

# **ISTANBUL TECHNICAL UNIVERSITY**



**BLG 433E**

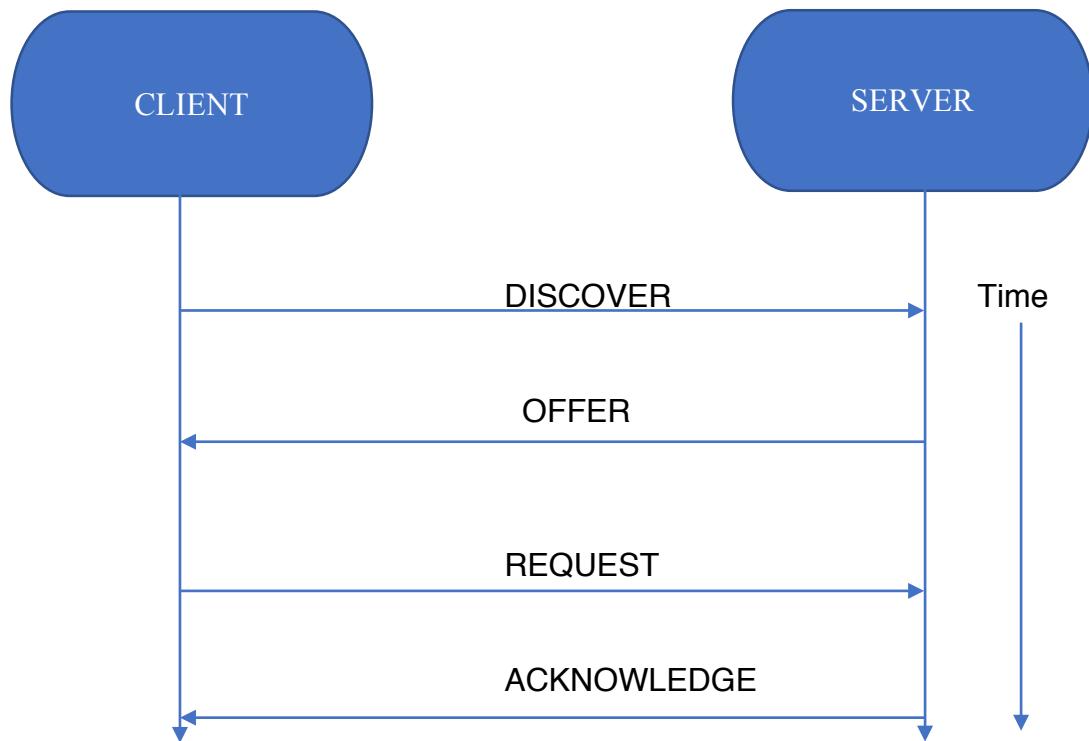
**COMPUTER COMMUNICATIONS**

**NETWORK ANALYZING WITH WIRESHARK**

## 1.DHCP (Dynamic Host Configuration Protocol)

### How DHCP Works?

DHCP protocol attaches IP addresses to clients on network automatically. So, it gives the capability of managing IP addresses from a center. DHCP also sends DNS server address and Default Gateway. It finishes the process in a 4-step sequence.



First, client sends a DHCP DISCOVER packet to find DHCP server. The packet contains client computer name and the MAC address.

```
▼ Bootstrap Protocol (Discover)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xd02746f2
  Seconds elapsed: 0
  ▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address:
  Client hardware address padding: 00000000000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  ▶ Option: (53) DHCP Message Type (Discover)
  ▶ Option: (55) Parameter Request List
  ▶ Option: (57) Maximum DHCP Message Size
  ▶ Option: (61) Client identifier
  ▶ Option: (51) IP Address Lease Time
  ▶ Option: (12) Host Name
  ▶ Option: (255) End
  Padding: 00000000000000000000000000000000
```

Picture 1 DHCP Discover

Second, DCHP server gets the packet and sends back a DHCP offer packet which includes IP address and Lease Time properties.

```
▼ Bootstrap Protocol (Offer)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 1
  Transaction ID: 0xd02746f2
  Seconds elapsed: 0
▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 161.9.66.65
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address:
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
▶ Option: (53) DHCP Message Type (Offer)
▶ Option: (54) DHCP Server Identifier
▶ Option: (51) IP Address Lease Time
▶ Option: (1) Subnet Mask
▶ Option: (3) Router
▶ Option: (6) Domain Name Server
▶ Option: (15) Domain Name
▶ Option: (0) Padding
▶ Option: (255) End
Padding: 00000000
```

*Picture 2 DHCP Offer*

Third, client accepts the DHCP offer packet and sends back DHCP request packet to say server DHCP offer packet accepted or not.

```
▼ Bootstrap Protocol (Request)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xd02746f2
  Seconds elapsed: 2
▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address:
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
▶ Option: (53) DHCP Message Type (Request)
▶ Option: (55) Parameter Request List
▶ Option: (57) Maximum DHCP Message Size
▶ Option: (61) Client identifier
▶ Option: (50) Requested IP Address
▶ Option: (54) DHCP Server Identifier
▶ Option: (12) Host Name
▶ Option: (255) End
Padding: 000000000000
```

