



MIDDLE EAST TECHNICAL UNIVERSITY
ELECTRICAL-ELECTRONICS ENGINEERING
EE444 Introduction to Computer Networks

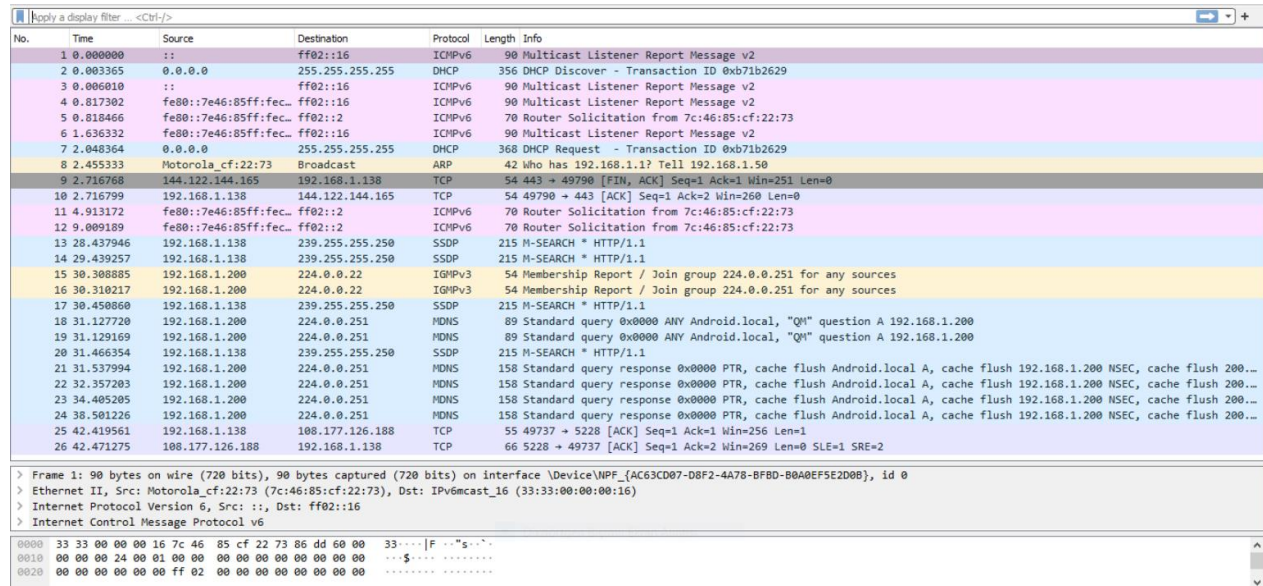
Mehmet ATAŞ 2304020

Submission Date: 27.05.2022

HW2 - Wireshark

Part 1 - Getting Familiar with Wireshark

To connect to the Internet, I utilize a wireless interface. I capture the packets in Wireshark for about 20 seconds, and the screen print of recorded packets is shown in Figure 1.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
2	0.003365	0.0.0.0	255.255.255.255	DHCP	356	DHCP Discover - Transaction ID 0xb71b2629
3	0.006010	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
4	0.017302	fe80::7e46:85ff:fec...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
5	0.018466	fe80::7e46:85ff:fec...	ff02::2	ICMPv6	70	Router Solicitation from 7c:46:85:cf:22:73
6	1.636332	fe80::7e46:85ff:fec...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
7	2.048364	0.0.0.0	255.255.255.255	DHCP	368	DHCP Request - Transaction ID 0xb71b2629
8	2.455333	Motorola_cf:22:73	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.50
9	2.716768	144.122.144.165	192.168.1.138	TCP	54	443 → 49790 [FIN, ACK] Seq=1 Ack=1 Win=251 Len=0
10	2.716799	192.168.1.138	144.122.144.165	TCP	54	49790 → 443 [ACK] Seq=1 Ack=2 Win=260 Len=0
11	4.913172	fe80::7e46:85ff:fec...	ff02::2	ICMPv6	70	Router Solicitation from 7c:46:85:cf:22:73
12	9.009189	fe80::7e46:85ff:fec...	ff02::2	ICMPv6	70	Router Solicitation from 7c:46:85:cf:22:73
13	28.437946	192.168.1.138	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
14	29.439257	192.168.1.138	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
15	30.308885	192.168.1.200	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
16	30.310217	192.168.1.200	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
17	30.450860	192.168.1.138	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
18	31.127720	192.168.1.200	224.0.0.251	MDNS	89	Standard query 0x0000 ANY Android.local, "QM" question A 192.168.1.200
19	31.129169	192.168.1.200	224.0.0.251	MDNS	89	Standard query 0x0000 ANY Android.local, "QM" question A 192.168.1.200
20	31.466354	192.168.1.138	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
21	31.537994	192.168.1.200	224.0.0.251	MDNS	158	Standard query response 0x0000 PTR, cache flush Android.local A, cache flush 192.168.1.200 NSEC, cache flush 200...
22	32.357203	192.168.1.200	224.0.0.251	MDNS	158	Standard query response 0x0000 PTR, cache flush Android.local A, cache flush 192.168.1.200 NSEC, cache flush 200...
23	34.405205	192.168.1.200	224.0.0.251	MDNS	158	Standard query response 0x0000 PTR, cache flush Android.local A, cache flush 192.168.1.200 NSEC, cache flush 200...
24	38.501226	192.168.1.200	224.0.0.251	MDNS	158	Standard query response 0x0000 PTR, cache flush Android.local A, cache flush 192.168.1.200 NSEC, cache flush 200...
25	42.419561	192.168.1.138	108.177.126.188	TCP	55	49737 → 5228 [ACK] Seq=1 Ack=1 Win=256 Len=1
26	42.471275	108.177.126.188	192.168.1.138	TCP	66	5228 → 49737 [ACK] Seq=1 Ack=2 Win=269 Len=0 SLE=1 SRE=2

> Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF_{AC63CD07-D8F2-4A78-BFBD-B0A0EF5E2D0B}, id 0
> Ethernet II, Src: Motorola_cf:22:73 (7c:46:85:cf:22:73), Dst: IPv6multicast_16 (33:33:00:00:00:16)
> Internet Protocol Version 6, Src: ::, Dst: ff02::16
> Internet Control Message Protocol v6

0000 33 33 00 00 00 16 7c 46 85 cf 22 73 86 dd 60 00 33...[F...s...
0010 00 00 00 24 00 01 00 00 00 00 00 00 00 00 00 00 ...\$.....
0020 00 00 00 00 00 00 ff 02 00 00 00 00 00 00 00 00

Figure 1: Screen Print of Captured Packets on Wireshark

The following protocols were observed:

TCP: Transmission Control Protocol

ARP: Address Resolution Protocol

DHCP: Dynamic Host Configuration Protocol

ICMP: Internet Control Message Protocol

IGMP: Internet Group Management Protocol

SSDP: Simple Service Discovery Protocol

MDNS: Multicast Domain Name System

Part 2 – HTTP, TCP, DNS

2.1.

