BLG433E

COMPUTER COMMUNICATIONS

PROJECT 2

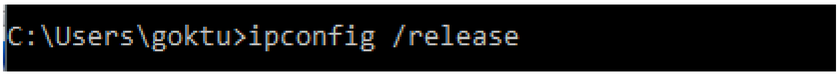
Wireshark Version: 2.6.5

AHMET GÖKTUĞ SEVİNÇ

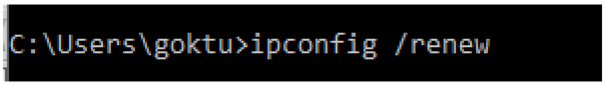
150140120

1) DHCP is a network management protocol that provides dynamically assignment of an IP address and other network configuration parameters to each device on a network. Here, DHCP server provides the dynamic IP addresses and each device is a DHCP client. This protocol uses ports 66 and 67.

In Wireshark, to be able to investigate DHCP protocol, we first release our IP configuration and, then renew it.

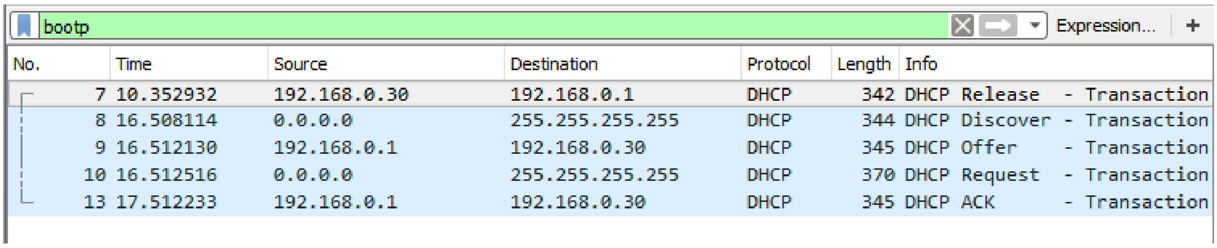


```
C:\Users\goktu>ipconfig /release
```



```
C:\Users\goktu>ipconfig /renew
```

To filter DHCP in Wireshark, we use *bootp* filter.



bootp

No.	Time	Source	Destination	Protocol	Length	Info
7	10.352932	192.168.0.30	192.168.0.1	DHCP	342	DHCP Release - Transaction
8	16.508114	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction
9	16.512130	192.168.0.1	192.168.0.30	DHCP	345	DHCP Offer - Transaction
10	16.512516	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction
13	17.512233	192.168.0.1	192.168.0.30	DHCP	345	DHCP ACK - Transaction

Between client and the DHCP server communication is provided with four messages.

Discover: First of all the DHCP client broadcasts a DHCPDiscover message on the network subnet using 255.255.255.255 destination address. It can also request its last known IP address. This message aims to find a DHCP server to request an IP address.

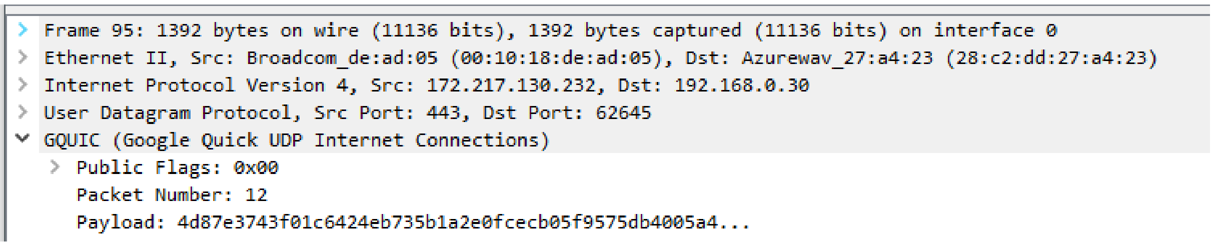
Offer: When the DHCP server receives DHCPDiscover message from the client, it reserves an IP address for the client and offers it to the client by using DHCPOffer message.

Request: In response to the DHCPOffer message, the client sends a DHCPRequest message to the server, indicating request of offered IP address.