*Picture 3 DHCP Request*

Fourth, DHCP server attaches the IP address and Lease Time. The client becomes TCP/IP client and participates to network.

```

▼ Bootstrap Protocol (ACK)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 1
  Transaction ID: 0xd02746f2
  Seconds elapsed: 2
▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 161.9.66.65
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address:
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  ▶ Option: (53) DHCP Message Type (ACK)
  ▶ Option: (54) DHCP Server Identifier
  ▶ Option: (51) IP Address Lease Time
  ▶ Option: (1) Subnet Mask
  ▶ Option: (3) Router
  ▶ Option: (6) Domain Name Server
  ▶ Option: (15) Domain Name
  ▶ Option: (0) Padding
  ▶ Option: (255) End
  Padding: 00000000

```

Picture 4 DHCP Acknowledge

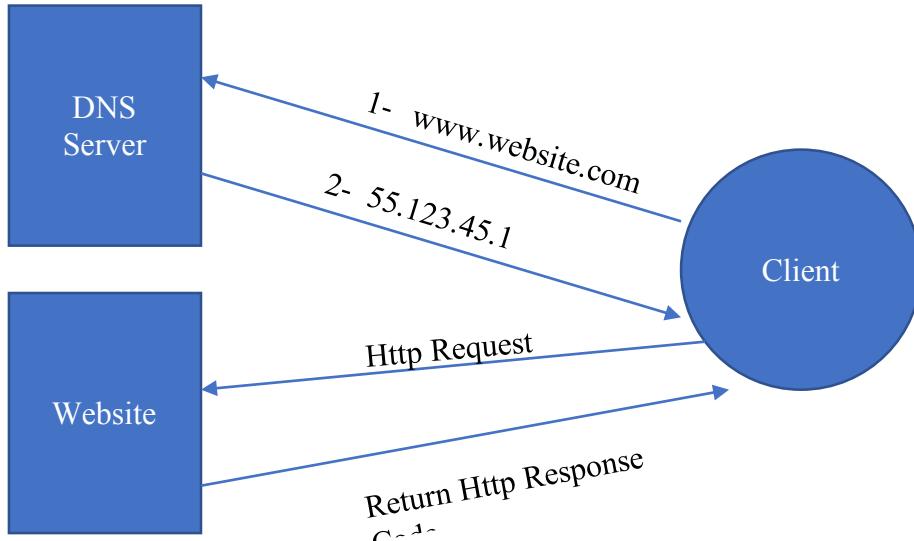
No.	Time	Source	Destination	Protocol	Length	Info
31	20.5614...	Apple_65:30:..	Broadcast	ARP	42	Gratuitous ARP for 161.9.66.65 (Request)
32	20.8814...	Apple_65:30:..	Broadcast	ARP	42	Gratuitous ARP for 161.9.66.65 (Request)
33	21.2046...	Apple_65:30:..	Broadcast	ARP	42	Gratuitous ARP for 161.9.66.65 (Request)
34	21.2067...	Apple_65:30:..	Broadcast	ARP	42	Who has 161.9.79.254? Tell 161.9.66.65
35	21.2126...	Cisco_9f:f0:..	Apple_65:30:..	ARP	60	161.9.79.254 is at 00:00:0c:9f:f0:cc
36	21.2405...	Apple_65:30:..	Broadcast	ARP	42	Who has 161.9.79.254? Tell 161.9.66.65
37	21.2502...	Cisco_9f:f0:..	Apple_65:30:..	ARP	60	161.9.79.254 is at 00:00:0c:9f:f0:cc
21	17.5658... 0.0.0.0			DHCP	342	DHCP Discover - Transaction ID 0xd02746f2
24	18.5826... 192.168.61.61	161.9.66.65		DHCP	346	DHCP Offer - Transaction ID 0xd02746f2
26	19.5881... 0.0.0.0			DHCP	342	DHCP Request - Transaction ID 0xd02746f2
27	19.5931... 192.168.61.61	161.9.66.65		DHCP	346	DHCP ACK - Transaction ID 0xd02746f2
38	21.2503... 161.9.66.65	160.75.25.1		DNS	98	Standard query 0x88ca PTR b._dns-sd._udp.0.64.9.161.in-addr.arpa
39	21.2503... 161.9.66.65	160.75.25.1		DNS	99	Standard query 0xebfb PTR db._dns-sd._udp.0.64.9.161.in-addr.arpa
40	21.2503... 161.9.66.65	160.75.25.1		DNS	99	Standard query 0x4d8f PTR lb._dns-sd._udp.0.64.9.161.in-addr.arpa
41	21.2503... 161.9.66.65	160.75.25.1		DNS	85	Standard query 0xf635 PTR b._dns-sd._udp.itu.edu.tr
42	21.2503... 161.9.66.65	160.75.25.1		DNS	86	Standard query 0xc36c PTR db._dns-sd._udp.itu.edu.tr
43	21.2503... 161.9.66.65	160.75.25.1		DNS	86	Standard query 0xe284 PTR lb._dns-sd._udp.itu.edu.tr
44	21.2503... 161.9.66.65	160.75.25.1		DNS	84	Standard query 0x17d8 PTR 65.66.9.161.in-addr.arpa
45	21.2530... 160.75.25.1	161.9.66.65		DNS	152	Standard query response 0x88ca No such name PTR b._dns-sd._udp.0.64.9.161.in-addr.arpa
46	21.2530... 160.75.25.1	161.9.66.65		DNS	153	Standard query response 0xebfb No such name PTR db._dns-sd._udp.0.64.9.161.in-addr.arpa
47	21.2530... 160.75.25.1	161.9.66.65		DNS	143	Standard query response 0xf635 No such name PTR b._dns-sd._udp.0.64.9.161.in-addr.arpa
48	21.2530... 160.75.25.1	161.9.66.65		DNS	144	Standard query response 0xc36c No such name PTR db._dns-sd._udp.0.64.9.161.in-addr.arpa
49	21.2530... 160.75.25.1	161.9.66.65		DNS	144	Standard query response 0xe284 No such name PTR lb._dns-sd._udp.0.64.9.161.in-addr.arpa
50	21.2530... 160.75.25.1	161.9.66.65		DNS	138	Standard query response 0x17d8 No such name PTR 65.66.9.161.in-addr.arpa
51	21.2530... 160.75.25.1	161.9.66.65		DNS	153	Standard query response 0x4d8f No such name PTR lb._dns-sd._udp.0.64.9.161.in-addr.arpa