TCP connections are created initially, followed by a DNS connection to www.metu.edu.tr with a typical query. TLS packets concerning client hello and client handshaking are also included. Following that, HTTP requests and HTTP answers are sent to the website sites.eee.metu.edu.tr/files/images/eehistory.jpg using base HTML first, followed by HTTP connections with image files. At the conclusion, TCP connections are closed.

2.2.

As shown in Figure 2, the IP address of the real source of the picture file is 144.122.145.144, and the port number is 80. 192.168.43.91 is my IP address. The TCP stream index associated with the picture download is 10.

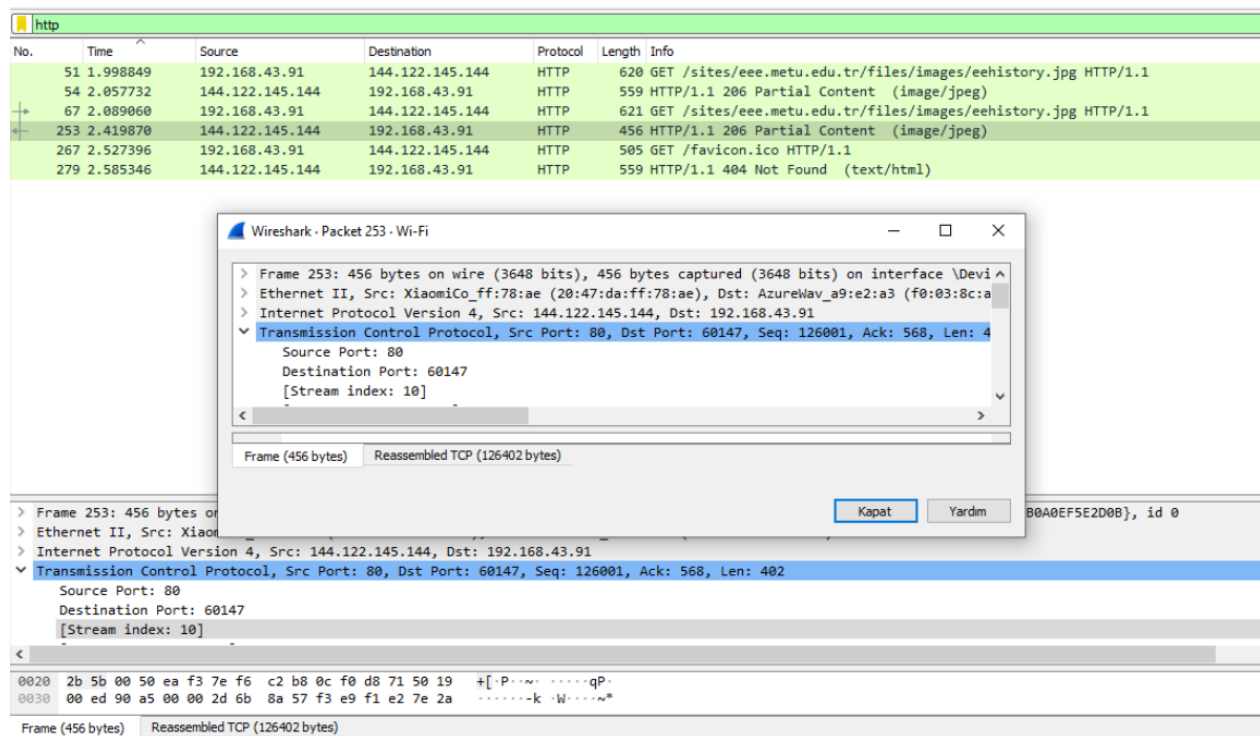


Figure 2: Screen Print of HTTP Traffic Only

2.3.

The internal Wireshark mapping for this TCP Stream 10 is [192.168.43.91, 60147, 144.122.145.144, 80]. As seen in Figure 3, all packets for this TCP stream value have the identical mapping field values.

3-way handshake for this TCP connection:

The first packet is a SYN packet from the client (me) to the server, and its sequence number (client Seq) is 0.

The second packet is the server's response (SYN+ACK) to the client; its sequence number (Seq of server) is 0, and its ACK number is 1 (which is Seq of client + 1).

The third packet is the client's response (ACK), and its sequence number (Seq of client) is 1. (which is equal to ACK coming from server). Its ACK number is 1 (which equals the server's Seq + 1).

tcp.stream eq 10							
No.	Time	Source	Destination	Protocol	Length	Info	
59	2.859344	192.168.43.91	144.122.145.144	TCP	66	60147 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	
65	2.888740	144.122.145.144	192.168.43.91	TCP	66	80 → 60147 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400 SACK_PERM=1 WS=128	
66	2.888778	192.168.43.91	144.122.145.144	TCP	54	60147 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0	
67	2.889060	192.168.43.91	144.122.145.144	HTTP	621	GET /sites/eee.metu.edu.tr/files/images/eehistory.jpg HTTP/1.1	
70	2.127282	144.122.145.144	192.168.43.91	TCP	54	80 → 60147 [ACK] Seq=1 Ack=568 Win=30336 Len=0	
71	2.129300	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=1 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
72	2.137598	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=1401 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
73	2.137644	192.168.43.91	144.122.145.144	TCP	54	60147 → 80 [ACK] Seq=568 Ack=2801 Win=65792 Len=0	
74	2.148906	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=2801 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
75	2.161156	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=4201 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
76	2.161196	192.168.43.91	144.122.145.144	TCP	54	60147 → 80 [ACK] Seq=568 Ack=5601 Win=65792 Len=0	
77	2.161463	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=5601 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
78	2.167037	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=7001 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
79	2.167331	192.168.43.91	144.122.145.144	TCP	54	60147 → 80 [ACK] Seq=568 Ack=8401 Win=65792 Len=0	
80	2.169519	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=8401 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
81	2.172680	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=9801 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
82	2.172716	192.168.43.91	144.122.145.144	TCP	54	60147 → 80 [ACK] Seq=568 Ack=11201 Win=65792 Len=0	
83	2.175970	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=11201 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
84	2.177975	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=12601 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
85	2.178045	192.168.43.91	144.122.145.144	TCP	54	60147 → 80 [ACK] Seq=568 Ack=14001 Win=65792 Len=0	
86	2.203263	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=14001 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	
87	2.203524	144.122.145.144	192.168.43.91	TCP	1454	80 → 60147 [ACK] Seq=15401 Ack=568 Win=30336 Len=1400 [TCP segment of a reassembled PDU]	

Header checksum: 0xce58 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.43.91
Destination Address: 144.122.145.144

Transmission Control Protocol, Src Port: 60147, Dst Port: 80, Seq: 568, Ack: 19601, Len: 0
Source Port: 60147
Destination Port: 80
[Stream index: 10]

0010	00 28 1e 69 40 00 80 06	ce 58 c0 a8 2b 5b 90 7a	·(·i@·····X··+·[·z
0020	91 90 ea f3 00 50 0c f0	d8 71 7e f5 23 18 50 10	·····P····q·~#·P·
0030	01 01 2e 12 00 00		·····

Figure 3: Screen Print of TCP Stream 10

2.4.