ACK: After receiving DHCPRequest message, the DHCP server sends DHCPAck message to the client to acknowledge the client about the IP configuration process is completed.

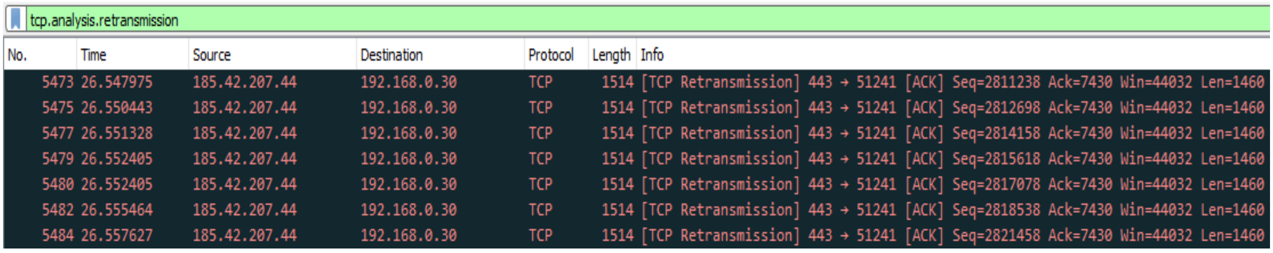
Note: In figure, also *Release* message is sent because of the *ipconfig /release* operation at the beginning. This message is sent by client to the server to inform the server that this IP address is deactivated by client.

2) For capturing video and audio traffics, I opened a *Youtube* video. The protocol that is used in that page was GQUIC (Google Quick UDP Internet Connecions) that is the Google's UDP. There were no retransmissions because it is a UDP connection.



```
> Frame 95: 1392 bytes on wire (11136 bits), 1392 bytes captured (11136 bits) on interface 0
> Ethernet II, Src: Broadcom_de:ad:05 (00:10:18:de:ad:05), Dst: Azurewav_27:a4:23 (28:c2:dd:27:a4:23)
> Internet Protocol Version 4, Src: 172.217.130.232, Dst: 192.168.0.30
> User Datagram Protocol, Src Port: 443, Dst Port: 62645
v GQUIC (Google Quick UDP Internet Connections)
    > Public Flags: 0x00
      Packet Number: 12
      Payload: 4d87e3743f01c6424eb735b1a2e0fcecb05f9575db4005a4...
```

Then, I opened *Twitch.tv* which is a web page providing live streaming. In that page TCP connection was used. Since this page uses TCP as the protocol, there were retransmissions.