- ▶ Frame 42: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
- ▶ Ethernet II, Src: Apple\_65:30:c6 (88:e9:fe:65:30:c6), Dst: Cisco\_9f:f0:cc (00:00:0c:9f:f0:cc)
- ▶ Internet Protocol Version 4, Src: 161.9.66.65, Dst: 160.75.25.1
- ▶ User Datagram Protocol, Src Port: 51269, Dst Port: 53
- ▶ Domain Name System (query)

Picture 5 DCHP Protocol

## 2.DNS (Domain Name System Protocol)

Applications use understandable, easy remember names instead of IP addresses. So, we need a system to resolve and matches names with IP address. Basically, DNS works in the given figure below. Now we will analyze DNS Protocol with Wireshark.

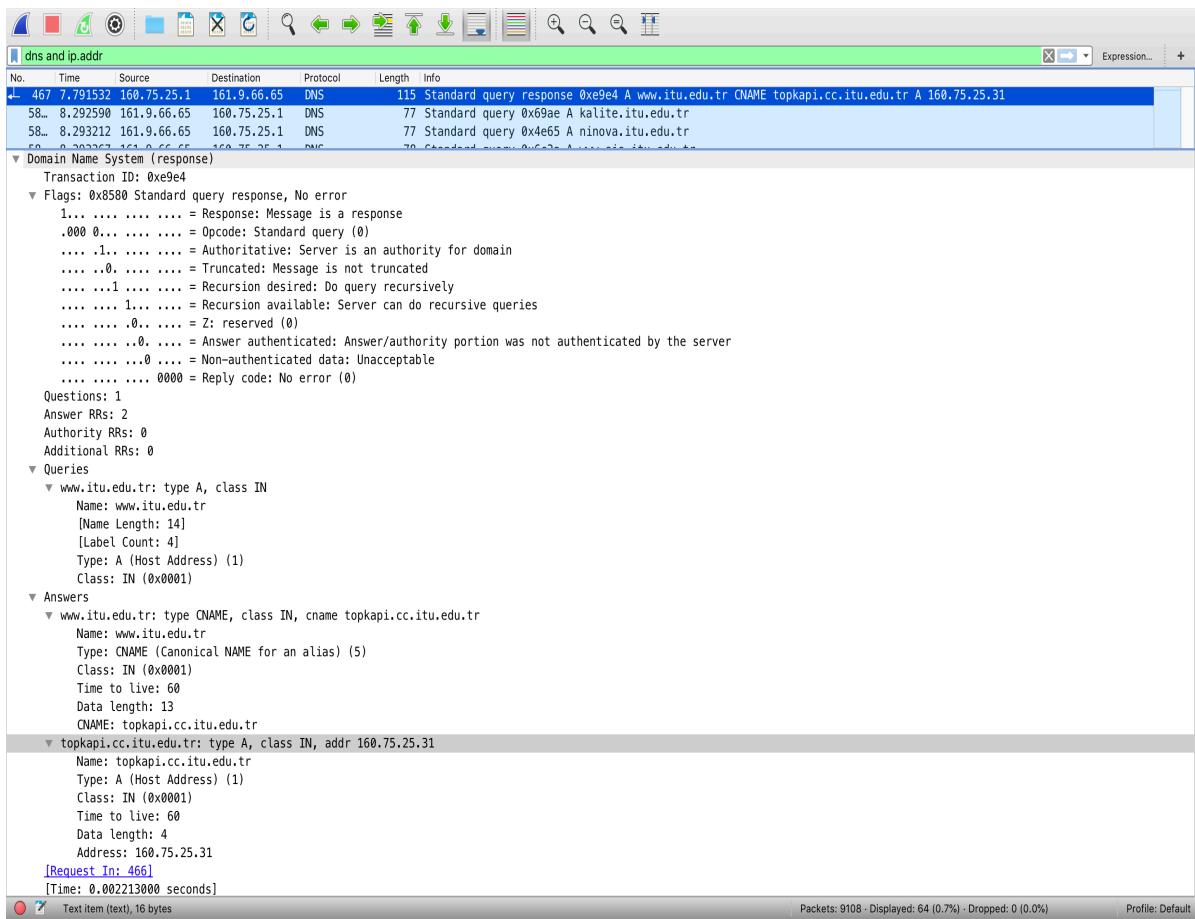


First, I deleted the history of the browser. After that I send a DNS query for www.itu.edu.tr. So, the 1<sup>st</sup> step completed and Picture 6 shows the query.