Figure 4 shows how 1400 bytes are utilized to encapsulate the largest frame (frame 251).

Wireshark · Packet 251 · HW1_Part2.pcapng	
<p>Frame 251: 1454 bytes on wire (11632 bits), 1454 bytes captured (11632 bits) on interface \Device\NPF_{AC63CD07-D8F2-4A78-BFBD-B0A0EF5E2D08}, id 0</p> <p>> Interface id: 0 (\Device\NPF_{AC63CD07-D8F2-4A78-BFBD-B0A0EF5E2D08})</p> <p>Encapsulation type: Ethernet (1)</p> <p>Arrival Time: May 11, 2021 13:00:17.916509000 Türkiye Standart Saati</p> <p>[Time shift for this packet: 0.000000000 seconds]</p> <p>Epoch Time: 1620727217.916509000 seconds</p> <p>[Time delta from previous captured frame: 0.001005000 seconds]</p> <p>[Time delta from previous displayed frame: 0.001005000 seconds]</p> <p>[Time since reference or first frame: 2.417414000 seconds]</p> <p>Frame Number: 251</p> <p>Frame Length: 1454 bytes (11632 bits)</p> <p>Capture Length: 1454 bytes (11632 bits)</p> <p>[Frame is marked: False]</p> <p>[Frame is ignored: False]</p> <p>[Protocols in frame: eth:ethertype:ip:tcp]</p> <p>[Coloring Rule Name: HTTP]</p> <p>[Coloring Rule String: http tcp.port == 80 http2]</p> <p>> Ethernet II, Src: XiaomiCo_ff:78:ae (20:47:da:ff:78:ae), Dst: AzureWav_a9:e2:a3 (f0:03:8c:a9:e2:a3)</p> <p>> Internet Protocol Version 4, Src: 144.122.145.144, Dst: 192.168.43.91</p> <p>Transmission Control Protocol, Src Port: 80, Dst Port: 60147, Seq: 124601, Ack: 568, Len: 1400</p> <p>Source Port: 80</p> <p>Destination Port: 60147</p> <p>[Stream index: 10]</p> <p>[TCP Segment Len: 1400]</p> <p>Sequence Number: 124601 (relative sequence number)</p> <p>Sequence Number (raw): 2130099520</p> <p>[Next Sequence Number: 126001 (relative sequence number)]</p> <p>Acknowledgment Number: 568 (relative ack number)</p> <p>Acknowledgment number (raw): 217110641</p> <p>0101 = Header Length: 20 bytes (5)</p> <p>> Flags: 0x010 (ACK)</p> <p>Window: 237</p> <p>[Calculated window size: 30336]</p> <p>[Window size scaling factor: 128]</p> <p>Checksum: 0x370c [unverified]</p> <p>[Checksum Status: Unverified]</p> <p>Urgent Pointer: 0</p>	

Figure 4: Information about Frame 251

There are two approaches for calculating this. The first approach is as follows:

Next Sequential Number – Sequential Number = 126001 – 124601 = 1400

(these values are seen in Figure 4) The bytes required to encapsulate this frame are merely the difference between the sequence numbers in TCP requests and answers.

The second way is to count the number of headers in a single frame. As seen in Figure 4, there are 1454 bytes in wire. However, not all of the 1454 bytes constitute data; some are headers. Let us compute the data length:

1454 kilobytes (length of the frame) – twenty bytes (IPv4) - TCP + checksum = 20 bytes DATA = 1400 bytes – 14 bytes (Ethernet) (and 54 B Overhead of TCP, IP and Ethernet)

2.5.

```
[Calculated window size: 30336]
[Window size scaling factor: 128]
Checksum: 0x370c [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
▼ [SEQ/ACK analysis]
  [iRTT: 0.029434000 seconds]
```

Figure 5: Window Size and RTT Values

W / RTT can be used to calculate throughput. The sender window size is W, and the round trip time is RTT.

With these numbers as shown in Figure 5, we can compute the throughput as follows:

W = 30336 bits

RTT = 0.02943 seconds

Throughput = 30336 / 0.02943 = 1030784 bits/seconds

However, this throughput is just for a single frame. To determine the throughput of this stream, sum all the lengths of received data, as shown in Figure 3, and divide by the time elapsed:

Received Data Length = 1454 bytes * 90 + 54 bytes * 7 + 66 bytes * 5 = 131568 bytes

Passed Time = 2.616978 – 1.838629 = 0.778349

Throughput = 169034 bytes / seconds for this stream.

2.6.

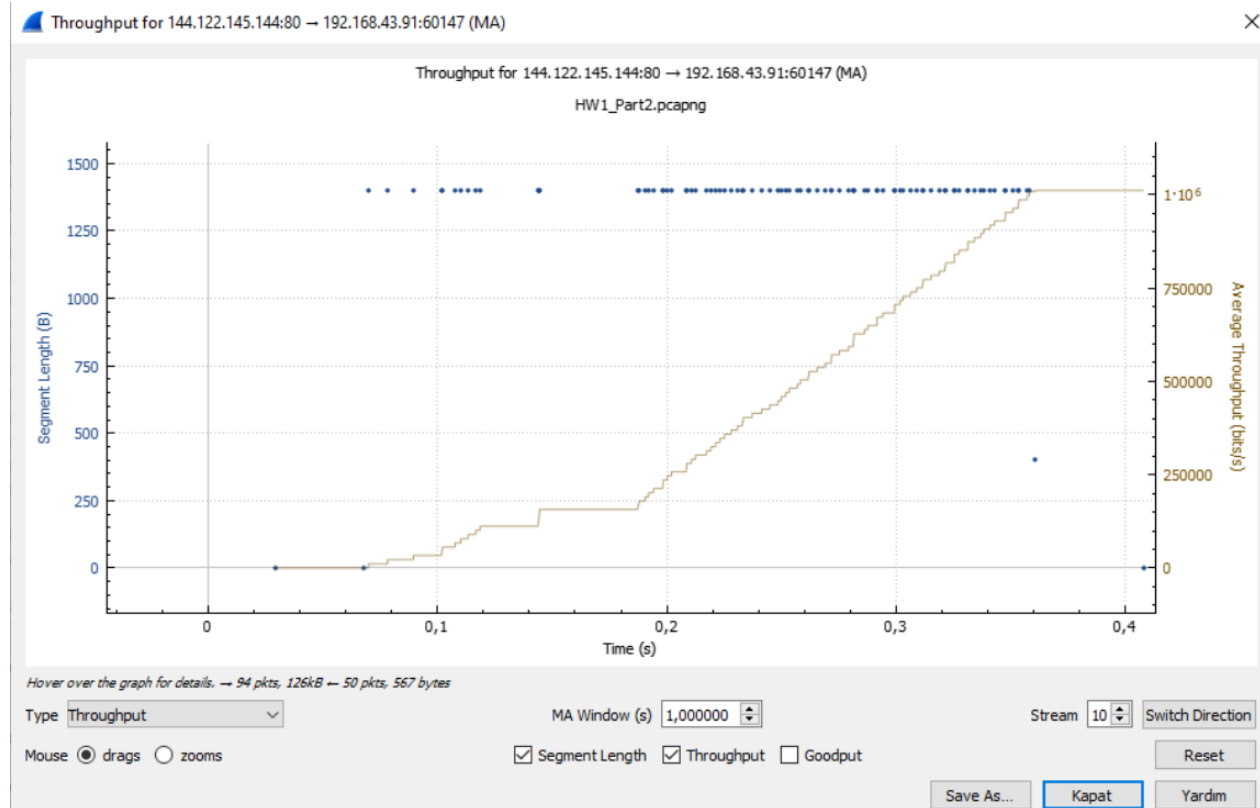


Figure 6: Throughput of TCP connection on Wireshark

2.7.

