BLG433E

COMPUTER COMMUNICATIONS

PROJECT 2

Wireshark Version: 2.6.5

AHMET GÖKTUĞ SEVİNÇ

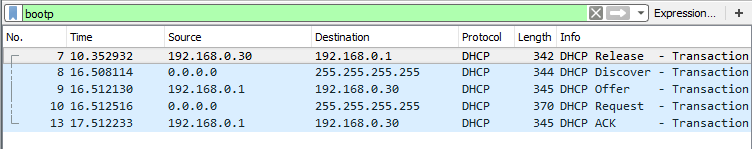
150140120

**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.



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



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.

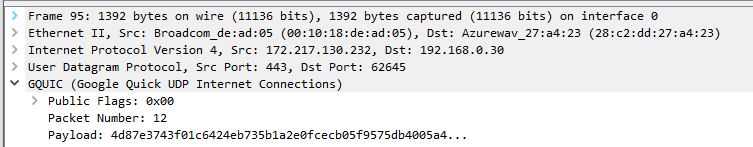
**Offer:** When the DHCP server receives DHCPDiscover message from the client, it reserves an IP address for the client and offers it to the client by using DHCPOffer message.

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

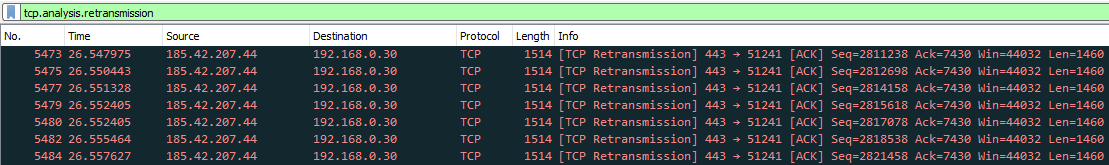
**ACK:** After receiving DHCPRequest message, the DHCP server sends DHCPAck message to the client to acknowledge the client about the IP configuration process is completed.

**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.

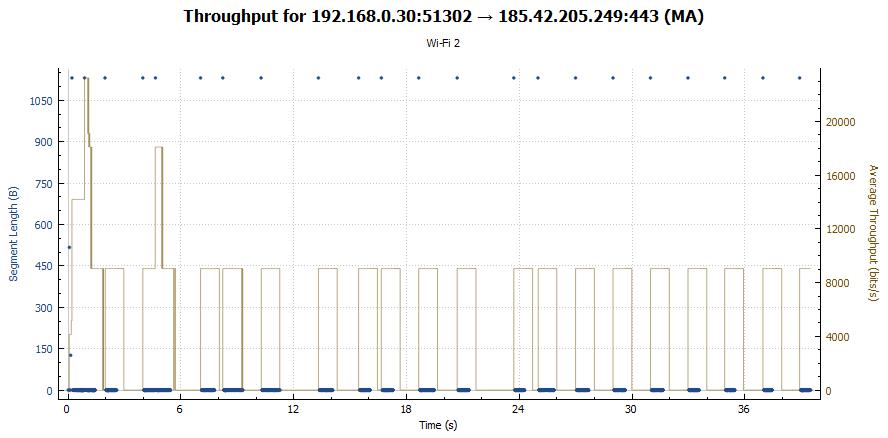


Then, I opened *Twitch.tv* which is a web page providing live streaming. In that page TCP connection was used. Since this page uses TCP as the protocol, there were retransmissions.

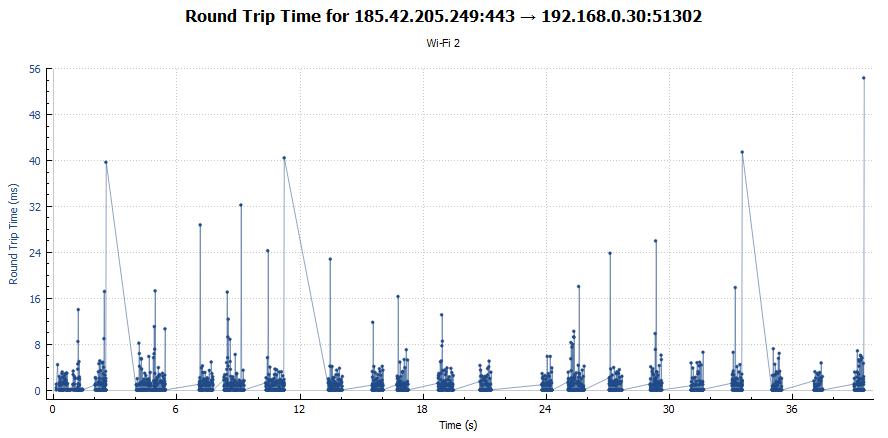


**3)** After watching the stream from ‘*Twitch.tv*’, I determined the packets and found its stream by *Right click -> Follow -> TCP Stream* options. After that operation TCP connection stream is filtered and I used Statistics tab to draw neccessary graphs.

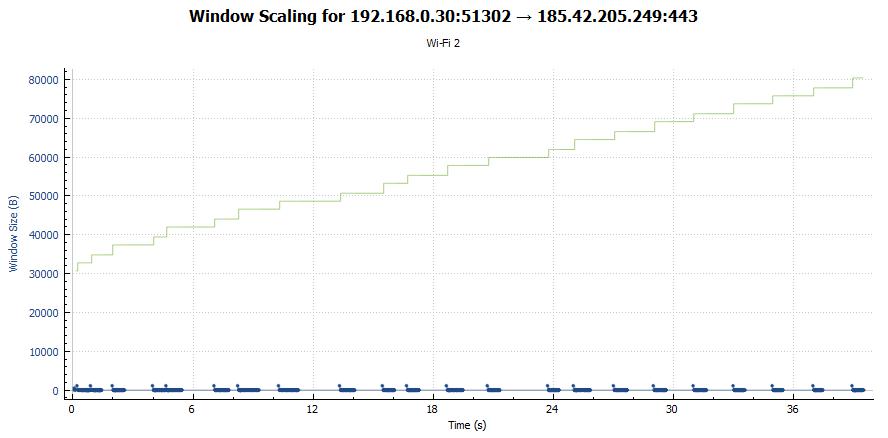
**Throughput:**



**RTT:**

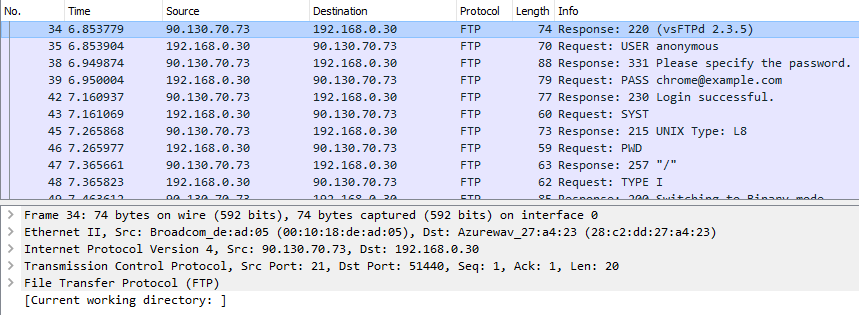


**Window Size:**



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.



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)**