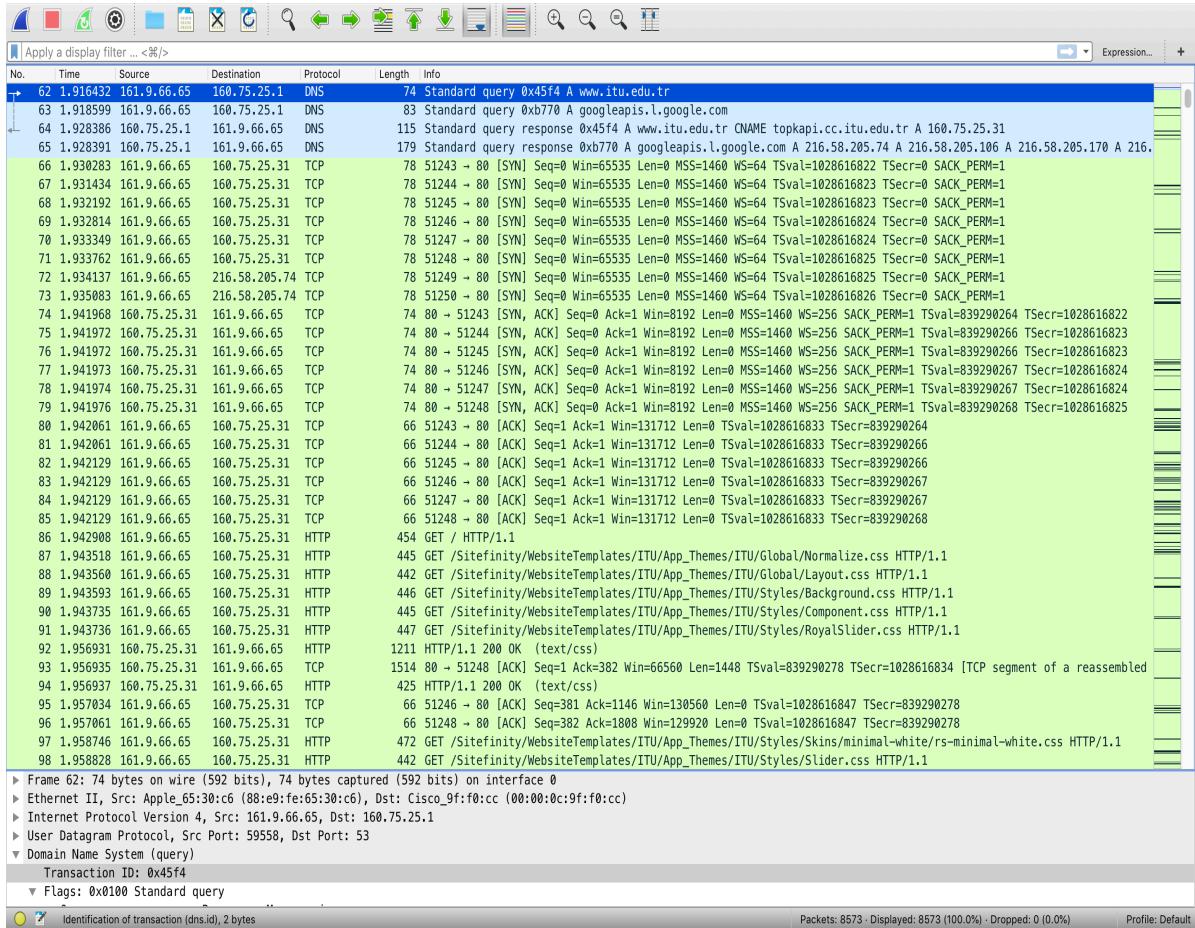
**dns and ip.addr**

No.	Time	Source	Destination	Protocol	Length	Info
377	6.406235	161.9.66.65	160.75.25.1	DNS	96	Standard query 0xc3f4 A googlehosted.l.googleusercontent.com
378	6.406529	160.75.25.1	161.9.66.65	DNS	112	Standard query response 0xc3f4 A googlehosted.l.googleusercontent.com A 216.58.198.1
4...	7.789319	161.9.66.65	160.75.25.1	DNS	74	Standard query 0xe9e4 A www.itu.edu.tr
467	7.791532	160.75.25.1	161.9.66.65	DNS	115	Standard query response 0xe9e4 A www.itu.edu.tr CNAME topkapi.cc.itu.edu.tr A 160.75.25.31
58...	8.292590	161.9.66.65	160.75.25.1	DNS	77	Standard query 0x69ae A kalite.itu.edu.tr
58...	8.293212	161.9.66.65	160.75.25.1	DNS	77	Standard query 0x4e65 A ninova.itu.edu.tr
58...	8.293367	161.9.66.65	160.75.25.1	DNS	78	Standard query 0x6c3a A www.sis.itu.edu.tr
61...	8.314807	160.75.25.1	161.9.66.65	DNS	133	Standard query response 0x6c3a A www.sis.itu.edu.tr CNAME uzay.sis.itu.edu.tr CNAME orion.sis.itu.edu.tr A 160.75
62...	8.320381	160.75.25.1	161.9.66.65	DNS	117	Standard query response 0x69ae A kalite.itu.edu.tr CNAME mozaik.cc.itu.edu.tr A 160.75.25.126
63...	9.320382	160.75.25.1	161.9.66.65	DNS	117	Standard query response 0x69ae A ninova.itu.edu.tr CNAME uzay.sis.itu.edu.tr A 160.75.25.146
Frame 466:	74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0					