As seen in Figure 7, I clicked the HTTP frame linked with the basic HTML file. I checked if the Connection is "keepalive" to see whether the HTTP is durable or not (if it keeps the connection alive). Because it did, I deduced that it is persistent HTTP. If it were non-persistent HTTP, the connection would be "closed" and re-established each time.

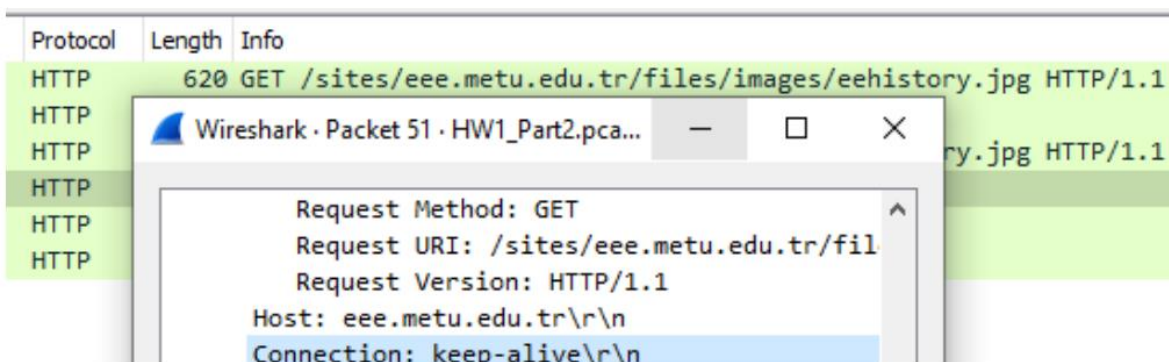


Figure 7: Connection: keep-alive

2.8.

The status code or error number for the website

<http://eee.metu.edu.tr/sites/eee.metu.edu.tr/files/images/eehistory.jpeg> is 404, which indicates that a file does not exist. Figures 8 and 9 show the fields that are connected.

No.	Time	Source	Destination	Protocol	Length	Info
39	1.266956	142.250.187.138	192.168.1.138	TLSv1.2	93	Application Data
40	1.284167	142.250.187.138	192.168.1.138	QUIC	654	Protected Payload (KP0)
41	1.284250	142.250.187.138	192.168.1.138	QUIC	118	Protected Payload (KP0)
42	1.284429	192.168.1.138	142.250.187.138	QUIC	75	Protected Payload (KP0), DCID=1bf61df72cfb704
43	1.297240	144.122.145.144	192.168.1.138	HTTP	834	HTTP/1.1 404 Not Found (text/html)

> Frame 43: 834 bytes on wire (6672 bits), 834 bytes captured (6672 bits) on interface \Device\NPF_{AC63CD07-D8F2-4A78-BFBD-B0A0EF5E2D0B}, id 0
 > Ethernet II, Src: Tilgin_a4:b4:60 (00:02:61:a4:b4:60), Dst: AzureWav_a9:e2:a3 (f0:03:8c:a9:e2:a3)
 > Internet Protocol Version 4, Src: 144.122.145.144, Dst: 192.168.1.138
 > Transmission Control Protocol, Src Port: 80, Dst Port: 63775, Seq: 1, Ack: 565, Len: 780
 > Hypertext Transfer Protocol
 > Line-based text data: text/html (1 lines)

Figure 8: 404 Error

No.	Time	Source	Destination	Protocol	Length	Info
84	1.508494	192.168.1.138	172.217.169.142	TLSv1.3	93	Application Data
85	1.527303	172.217.169.142	192.168.1.138	TCP	54	443 → 63777 [ACK] Seq=5010 Ack=1497 Win=68096 Len=0
86	1.568224	192.168.1.138	144.122.145.144	HTTP	562	GET /favicon.ico HTTP/1.1
87	1.572239	144.122.145.144	192.168.1.138	TCP	54	80 → 63776 [ACK] Seq=1 Ack=509 Win=30336 Len=0
88	1.576133	144.122.145.144	192.168.1.138	HTTP	559	HTTP/1.1 404 Not Found (text/html)

> Frame 43: 834 bytes on wire (6672 bits), 834 bytes captured (6672 bits) on interface \Device\NPF_{AC63CD07-D8F2-4A78-BFBD-B0A0EF5E2D0B}, id 0
 > Ethernet II, Src: Tilgin_a4:b4:60 (00:02:61:a4:b4:60), Dst: AzureWav_a9:e2:a3 (f0:03:8c:a9:e2:a3)
 > Internet Protocol Version 4, Src: 144.122.145.144, Dst: 192.168.1.138
 > Transmission Control Protocol, Src Port: 80, Dst Port: 63775, Seq: 1, Ack: 565, Len: 780
 > Hypertext Transfer Protocol
 > Line-based text data: text/html (1 lines)

Figure 9: 404 Error

The URL <http://capstone.eee.metu.edu.tr/> is then accessed. Figure 10 shows the status code "200 OK," which indicates that the request was successful.

