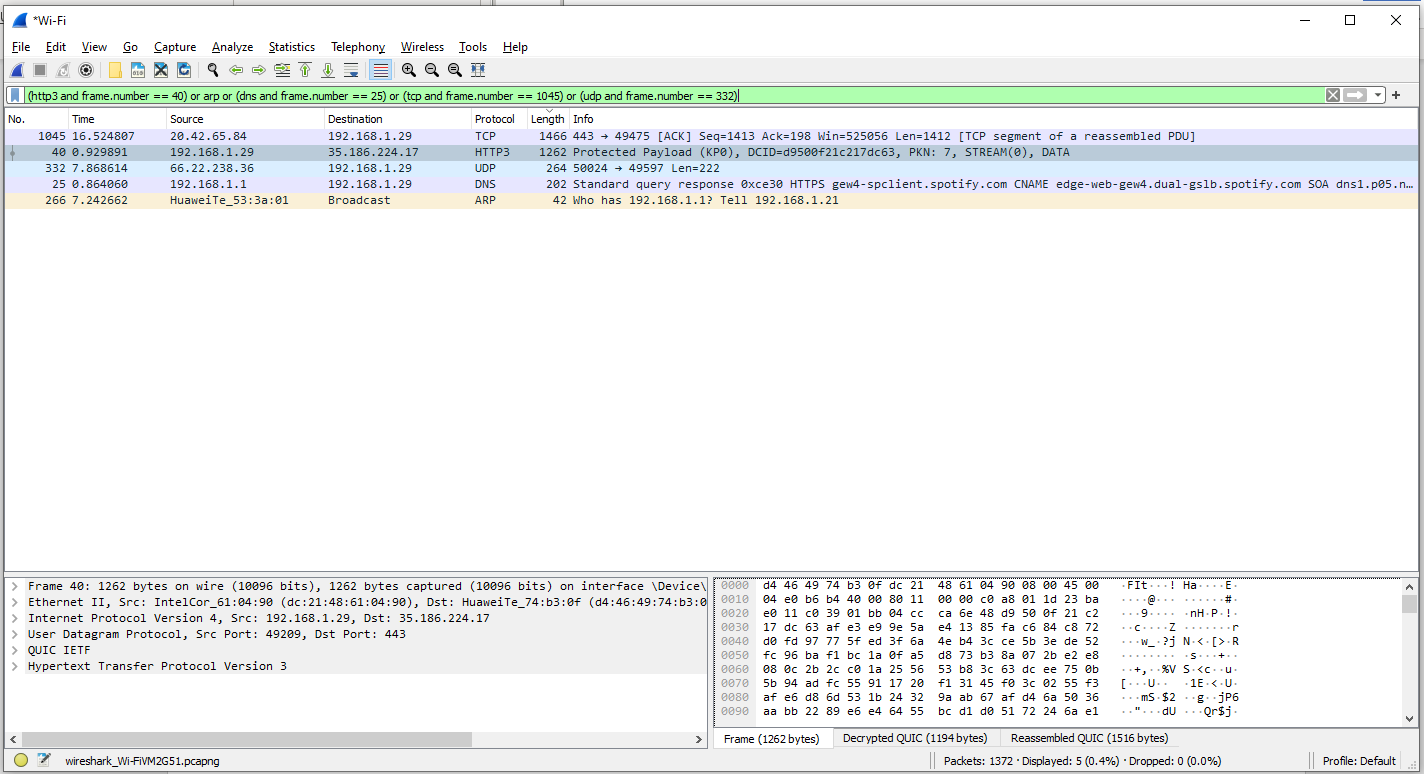


Figure A sample screen print of Wireshark with applied filters

To show the unique protocols only, I applied a filter. First, I searched for the protocol name and then applied its frame number to the filter.

(**http3** and frame.number == 40) or **arp** or (**dns** and frame.number == 25) or (**tcp** and frame.number == 1045) or (**udp** and frame.number == 332) or **icmpv6**

1. **HTTP3 (Hypertext Transfer Protocol version 3):** It is a transport protocol used for secure and efficient communication between web browsers and servers.
2. **ARP (Address Resolution Protocol):** It resolves an IP address to its corresponding MAC address on a local network, facilitating communication between devices.
3. **DNS (Domain Name System):** It translates domain names (e.g., www.example.com) into IP addresses, enabling users to access websites using human-readable names instead of numeric IP addresses.
4. **TCP (Transmission Control Protocol):** It provides reliable, connection-oriented data transmission between devices on a network, ensuring packets are delivered in the correct order and without errors.
5. **UDP (User Datagram Protocol):** It is a lightweight, connectionless protocol that allows for faster data transmission but does not guarantee reliability, making it suitable for applications that prioritize speed over error checking.
6. **ICMPv6 (Internet Control Message Protocol version 6):** It is used for diagnostic and error reporting purposes in IPv6 networks, allowing devices to exchange control messages to verify network connectivity, perform neighbor discovery, and handle error conditions.