> Ethernet II, Src: Apple_65:30:c6 (88:e9:fe:65:30:c6), Dst: Cisco_9f:f0:cc (00:00:00:9f:f0:cc)						
> Internet Protocol Version 4, Src: 161.9.66.65, Dst: 160.75.25.1						
> User Datagram Protocol, Src Port: 57370, Dst Port: 53						
▼ Domain Name System (query)						
Transaction ID: 0xe9e4						
▼ 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						
▼ Queries						
▼ www.itu.edu.tr: type A, class IN						
Name: www.itu.edu.tr						
[Name Length: 14]						
[Label Count: 4]						
Type: A (Host Address) (1)						
Class: IN (0x0001)						
(Response In: 467)						
0000 00 00 0c 9f f0 cc 88 e9 fe 65 30 c6 08 00 45 00 :.....e0..E						
0010 00 3c d2 89 00 00 ff 11 4c 90 a1 09 42 41 a0 4b <.....L...BA.K						
0020 19 01 e0 1a 00 35 00 28 c8 b3 e9 e4 01 00 00 01 .....5(.....						
0030 00 00 00 00 00 03 77 77 03 69 74 75 03 65 .....w ww:itu.e						
0040 64 75 02 74 72 00 00 01 00 01 du:tr....						

Picture 6 DNS Query with Wireshark



Picture 7 Source and Destination IP



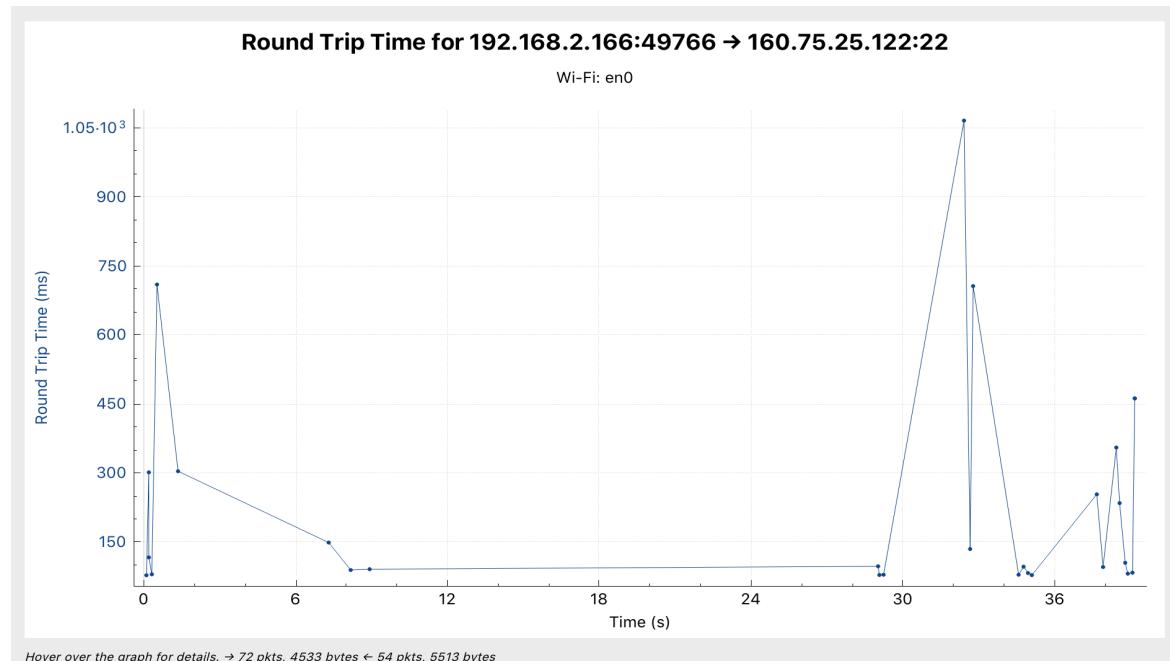
Picture 8 All Stages of DNS Protocol

In Picture 7 we can see that DNS server sent the IP address of the website by resolving it recursively. After that a TCP connection occurred between website and my computer. All in all, we can see the DNS server as 160.75.25.1 and IP address of website as 160.75.25.31 from screenshots.