No.	Time	Source	Destination	Protocol	Length	Info
95	4.072186	192.168.1.138	144.122.145.171	HTTP	578	GET / HTTP/1.1
159	5.300167	192.168.1.138	144.122.145.171	HTTP	569	GET /wp-content/plugins/jquery-lightbox-for-native-galleries/colorbox/theme1/colorbox.css
165	5.306310	192.168.1.138	144.122.145.171	HTTP	532	GET /wp-includes/css/dist/block-library/style.min.css?ver=5.7.1 HTTP/1.1
168	5.308268	192.168.1.138	144.122.145.171	HTTP	534	GET /wp-content/plugins/captcha/css/front_end_style.css?ver=4.4.5 HTTP/1.1
172	5.308980	192.168.1.138	144.122.145.171	HTTP	517	GET /wp-includes/css/dashicons.min.css?ver=5.7.1 HTTP/1.1
174	5.311145	144.122.145.171	192.168.1.138	HTTP	262	HTTP/1.1 200 OK (text/css)
176	5.311286	144.122.145.171	192.168.1.138	HTTP	123	HTTP/1.1 200 OK (text/html)
182	5.312358	192.168.1.138	144.122.145.171	HTTP	532	GET /wp-content/plugins/captcha/css/desktop_style.css?ver=4.4.5 HTTP/1.1
183	5.312537	192.168.1.138	144.122.145.171	HTTP	541	GET /wp-content/plugins/contact-form-7/includes/css/styles.css?ver=5.4.1 HTTP/1.1
184	5.312794	192.168.1.138	144.122.145.171	HTTP	539	GET /capstone/wp-content/plugins/my-twitter-widget/css/mtw-style.css?ver=5.7.1 HTTP/1.1
185	5.313988	192.168.1.138	144.122.145.171	HTTP	527	GET /wp-content/themes/picolight/pagenavi-css.css?ver=2.70 HTTP/1.1
196	5.320582	144.122.145.171	192.168.1.138	HTTP	395	HTTP/1.1 200 OK (text/css)
197	5.320582	144.122.145.171	192.168.1.138	HTTP	1145	HTTP/1.1 200 OK (text/css)
199	5.321409	192.168.1.138	144.122.145.171	HTTP	537	GET /wp-content/plugins/widget-options/assets/css/widget-options.css HTTP/1.1
200	5.322767	192.168.1.138	144.122.145.171	HTTP	521	GET /wp-content/themes/picolight/style.css?ver=5.7.1 HTTP/1.1
217	5.334497	144.122.145.171	192.168.1.138	HTTP	830	HTTP/1.1 200 OK (text/css)
218	5.334497	144.122.145.171	192.168.1.138	HTTP	1353	HTTP/1.1 200 OK (text/css)
220	5.334645	144.122.145.171	192.168.1.138	HTTP	705	HTTP/1.1 200 OK (text/css)
221	5.337618	192.168.1.138	144.122.145.171	HTTP	532	GET /wp-content/plugins/tablepress/css/default.min.css?ver=1.13 HTTP/1.1
222	5.337879	192.168.1.138	144.122.145.171	HTTP	545	GET /wp-content/plugins/youtube-embed-plus/styles/ytprefs.min.css?ver=13.4.2 HTTP/1.1
223	5.338169	192.168.1.138	144.122.145.171	HTTP	504	GET /wp-includes/js/jquery/jquery.min.js?ver=3.5.1 HTTP/1.1
244	5.344029	144.122.145.171	192.168.1.138	HTTP	1305	HTTP/1.1 200 OK (text/css)
249	5.345388	144.122.145.171	192.168.1.138	HTTP	1213	HTTP/1.1 200 OK (text/css)
250	5.345388	144.122.145.171	192.168.1.138	HTTP	303	HTTP/1.1 200 OK (text/css)
252	5.345442	192.168.1.138	144.122.145.171	HTTP	512	GET /wp-includes/js/jquery/jquery-migrate.min.js?ver=3.3.2 HTTP/1.1

Figure 10: 200 OK

The URL <http://mackolik.com> is then viewed. Figure 11 shows the status code "301 Relocated Permanently," which indicates that the requested item was moved and a new location was supplied later in this message (Location:). Figure 11 depicts the newly defined position.

No.	Time	Source	Destination	Protocol	Length	Info
4969	4.932032	192.168.1.138	93.184.220.66	TCP	54	63953 → 443 [ACK] Seq=2101 Ack=117648 Win=1043 Len=0
4970	4.938078	93.184.220.66	192.168.1.138	TCP	1506	443 → 63953 [ACK] Seq=117648 Ack=2101 Win=153 Len=1452 [TCP segment of a reassembled PDU]
4971	4.939137	93.184.220.66	192.168.1.138	TCP	1506	443 → 63953 [ACK] Seq=119100 Ack=2101 Win=153 Len=1452 [TCP segment of a reassembled PDU]
4972	4.939193	192.168.1.138	93.184.220.66	TCP	54	63953 → 443 [ACK] Seq=2101 Ack=120552 Win=1043 Len=0
4973	4.940294	93.184.220.66	192.168.1.138	TCP	1506	443 → 63953 [ACK] Seq=120552 Ack=2101 Win=153 Len=1452 [TCP segment of a reassembled PDU]
4974	4.941148	93.184.220.66	192.168.1.138	TLSv1.2	1463	Application Data
4975	4.941203	192.168.1.138	93.184.220.66	TCP	54	63953 → 443 [ACK] Seq=2101 Ack=123413 Win=1043 Len=0
4976	4.941286	54.192.233.127	192.168.1.138	TCP	54	80 → 64051 [ACK] Seq=20221 Ack=1686 Win=5120 Len=0
4977	4.941945	216.58.212.34	192.168.1.138	QUIC	67	Protected Payload (KP0)
4978	4.942101	104.26.10.25	192.168.1.138	TCP	54	80 → 64052 [ACK] Seq=1 Ack=323 Win=67584 Len=0
4979	4.942870	104.26.10.25	192.168.1.138	HTTP	694	HTTP/1.1 301 Moved Permanently

```

Connection: keep-alive\r\n
Cache-Control: max-age=3600\r\n
Expires: Tue, 11 May 2021 23:32:12 GMT\r\n
Location: https://hb.adpone.com/prebid_v4_21.js\r\n
cf-request-id: 09ff283daa0005166dc94c00000001\r\n
Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report?s=0S0WehEQTC%2BF9X1pRsnGJ%2FSynWkp2CBg0PrRH2yJn%2FKzc8H1cCe9J%2Bt11LN0X%5zeL8rgeRksmQ1M"}]}
NEL: {"report_to":"cf-nel","max_age":604800}\r\n
Vary: Accept-Encoding\r\n
Server: cloudflare\r\n
CF-RAY: 64dedca91fa55166-IST\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.120418000 seconds]
[Request in frame: 4954]

```

Figure 11: 301 Moved Permanently

2.9.

No.	Time	Source	Destination	Protocol	Length	Info
47	0.474776	178.18.206.165	192.168.1.9	HTTP2	1506	HEADERS[1]: 200 OK

```

Wireshark - Packet 47 - Wi-Fi
Header: set-cookie: odtuYayincilik=1ed8940dd67a919299186dbb9d3291dc3e8dfa38; expires=Sun, 16-May-2021 20:07:43 GMT; Max-Age=7200; path=/; secure; HttpOnly; SameSite=None
Name Length: 10
Name: set-cookie
Value Length: 149
Value: odtuYayincilik=1ed8940dd67a919299186dbb9d3291dc3e8dfa38; expires=Sun, 16-May-2021 20:07:43 GMT; Max-Age=7200; path=/; secure; HttpOnly; SameSite=None
set-cookie: odtuYayincilik=1ed8940dd67a919299186dbb9d3291dc3e8dfa38; expires=Sun, 16-May-2021 20:07:43 GMT; Max-Age=7200; path=/; secure; HttpOnly; SameSite=None
[Unescaped: odtuYayincilik=1ed8940dd67a919299186dbb9d3291dc3e8dfa38; expires=Sun, 16-May-2021 20:07:43 GMT; Max-Age=7200; path=/; secure; HttpOnly; SameSite=None]
Representation: Literal Header Field with Incremental Indexing - Indexed Name
Index: 55

```

Figure 12: HTTP2 set_cookie

Figure 12 shows that the Value, set-cookie, and Unescaped fields in the Header include unique IDs for me, which are "1ed8940dd67a919299186dbb9d3291dc3e8dfa38."