# Part 2 – HTTP, TCP, DNS

## Q.2.1)

A screenshot of a computer

Description automatically generated

Figure Screenshot of Wireshark after entering the website

A screenshot of a computer

Description automatically generated

Figure IP filter applied to the Figure 2, steps after DNS resolution

1. DNS resolution happens for eee.metu.edu.tr. It sends a DNS query to a DNS server, and then IP address is resolved.
2. Once the IP address is resolved, a TCP connection is established with the server. It can be seen on screenshots that there is a three-way handshake with SYN packet to the server, SYN-ACK packet response from server and ACK packet as acknowledgment from browser.
3. After the TCP connection is established, an HTTP request is sent to the server to GET the image.
4. The server receives the HTTP request from the client and sends an HTTP response. Normally, we would see HTTP 200 OK status, but the response is HTTP 301 Moved Permanently. The clients sends another GET request to the redirection link in HTTP 301 location headers, and receives the HTTP 200 OK status. For the GET request for favicon.ico file, we receive a HTTP 404 Not Found response and it is not loaded.
5. After the HTTP response, the client downloads the image/webpage via TCP connection to reconstruct the image locally.
6. After the task is done (image is downloaded, loaded and webpage is ready), the client sends a TCP FIN packet to server to close the connection. The server sends an ACK packet as a response to the FIN packet.

## Q.2.2)

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark screenshot with http filter

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark packet details, showing tcp.stream value

* Here, we can see that the IP address and the port of the image is **144.122.145.144:80**
* My IP address and the port is **192.168.1.29:52923** (This is a local IP address)
* After clicking the response, right clicking the packet and selecting Follow->TCP Stream, we can see the TCP stream index of the image. **tcp.stream eq 2** means that, our index for image is 2.

## Q.2.3)

A screenshot of a computer

Description automatically generated

Figure Wireshark screenshot to show SYN, SYN-ACK, ACK messages

After applying the filter, we can see that the first 3 packets are used for the 3-way handshake.

1. **Packet 1 (SYN):** The initial packet that the client sends for the handshake. Details can be seen in the Info column for Seq, Win. Also, flag details can be seen in packet details pane. Source and Destination IPs are observed in the same row. Port direction is also shown in Info (52923 -> 80)
2. **Packet 2 (SYN-ACK):** This packet is the response from the server to acknowledge the client’s SYN packet. Similar details can be found as it was done for Packet 1.
3. **Packet 3 (ACK):** This packet is the final step of the handshake, and it is sent by the client to the server to acknowledge the server’s SYN-ACK packet. Similar details can be found as it was done for Packet 1 and 2.

## Q.2.4)

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark screenshot to show encapsulation

By looking at the header lengths, we can calculate the encapsulation size

- Total Length is 1466 bytes, IPv4 is 1452 bytes -> Ethernet header size = 1466 – 1452 = 14 bytes  
- IPv4 Header size = 20 Bytes  
- TCP Header size = 20 Bytes  
- Payload size = 1466 - 14 - 20 - 20 = 1412 bytes  
- Method 1: Encapsulation size = 1466 – 1412 = **54 bytes**  
- Method 2: Encapsulation size = 14 + 20 + 20 = **54 bytes**

## Q.2.5

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark screenshot to find througput with calculation

First, we sort the TCP stream, then take the first frame as time reference(CTRL+T). With this way, by looking at the last frame in the stream, we see that the duration in the stream is 0.085685 sec.

## Q.2.6)

A picture containing text, line, diagram, plot

Description automatically generated

Figure Wireshark statistics->throughput graph

## Q.2.7)

We see ACK messages sent for each frame, FIN-ACK has been send after the last packet (not in the throughput graph), and at the end, HTTP 200 OK message is received by client. With checking these information we can say HTTP is consistent.

## Q.2.8)

**HTTP 404 Not Found**

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark screenshot with 404 Message

**HTTP 304 Not Modified, HTTP 302 Found**

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark screenshot with 304 and 302 Messages

**HTTP 404 Not Found:** This response appears when the request is not found in the destination.

**HTTP 304 Not Modified:** This response appears if the requested resource has not been modified since last visit. The purpose of this code is to save bandwidth by not loading unchanged content again from the server, so it uses its cached copy.

**HTTP 302 Found:** This is a redirect response and it redirects the client to the its new location. In our example for https://developer.mozilla.org/, **GET /favicon.ico HTTP/1.1** moved to **GET /wp-content/uploads/favicon-1.ico HTTP/1.1** and response is HTTP 200 OK for the redirected address.

## Q.2.9)

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark screenshot to find cookie id, expiration date

Header: **set\_cookie** includes our unique ID for the cookie, and **expires** section shows when this cookie expires, which means that the website will forget about our basket.