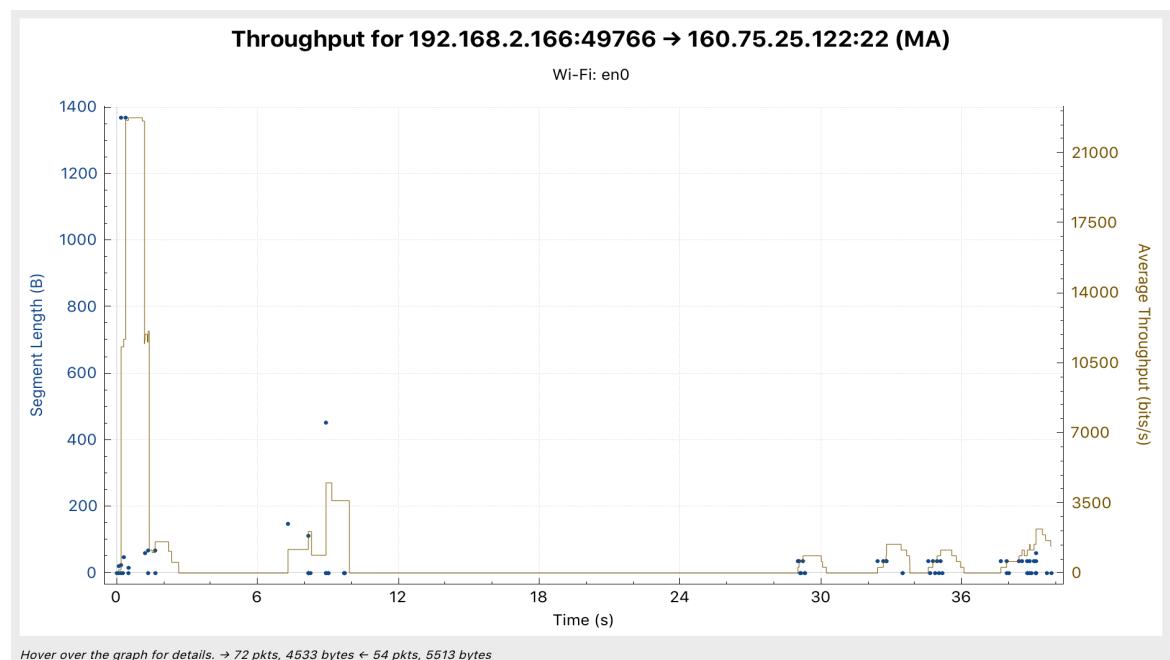
### 3.TRANSPORT LAYER PROTOCOLS

#### a.Capture A Session That Uses TCP

I know SSH uses TCP, so I created a TCP connection between ssh.itu.edu.tr and my computer. I started the Wireshark and made ssh login operation from terminal. After that I stopped the session by using logout command and results are:

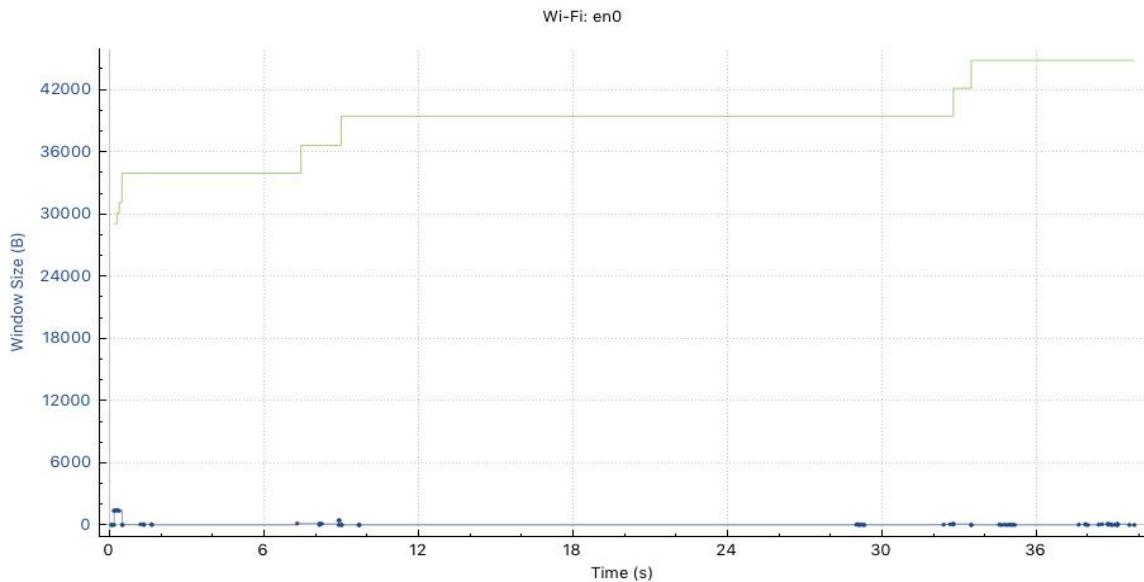


Picture 9 Round Trip Time



Picture 10 Throughput

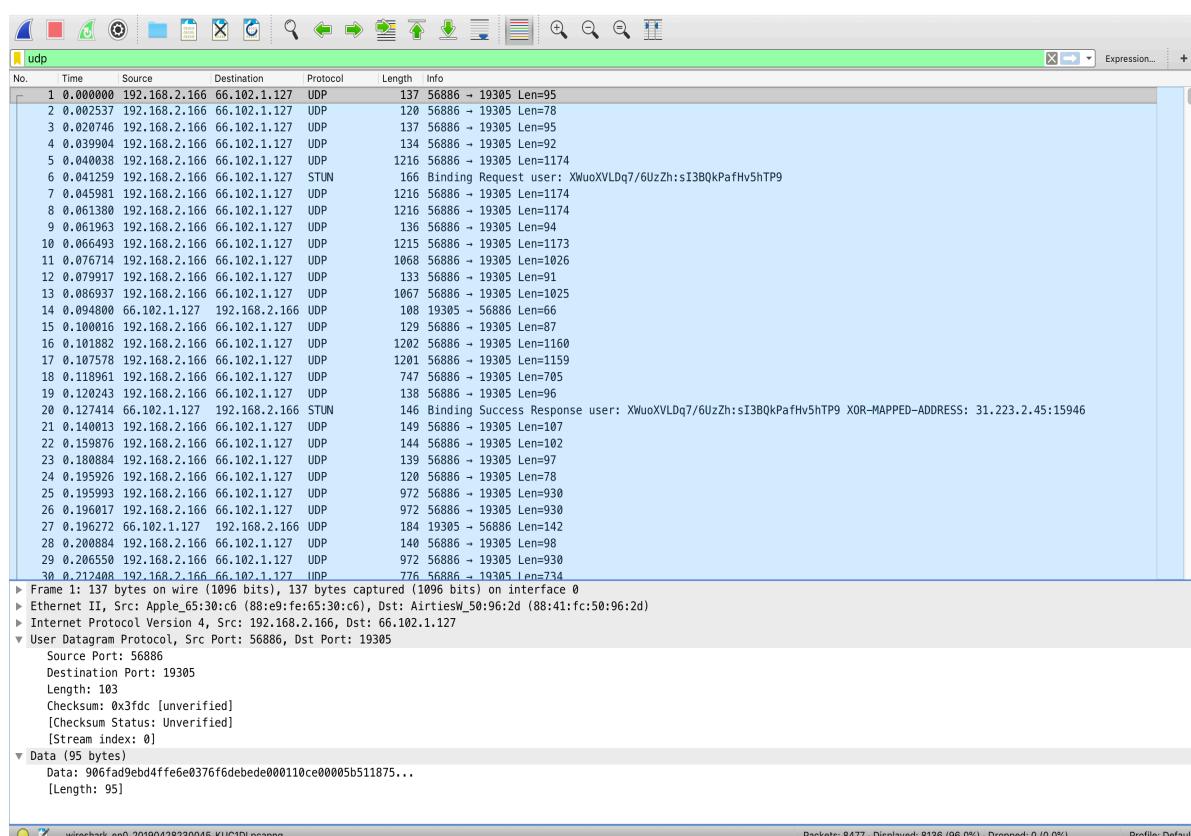
### Window Scaling for 192.168.2.166:49766 → 160.75.25.122:22



According to TCP Congestion Mechanism, when the network delivery becomes slow down, it gives warning to sender for slowing down. So according to congestion policy, in slow stop phase it exponentially increases window size. In congestion avoidance phase it increases 1 after reaching threshold. In congestion detection phase sender returns slow start and congestion avoidance phase.

#### b.Capture A Session That Uses UDP

I used Google Hangouts for capturing UDP packets. I started Wireshark and I called my friend then I used “udp” filter to see UDP packets.





Picture 12 IO Statistics For UDP

According to my researches, it is not proper to use statistical methods for UDP as in TCP. So, in this step for UDP connections Wireshark gives a practical way to analyze UDP packets named as I/O Graph.

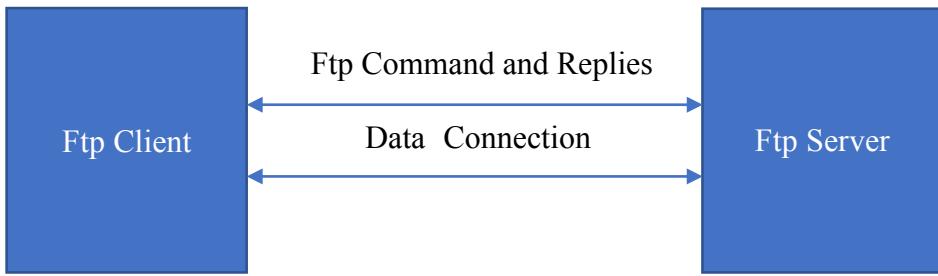
Best Effort Service in UDP means, network does not work for retransmit lost or corrupted packets. Using this method network provides more performance. In TCP connection between client and server guarantees the package delivery, so this causes checking packages and more cost. But in UDP method this is not a problem if package loss occurs.