Figure 12 shows two fields relating keeping the basket for me: expires = Sun, 16-May-2021 20:07:43 GMT and Max-Age=7200. I looked for it and discovered that when both are utilized, Max-Age takes precedence over Expires. So the site will keep my basket for 7200 seconds, 120 minutes, or 2 hours.

2.10.

A DNS header is 12 bytes long and contains the following fields: identification, flags, number of questions, number of answers, number of authoritative resource records (RRs), and number of further RRs. Each field is 2 bytes long and occurs in this sequence.

<https://www.ece.cmu.edu/> Transaction Id is 0x83e3, as seen in Figure 13.

Transaction id of <https://www.ece.cmu.edu.tr/> is 0xf567, as seen in Figure 14.

The first flag bit indicates whether the DNS message is a query or a response. A "0" represents a question, while a "1" indicates an answer. Figure 13 depicts a question, whereas Figure 14 depicts a response.

The difference in the replies for the given addresses is that the website name <https://www.ece.cmu.edu/> is found, but not <https://www.ece.cmu.edu.tr/>.

As a result, it received the message "No such name," as seen in Figure 14.

Time	Source	Destination	Protocol	Length	Info
135	192.168.43.91	192.168.43.1	DNS	75	Standard query 0x83e3 A www.ece.cmu.edu
143	192.168.43.1	192.168.43.91	DNS	91	Standard query response 0x83e3 A www.ece.cmu.edu A 128.2.131.95
144	1.940450	192.168.43.91	DNS	89	Standard query 0x7206 A nav.smartscreen.microsoft.com
215	2.948273	192.168.43.91	DNS	89	Standard query 0x7206 A nav.smartscreen.microsoft.com
268	3.672250	192.168.43.1	DNS	214	Standard query response 0x7206 A nav.smartscreen.microsoft.com CNAME
459	5.110965	192.168.43.91	DNS	90	Standard query 0xa583 A smartscreen-prod.microsoft.com
464	5.160267	192.168.43.1	DNS	215	Standard query response 0xa583 A smartscreen-prod.microsoft.com CNAME
499	5.311691	192.168.43.91	DNS	80	Standard query 0xbcd0 A fonts.googleapis.com
501	5.361410	192.168.43.1	DNS	96	Standard query response 0xbcd0 A fonts.googleapis.com A 216.58.214.13
646	5.573504	192.168.43.91	DNS	81	Standard query 0x31e8 A live.staticflickr.com
732	5.762237	192.168.43.1	DNS	140	Standard query response 0x31e8 A live.staticflickr.com CNAME d3j7xsc0
1017	6.186021	192.168.43.91	DNS	80	Standard query 0x1d46 A platform.twitter.com
1126	6.259944	192.168.43.1	DNS	255	Standard query response 0x1d46 A platform.twitter.com CNAME cs472.wac
3659	8.716419	192.168.43.91	DNS	80	Standard query 0x8349 A platform.twitter.com
3661	8.719175	192.168.43.1	DNS	299	Standard query response 0x8349 A platform.twitter.com CNAME cs472.wac
4930	10.381761	192.168.43.91	DNS	73	Standard query 0xf719 A ton.twimg.com

Wireshark - Packet 135 - ececmuedu.pcapng

> Frame 135: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Device\NPF_{AC63CD07-D8F2-4A78-BFBD-B0A0EF5E2D0B}, id 0

> Ethernet II, Src: AzureWav_a9:e2:a3 (f0:03:8c:a9:e2:a3), Dst: XiaomiCo_ff:78:ae (20:47:da:ff:78:ae)

> Internet Protocol Version 4, Src: 192.168.43.91, Dst: 192.168.43.1

> User Datagram Protocol, Src Port: 54781, Dst Port: 53

> Domain Name System (query)

Transaction ID: 0x83e3

Flags: 0x0100 Standard query

0... .. = Response: Message is a query

.000 0... .. = Opcode: Standard query (0)

... ..0... .. = Truncated: Message is not truncated

... ..1... .. = Recursion desired: Do query recursively

... ..0... .. = Z: reserved (0)

... ..0... .. = Non-authenticated data: Unacceptable

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

Figure 13: <https://www.ece.cmu.edu/>

Time	Source	Destination	Protocol	Length	Info
18	1.867966	192.168.43.91	DNS	78	Standard query 0xf567 A www.ece.cmu.edu.tr
19	1.880144	192.168.43.1	DNS	78	Standard query response 0xf567 No such name A www.ece.cmu.edu.tr
20	1.975687	192.168.43.91	DNS	70	Standard query 0x79b6 A google.com
21	1.976279	192.168.43.91	DNS	70	Standard query 0xef8c A google.com
22	1.980438	192.168.43.1	DNS	86	Standard query response 0xef8c A google.com A 142.250.187.142
23	2.032557	192.168.43.91	DNS	75	Standard query 0x4a0a A ssl.gstatic.com
24	2.038247	192.168.43.1	DNS	91	Standard query response 0x4a0a A ssl.gstatic.com
26	2.222245	8.8.8.8	DNS	86	Standard query response 0x4a0a A ssl.gstatic.com

Wireshark - Packet 19 - Wi-Fi

... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 64

Identification: 0x13e3 (5091)

> Flags: 0x40, Don't fragment

Fragment Offset: 0

Time to Live: 64

Protocol: UDP (17)

Header Checksum: 0x4f1d [validation disabled]

[Header checksum status: Unverified]

Source Address: 192.168.43.1

Destination Address: 192.168.43.91

> User Datagram Protocol, Src Port: 53, Dst Port: 50339

> Domain Name System (response)

Transaction ID: 0xf567

Flags: 0x8183 Standard query response, No such name

1... .. = Response: Message is a response

.000 0... .. = Opcode: Standard query (0)

... ..0... .. = Authoritative: Server is not an authority for domain

... ..0... .. = Truncated: Message is not truncated

... ..1... .. = Recursion desired: Do query recursively

... ..1... .. = Recursion available: Server can do recursive queries

... ..0... .. = Z: reserved (0)

... ..0... .. = Answer authenticated: Answer/authority portion was not au

... ..0... .. = Non-authenticated data: Unacceptable

... ..0011 = Reply code: No such name (3)