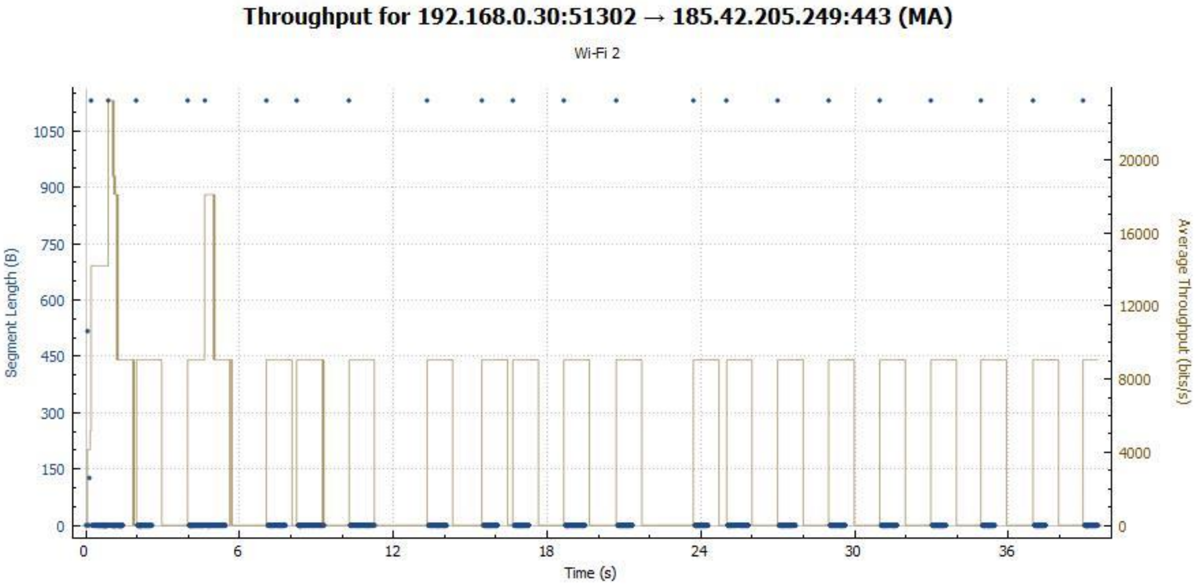


tcp.analysis.retransmission

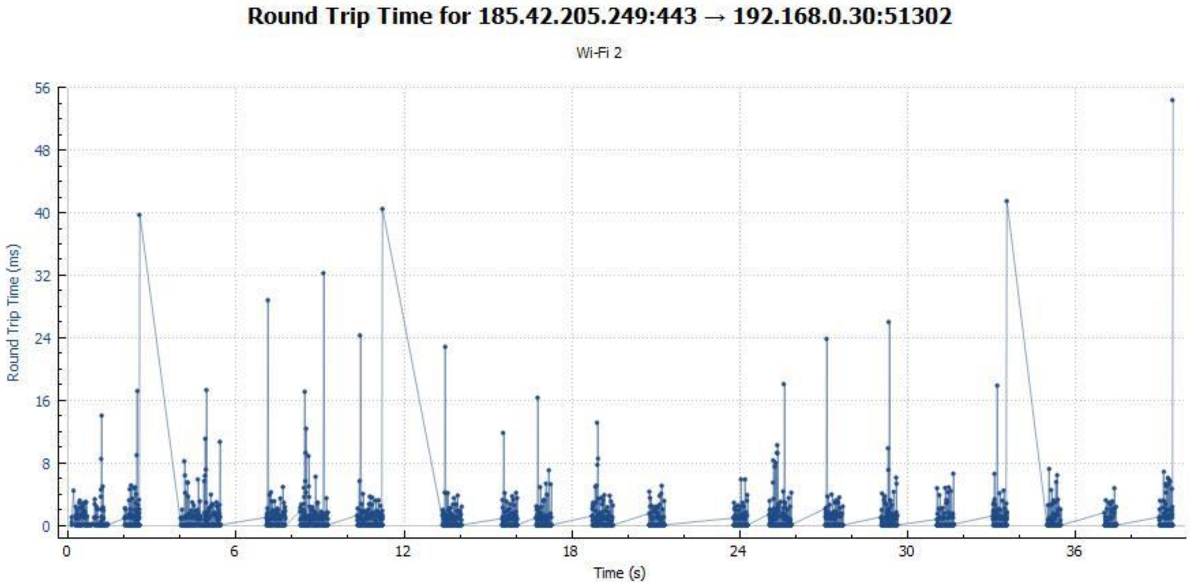
No.	Time	Source	Destination	Protocol	Length	Info
5473	26.547975	185.42.207.44	192.168.0.30	TCP	1514	[TCP Retransmission] 443 → 51241 [ACK] Seq=2811238 Ack=7430 Win=44032 Len=1460
5475	26.550443	185.42.207.44	192.168.0.30	TCP	1514	[TCP Retransmission] 443 → 51241 [ACK] Seq=2812698 Ack=7430 Win=44032 Len=1460
5477	26.551328	185.42.207.44	192.168.0.30	TCP	1514	[TCP Retransmission] 443 → 51241 [ACK] Seq=2814158 Ack=7430 Win=44032 Len=1460
5479	26.552405	185.42.207.44	192.168.0.30	TCP	1514	[TCP Retransmission] 443 → 51241 [ACK] Seq=2815618 Ack=7430 Win=44032 Len=1460
5480	26.552405	185.42.207.44	192.168.0.30	TCP	1514	[TCP Retransmission] 443 → 51241 [ACK] Seq=2817078 Ack=7430 Win=44032 Len=1460
5482	26.555464	185.42.207.44	192.168.0.30	TCP	1514	[TCP Retransmission] 443 → 51241 [ACK] Seq=2818538 Ack=7430 Win=44032 Len=1460
5484	26.557627	185.42.207.44	192.168.0.30	TCP	1514	[TCP Retransmission] 443 → 51241 [ACK] Seq=2821458 Ack=7430 Win=44032 Len=1460

3) After watching the stream from '*Twitch.tv*', I determined the packets and found its stream by *Right click -> Follow -> TCP Stream* options. After that operation TCP connection stream is filtered and I used Statistics tab to draw neccessary graphs.