### c.Any Other Transport Layer Protocol During Sessions?

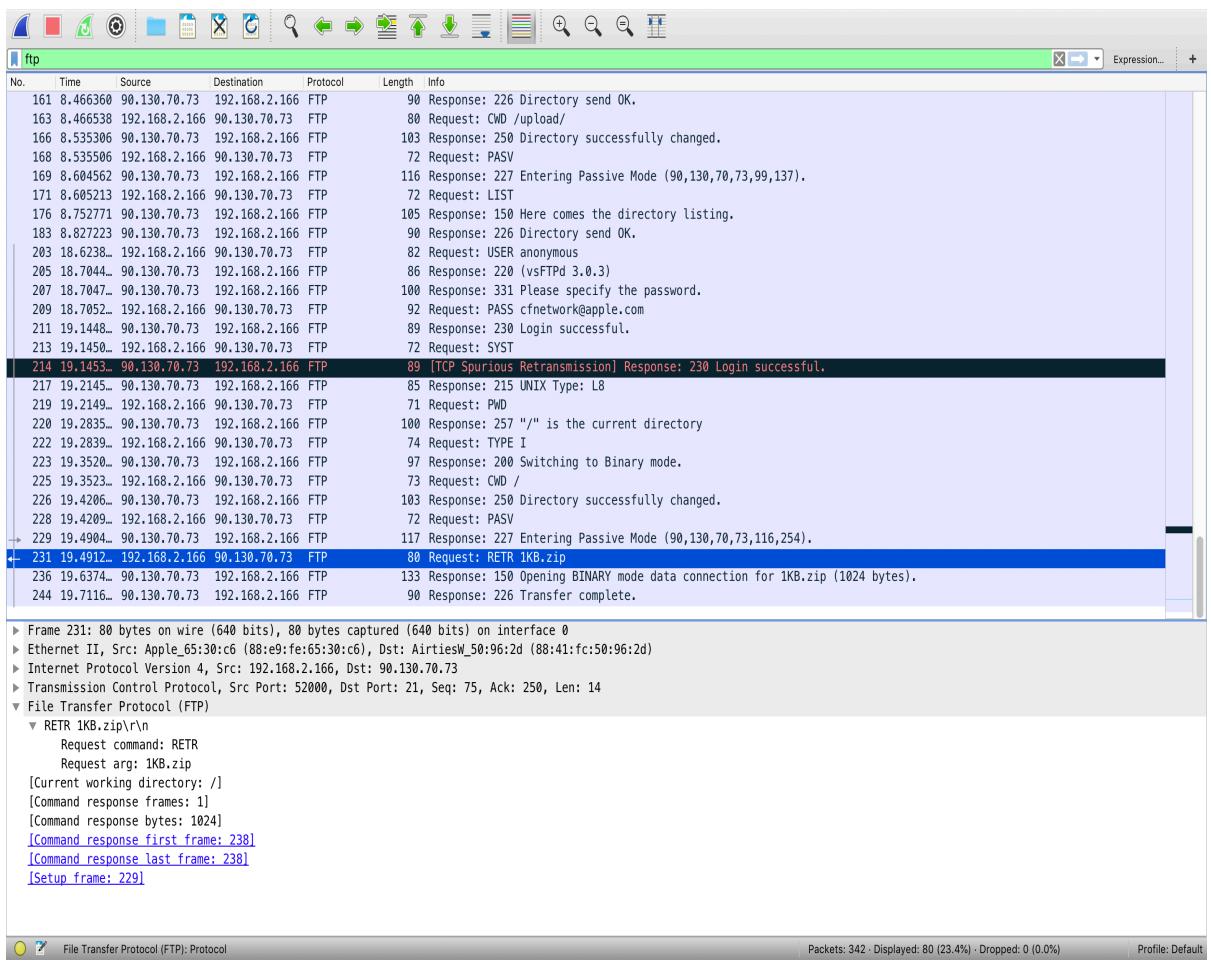
I made a detailed search and I found; AEP, AH, DCCP, ESP, FCP, NETBIOS, IL, ISCSI, NBF, SCTP, SINEC H1, SPX, NBP protocols near TCP and UDP in transport layer. But when I am using Wireshark I could not detected any other transport layer protocols except TCP and UDP.

## 4.FILE TRANSFER

When I tried to download a file from Ninova Wireshark does not show any FTP protocol it all includes TCP protocols. Because Http supports downloading file. The reason for this situation is FTP requires a new connection for each new data transfer, so this decreases the performance and increases the cost for a website like Ninova. So basically Http use request-response methodology in file transfer too.



The given Figure represents the Ftp connection and file downloading process. So we can easily see the process by using Wireshark.



Picture 13 Ftp Connection and File Download

## 5.PROTOCOL ANALYSIS INTERPRETATION

### a.How Many Different Protocols Captured?

DNS – TCP – HTTP – OCSP – TLSv1.2 – UDP – FTP – SSDP – ARP – STUN - ICMP

### b.What Are Specific Protocol Numbers For Transport Protocol?

TCP uses 6 as a protocol number.

```
▼ Internet Protocol Version 4, Src: 192.168.2.166, Dst: 2.18.82.217
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 64
    Identification: 0x0000 (0)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0x227f [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.2.166
  Destination: 2.18.82.217
  ▶ Transmission Control Protocol, Src Port: 52004, Dst Port: 443, Seq: 0, Len: 0
```

UDP uses 17 as a protocol number.

```
▼ Internet Protocol Version 4, Src: 66.102.1.127, Dst: 192.168.2.166
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 74
    Identification: 0xd7be (55230)
  ▶ Flags: 0x0000
    Time to live: 102
  Protocol: UDP (17)
  Header checksum: 0x75b1 [validation disabled]
  [Header checksum status: Unverified]
  Source: 66.102.1.127
  Destination: 192.168.2.166
  ▼ User Datagram Protocol, Src Port: 19305, Dst Port: 55594
    Source Port: 19305
    Destination Port: 55594
    Length: 54
    Checksum: 0xb5f4 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 3]
```

ICMP uses 1 as a protocol number.

```
▼ Internet Protocol Version 4, Src: 192.168.2.166, Dst: 216.58.215.228
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x9000 (36864)
  ▶ Flags: 0x0000
    Time to live: 64
  Protocol: ICMP (1)
  Header checksum: 0x773b [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.2.166
  Destination: 216.58.215.228
  ▶ Internet Control Message Protocol
```