Questions: 1

Figure 14: <https://www.ece.cmu.edu.tr/>

3.1.

```
Pinging 208.67.222.222 with 1000 bytes of data:
Reply from 208.67.222.222: bytes=1000 time=38ms TTL=52
Reply from 208.67.222.222: bytes=1000 time=30ms TTL=52
Reply from 208.67.222.222: bytes=1000 time=31ms TTL=52
Reply from 208.67.222.222: bytes=1000 time=31ms TTL=52

Ping statistics for 208.67.222.222:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 30ms, Maximum = 38ms, Average = 32ms

C:\Users\Casper> ping -l 2000 208.67.222.222

Pinging 208.67.222.222 with 2000 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 208.67.222.222:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Casper> ping -l 9001 208.67.222.222

Pinging 208.67.222.222 with 9001 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 208.67.222.222:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Figure 15: Ping Instructions

In ping instructions, the -l command specifies the size, and the address specifies where to ping, in this example 208.67.222.222. As shown in Figure 15, the size of the "ping -l 1000 208.67.222.222" command is 1000 bytes; the size of the "ping -l 2000 208.67.222.222" command is 2000 bytes; and the size of the "ping -l 9001 208.67.222.222" command is 9001 bytes.

The average RTT in the first instruction is 32 ms. It is believed that if it is less than 50 ms, it is OK, and the TTL for all packets is 52. In this situation, no packets are lost. Request timed out error is taken in the second and third commands. This message shows that no Echo Reply messages were received within the

1 second time limit. This can be caused by a variety of factors, the most common of which being network congestion, ARP request failure, packet filtering, routing error, or a quiet discard.

Jumbo frames are Ethernet frames containing more than 1500 bytes of payload, as defined by the IEEE 802.3 standard. Jumbo frames can typically store up to 9000 bytes of payload, however smaller and bigger versions occur, therefore the word should be used with caution.

As a result, the request for 2000 bytes has timed out on my machine, but it cannot be on another computer on another network. However, 9001 bytes is beyond the limit, and the request was supposed to time out.

```
Pinging 208.67.222.222 with 1472 bytes of data:
Reply from 208.67.222.222: bytes=1472 time=34ms TTL=52
Reply from 208.67.222.222: bytes=1472 time=33ms TTL=52
Reply from 208.67.222.222: bytes=1472 time=33ms TTL=52
Reply from 208.67.222.222: bytes=1472 time=33ms TTL=52

Ping statistics for 208.67.222.222:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 33ms, Maximum = 34ms, Average = 33ms

C:\Users\Casper> ping -l 1473 208.67.222.222

Pinging 208.67.222.222 with 1473 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 208.67.222.222:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Figure 16: My Trials to find the limit

When I tried what is the limit on my computer, I saw that the limit was 1472 bytes, as seen in Figure 16.


```

Pinging 0.0.0.0 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 0.0.0.0:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Casper> ping 127.0.0.0

Pinging 127.0.0.0 with 32 bytes of data:
General failure.
General failure.
General failure.
General failure.

Ping statistics for 127.0.0.0:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Casper> ping 255.255.255.255
Ping request could not find host 255.255.255.255. Please check the name and try again.

```

Figure 17: Ping Commands

Figure 17 shows that I received General Failure errors and that packets were lost. It is because the router is not in the same subnet as the "pinging" server. In this case, 0.0.0.0 represents this network device while 127.0.0.1 represents the local host. As a result, they are not on the same subnet as 208.67.222.222.

As shown in Figure 17, when the command "ping 255.255.255.255" is sent, the message "Ping request could not discover host 255.255.255.255." Please double-check the name and try again." In this case, 255.255.255.255 is a broadcast address, and I am sending a ping to every device on my local network, expecting a response from each device. This did not work since not all of the devices responded to my message.

```

C:\Users\Casper>tracert twitter.com

Tracing route to twitter.com [104.244.42.1]
over a maximum of 30 hops:

  1      1 ms      1 ms      1 ms  router.asus.com [192.168.1.1]
  2      5 ms      8 ms      5 ms  172.17.1.135
  3      5 ms      *          *    85.118.223.31.srv.turk.net [31.223.118.85]
  4     21 ms      6 ms      5 ms  86.118.223.31.srv.turk.net [31.223.118.86]
  5     13 ms      7 ms      7 ms  81.118.223.31.srv.turk.net [31.223.118.81]
  6      *          *          *    Request timed out.
  7      *         21 ms      *    22.100.146.159.srv.turk.net [159.146.100.22]
  8     68 ms     55 ms     55 ms  be2549.ccr31.sof02.atlas.cogentco.com [154.54.36.137]
  9    155 ms     56 ms     58 ms  be3421.ccr51.beg03.atlas.cogentco.com [130.117.0.94]
 10    155 ms     56 ms     62 ms  be3422.ccr31.bud01.atlas.cogentco.com [130.117.0.125]
 11    155 ms     75 ms    155 ms  be3261.ccr21.bts01.atlas.cogentco.com [130.117.3.137]
 12     55 ms     55 ms     57 ms  be2988.ccr51.vie01.atlas.cogentco.com [154.54.59.86]
 13     54 ms     49 ms     48 ms  130.117.14.182
 14     50 ms     50 ms     50 ms  win-bb4-link.ip.twelve99.net [62.115.114.182]
 15     50 ms     51 ms     47 ms  ffm-bb2-link.ip.twelve99.net [62.115.138.22]
 16      *         49 ms      *    ffm-b11-link.ip.twelve99.net [62.115.124.119]
 17    168 ms     56 ms     54 ms  twitter-ic322866-ffm-b11.ip.twelve99-cust.net [62.115.49.187]
 18    155 ms     55 ms    159 ms  104.244.42.1

Trace complete.

```

Figure 18: Trace Route Command

The trace route command tracks the path followed by a packet on an IP network from a source to a destination in real time, revealing the IP addresses of all routers pinged in between and the time spent for each hop. Figure 18 shows that the twitter.com website is opened in 18 hops, with 18 distinct IP addresses.

```
Tracing route to www.google.com.tr [216.58.205.195]
over a maximum of 30 hops:

  1  1 ms    4 ms    1 ms  router.asus.com [192.168.1.1]
  2  5 ms    6 ms    5 ms  172.17.1.135
  3  *        5 ms    *      85.118.223.31.srv.turk.net [31.223.118.85]
  4  5 ms    5 ms    59 ms  86.118.223.31.srv.turk.net [31.223.118.86]
  5  6 ms    6 ms    6 ms  81.118.223.31.srv.turk.net [31.223.118.81]
  6  *        22 ms   *      186.100.146.159.srv.turk.net [159.146.100.186]
  7  63 ms   124 ms   *      22.100.146.159.srv.turk.net [159.146.100.22]
  8  236 ms  457 ms   337 ms 195.175.51.209.static.turktelekom.com.tr [195.175.51.209]
  9  15 ms   16 ms   23 ms  00-gayrettepe-xrs-t2-2---00-gayrettepe-t3-9.statik.turktelekom.com.tr [212.156.121.138]
 10  *        14 ms   13 ms  00-ebgp-gayrettepe-k---00-gayrettepe-xrs-t2-2.statik.turktelekom.com.tr [81.212.202.19]
 11  23 ms   23 ms   23 ms  307-sof-col-2---00-ebgp-gayrettepe-k.statik.turktelekom.com.tr [212.156.104.156]
 12  51 ms   51 ms   51 ms  72.14.204.10
 13  54 ms   52 ms   52 ms  216.239.59.239
 14  54 ms   51 ms   51 ms  108.170.250.162
 15  52 ms   52 ms   52 ms  142.250.213.228
 16  52 ms   54 ms   51 ms  172.253.66.215
 17  51 ms   61 ms   51 ms  216.239.46.102
 18  52 ms   53 ms   52 ms  74.125.244.225
 19  53 ms   52 ms   53 ms  142.250.46.97
 20  51 ms   51 ms   54 ms  mil04s29-in-f3.1e100.net [216.58.205.195]

Trace complete.
```

Figure 19: Route Trace of google.com

3.2.