## Q.2.10)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Figure Wireshark screenshot for DNS resolution

A screenshot of a computer error

Description automatically generated with low confidence

Figure Flags section in DNS Packet

The DNS header(Transaction ID, Flags, Questions, Answer RRs, Authority RRs, Additional RRs) is 12 bytes and Transaction ID is 2 bytes long. Main differences are Answer RRs and Authority RRs. Flags value provides detailed information on DNS solution.

# Part 3 – ICMP

## Q.3.1)

**ping –l 1000 208.67.222.222**

A screenshot of a computer program

Description automatically generated with low confidence

Figure Console screenshot for 1st ping operation

* ping –l 1000 208.67.222.222 sends ICMP echo request packets to the IP address 208.67.222.222. This IP address belongs to OpenDNS.
* The -l flag in the command lets us set the payload size.
* 1000 parameter that we have in our command corresponds to the payload size of 1000 bytes.
* **bytes** value is our packet size, **time** value is our RTT for each packet, **TTL** is remaining number of hops the packet can travel before being removed.
* **Pinging statistics** show statistics for our 4 packet pings.

**ping –l 2000 208.67.222.222**

A screenshot of a computer program

Description automatically generated with low confidence

Figure Console screenshot for 2nd ping operation

* ping –l 2000 208.67.222.222 is similar to our previous command, but with packet\_size = 2000 bytes. This time, due to the packet size, we lost all of our packets since they were not delivered in a certain time limit, or other limits that might have been decided for the OpenDNS.

**ping –l 9001 208.67.222.222**

A picture containing text, screenshot

Description automatically generated

Figure Console screenshot for 3rd ping operation

* For ping –l 9001 208.67.222.222 also we received a time out. This was highly expected since 2000 bytes already timed out.
* Jumboframes are Ethernet frames that are larger than the standard maximum frame size, which is 1500 bytes as a standard. If the network is not configured to handle jumboframes, it will respond with a time out.

**ping 0.0.0.0**

A picture containing text, screenshot, font

Description automatically generated

Figure Console screenshot for 4th ping operation

* ping 0.0.0.0 ping no addresses since it represents an blocked network destination. For this one, we did not set a size flag, so we ping with 32 bytes of data. Due to nature of 0.0.0.0 IP, we receive failure for each try.

**ping 127.0.0.0**

A picture containing text, screenshot, font

Description automatically generated

Figure Console screenshot for 5th ping operation

* For this one also, we see a general failure. 127.0.0.0 represents a loopback network and it allows user to communicate with its own machine by redirecting any message to its origin that is sent to it. Since it only represents the network, our ping fails but as an example, we can ping our own machine with 127.0.0.1

**ping 255.255.255.255**

A picture containing screenshot, text

Description automatically generated

Figure Console screenshot for 6th ping operation

* 255.255.255.255 is a special IP address named as broadcast. It sends network packets to all the devices in the local network. We are not able to find any devices in our local network and this can be caused by the network settings to prevent exploits such as **smurf attacks**.

**tracert twitter.com**

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure Console screenshot for tracert operation

* tracert twitter.com command traces the route to the IP address of the twitter.com. We see 17 network hops, which are routers, along the path with maximum limit of 30. The first column represents the sequence of the hop, the following 3 columns represents three roundtrip delay measurements. With tracert command, we can investigate the network path between the machine(client) and destination with the routers it visited, delays, number of hops.

## Q.3.2)

A screenshot of a computer

Description automatically generated

Figure Console screenshot for tracert operation

A screenshot of a computer

Description automatically generated with medium confidence

Figure Wireshark screenshot to find header and payload length

The upper layer protocol field name is ICMP, Internet Control message Protocol. ICMP is a error-reporting protocol for network devices. As it can be observed from the screenshot, there are 20 bytes in the IP header and 72 bytes (Total Length – Header Length = 92 – 20) for the payload.

## Q.3.3)

If we compare two Echo (ping) requests:

A picture containing text, number, font, screenshot

Description automatically generatedA picture containing text, screenshot, number, font

Description automatically generated

Figure Echo Pings Screenshot Comparison

* Identification, Checksum, Sequence Number changes, other fields must stay same. Time to Live changes after 3 packets.

## Q.3.4)

We have 13 TTL triplets and the reason behind is that, tracert command sends 3 packets for each TTL. Since there are 13 hops in the network, we have 13 TTL triples, which makes 39 echo (ping) requests. Tracert sends 3 packets instead of 1 to have more reliable, consistent statistical analysis for network information.

## Q.3.5)

A picture containing text, receipt, number, font

Description automatically generatedA picture containing text, receipt, number, parallel

Description automatically generatedA picture containing text, receipt, number, font

Description automatically generated

Figure ICMP TTL- exceeded replies comparison

When we compare multiple examples of ICMP TTL-exceeded replies, we see that total length, identification, flags, TTL, Header Checksum, Source Address, Destination address, checksum changes. The significant changes such as identification, total length happens due to different network configurations, devices and routing paths. Again, we see a pattern of TTL triples for this example too.