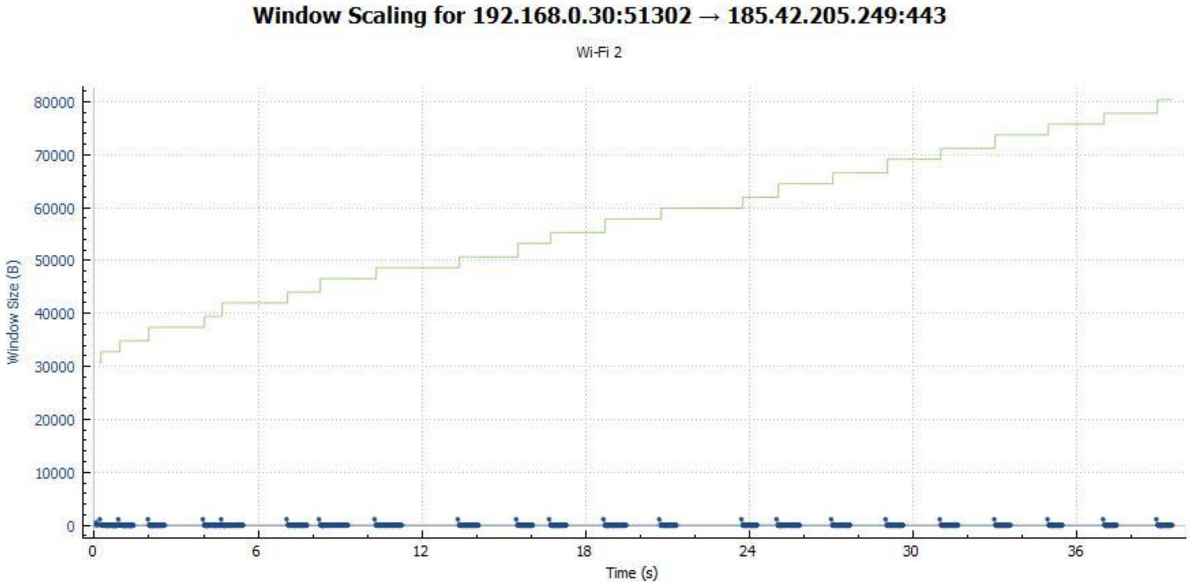
Throughput:



RTT:

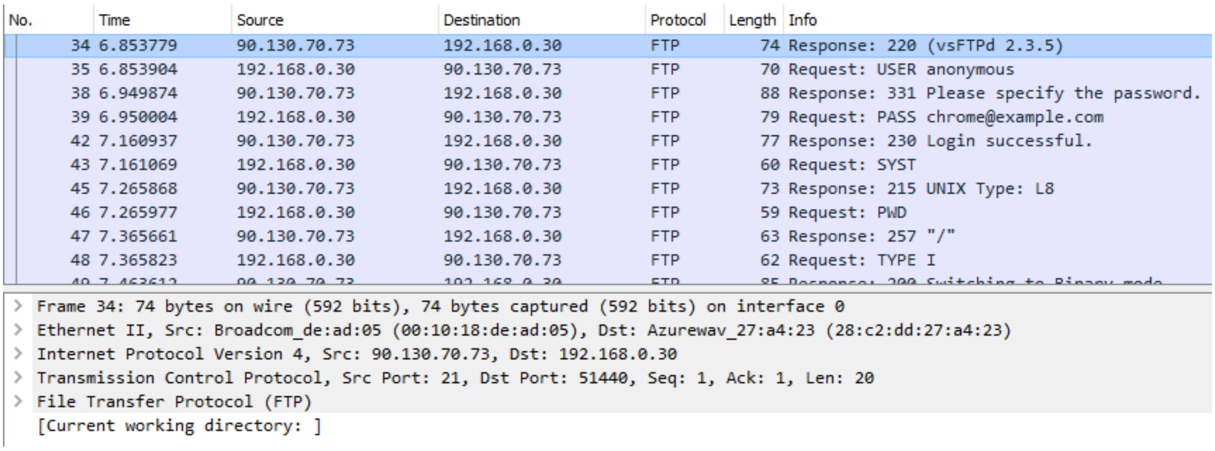


Window Size:



In TCP, there is a congestion control mechanism that limits senders' load rate. TCP perceives congestion in case of packet drops. So, in TCP senders sends more and more packets until congestion occurs and after detection of congestion, TCP limits their sending rate. Here, in our RTT graph, we can see that at some certain points, RTT increased a lot. That means ACK was not received for long period of times and possible reason for that event is the package loss. At that point, protocol limits the throughput of the sender. As we can see in throughput graph, throughtput keeps increasing until congestion occurs and, after that point, it starts to decrease.

4) In regular dowlands HTTP is used for file transferring. Because of that, we can't observe FTP while dowlanding a regular file from the browser. For this reason, we need a server that proved FTP as you supplied in homework.



```
No.   Time        Source          Destination     Protocol  Length  Info
   34 6.853779    90.130.70.73    192.168.0.30    FTP           74  Response: 220 (vsFTPd 2.3.5)
   35 6.853904    192.168.0.30    90.130.70.73    FTP           70  Request: USER anonymous
   38 6.949874    90.130.70.73    192.168.0.30    FTP           88  Response: 331 Please specify the password.
   39 6.950004    192.168.0.30    90.130.70.73    FTP           79  Request: PASS chrome@example.com
   42 7.160937    90.130.70.73    192.168.0.30    FTP           77  Response: 230 Login successful.
   43 7.161069    192.168.0.30    90.130.70.73    FTP           60  Request: SYST
   45 7.265868    90.130.70.73    192.168.0.30    FTP           73  Response: 215 UNIX Type: L8
   46 7.265977    192.168.0.30    90.130.70.73    FTP           59  Request: PWD
   47 7.365661    90.130.70.73    192.168.0.30    FTP           63  Response: 257 "/"
   48 7.365823    192.168.0.30    90.130.70.73    FTP           62  Request: TYPE I
   49 7.463612    90.130.70.73    192.168.0.30    FTP           85  Response: 200 Switching to Binary mode

> Frame 34: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: Broadcom_de:ad:05 (00:10:18:de:ad:05), Dst: Azurewav_27:a4:23 (28:c2:dd:27:a4:23)
> Internet Protocol Version 4, Src: 90.130.70.73, Dst: 192.168.0.30
> Transmission Control Protocol, Src Port: 21, Dst Port: 51440, Seq: 1, Ack: 1, Len: 20
> File Transfer Protocol (FTP)
  [Current working directory: ]
```

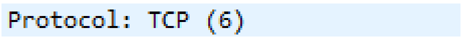
In this FTP communication, firstly client sends username (USER) to the server, then server requires a password, after sending password (PASS), server responds if login was successfull or not. Then, client sends system information (SYST) and server returns acknowledgement. After these operations clients asks for working directory (PWD) and server returns current working directory. Then, client sets transfer type (TYPE), and server returns response. After that, client asks for size of the file (SIZE) and server returns file size. Client requests for working directory change for dowland operation (CWD) and if directory change is successfull client finally retrieves the file (RETR) and quits (QUIT) from communication.

5)

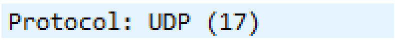
a)

- TCP
- DNS
- UDP
- HTTP
- FTP
- TELNET
- SMTP
- ICMP
- BOOTP
- ANCP
- IMAP
- NTP
- SSH
- NNTP
- DCCP
- RTP
- RTCP

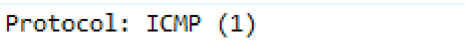
b)



```
Protocol: TCP (6)
```



```
Protocol: UDP (17)
```



```
Protocol: ICMP (1)
```