Figure 20 shows the name ICMP in the upper layer protocol field of the IP packet header (1). The IP header has 20 bytes. The payload of the IP datagram has 92 bytes, for a total of 112 bytes in the packet. To calculate the amount of payload bytes, subtract the IP header size (20 bytes) from the entire length (112 bytes), and the remaining is the payload size in bytes.

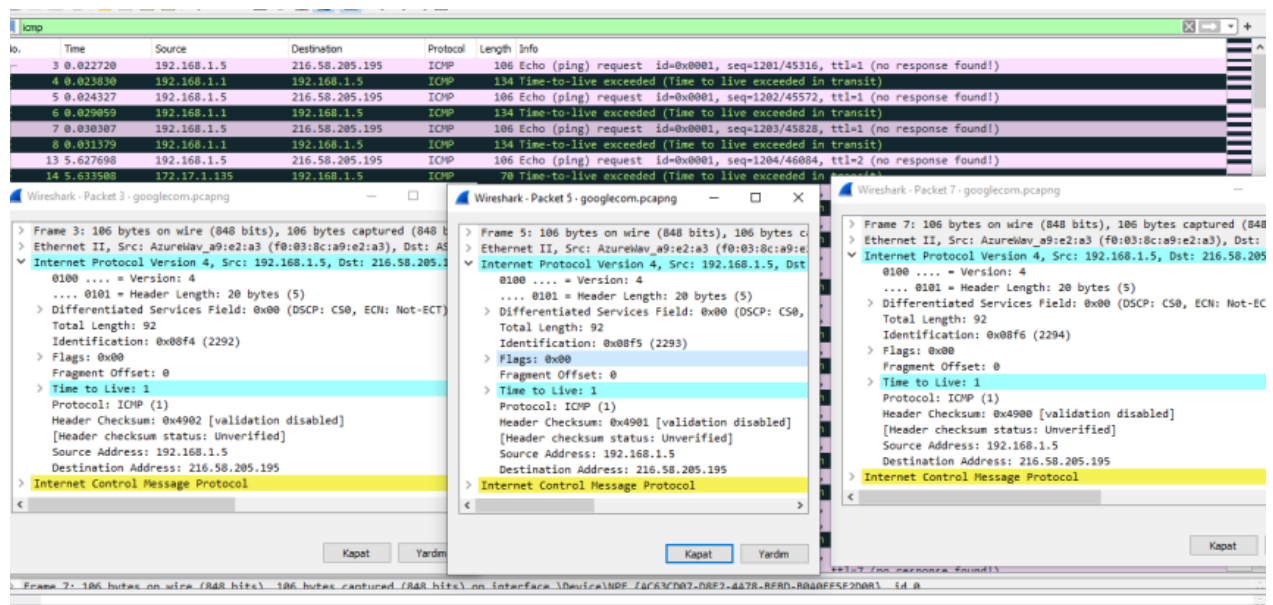


Figure 20: ICMP Echo Request Messages

3.3.

The IP datagram's Time to Live, Identification, and Header checksum fields are continually changing from one to the next. Because each packet's identifier is a unique number, it must always change. As a result, the Header checksum and Time to Live will both alter.

Version (because I'm using IPv4 for all packets), Header Length (because these are ICMP packets), Source IP (because I'm sending from the same source), Destination IP (because I'm sending to the same destination), Differentiated Services (because all packets are ICMP, they use the same Type of Service class), and Upper Layer Protocol must all remain constant (since these are ICMP packets).

3.4.

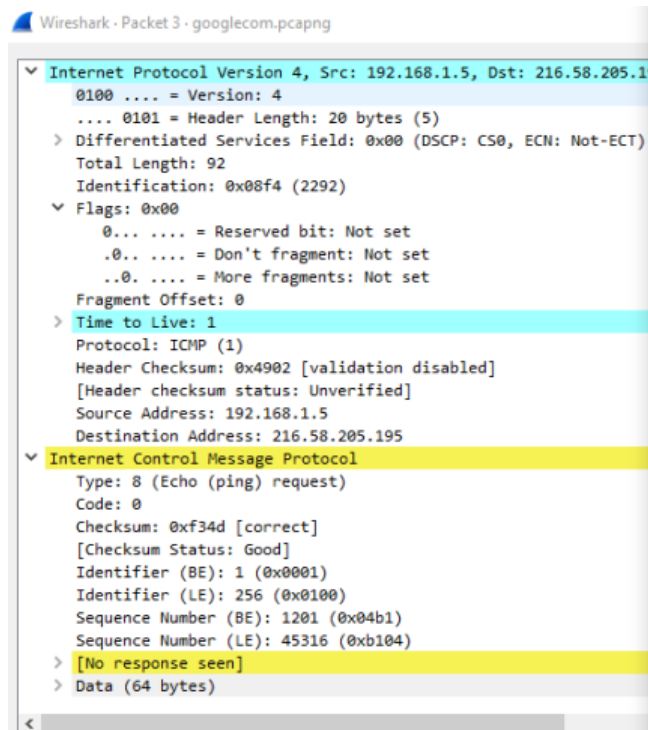
Three packets with the same TTL are sent.

Because trace-route delivers a message three times, the source really sends $3*N$ packets to the destination. The goal is to capture the source of each ICMP TTL exceeded message in order to give a trail of the path the packet followed to reach its destination. The differences in hop durations in Figure 19 can be attributed to a variety of factors such as network congestion, routing errors, and so on.

3.5.

When using the same router, the TTL will remain constant. Because it is allocated to a unique value, the identification field for every ICMP TTL-exceeded answers will vary. When two or more IP datagrams have the same identification value, it implies that they are fragments of a larger IP datagram, which is not the case here. Header Verification The number of messages sent from the same network varies as well. It is the same time to live. As seen in Figure 20, the remainder is similarly as predicted.

Header Checksum and Identification, as in the case of the same router, were the IP header fields that changed when the source router changed. In this instance, Time to Live is also shifting. Others refuse to change. Figure 21 shows an example of a comparison between two messages arriving from separate routers.



```
Wireshark · Packet 69 · googlecom.pcapng

> Frame 69: 106 bytes on wire (848 bits), 106 bytes captured (848) on interface 0
> Ethernet II, Src: AzureWav_a9:e2:a3 (f0:03:8c:a9:e2:a3), Dst: ASI
  > Internet Protocol Version 4, Src: 192.168.1.5, Dst: 216.58.205.195
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 92
    Identification: 0x08ff (2303)
  > Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    Fragment Offset: 0
  > Time to Live: 4
    Protocol: ICMP (1)
    Header Checksum: 0x45f7 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.5
    Destination Address: 216.58.205.195
  > Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf342 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 1212 (0x04bc)
    Sequence Number (LE): 48132 (0xbc04)
  > [No response seen]
  > Data (64 bytes)
```

Figure 21: Messages Coming from Two Different Routers

Appendix

Total time spent on report writing: 9 hours