CS 39006: Networks Lab

Assignment 4: Basic Socket Programing - UDP (Working with A Single Threaded File Transfer Application)

Report by:

Adha Ranjith Kumar(15CS30002)

Chittepu Harsha Sai Reddy(15CS30011)

Penchala Akhil(15CS30026)

Client/Server Using UDP Socket

Steps:

1. Compile both udpserver.c and udpclient.c using command **make**.
2. Change directory to Server/. Run udpserver using command
   ./udpserver <port number>
3. Open another terminal and Change directory to Client/. Run udpclient using command
   ./udpclient <host address> <port number> <file name>
4. Open wireshark and set filter according to host address and your IP address to display packets. Go To to Statistics->Packet Lengths to get packet lengths and Statistics->FlowGraph for total time of file transfer.

Outline of Protocol Working:

a) The client first informs the filename and file size to the server by sending the hello message.
b) The server acknowledges the hello message.
c) The client forwards the file data over the datagram socket to the server.
d) The server receives the data,reconstructs the file at the server side,creates the md5 checksum of the entire file.
e) The server acknowledges the file with the md5 checksum of the the file.
f) The client creates the MD5 checksum of the original file before transfer, and matches it with the received MD5 checksum from the server. The client prints a message at the console “MD5 matched” or “MD5 not matched” and exists.

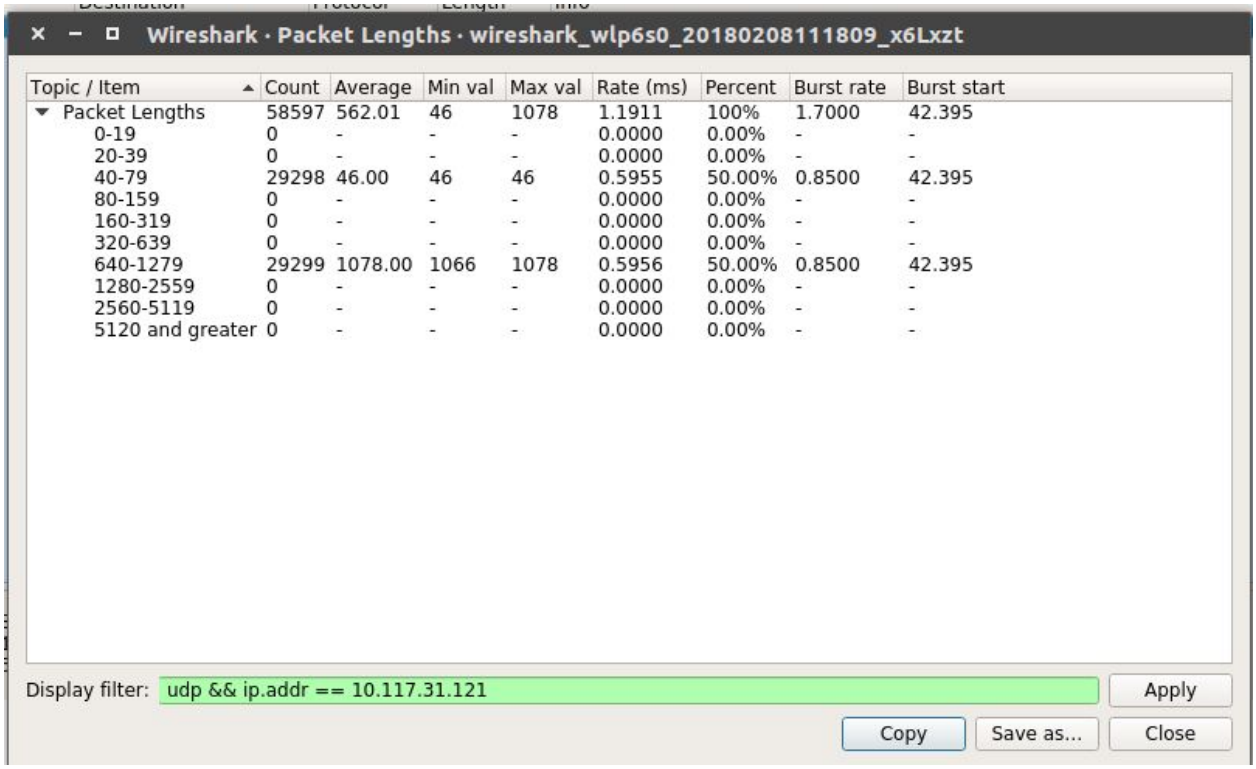
Steps taken to ensure reliability at Application Layer :

1. Divide the file into chunks of 1 KB (last chunk can be less than 1 KB).
2. Inform the server about the file name , the total file size and the number of chunks to be sent
3. Add an application header to every chunk. Each header should contain following fields,
    a. Sequence number - 4 Bytes
    b. Length of the chunk - 4 Bytes

   This info will help in ensuring reliability.
4. Forward the chunks using Stop and Wait protocol with timeout=1 second, to ensure the reliability of data transfer at the application layer.

Observations :

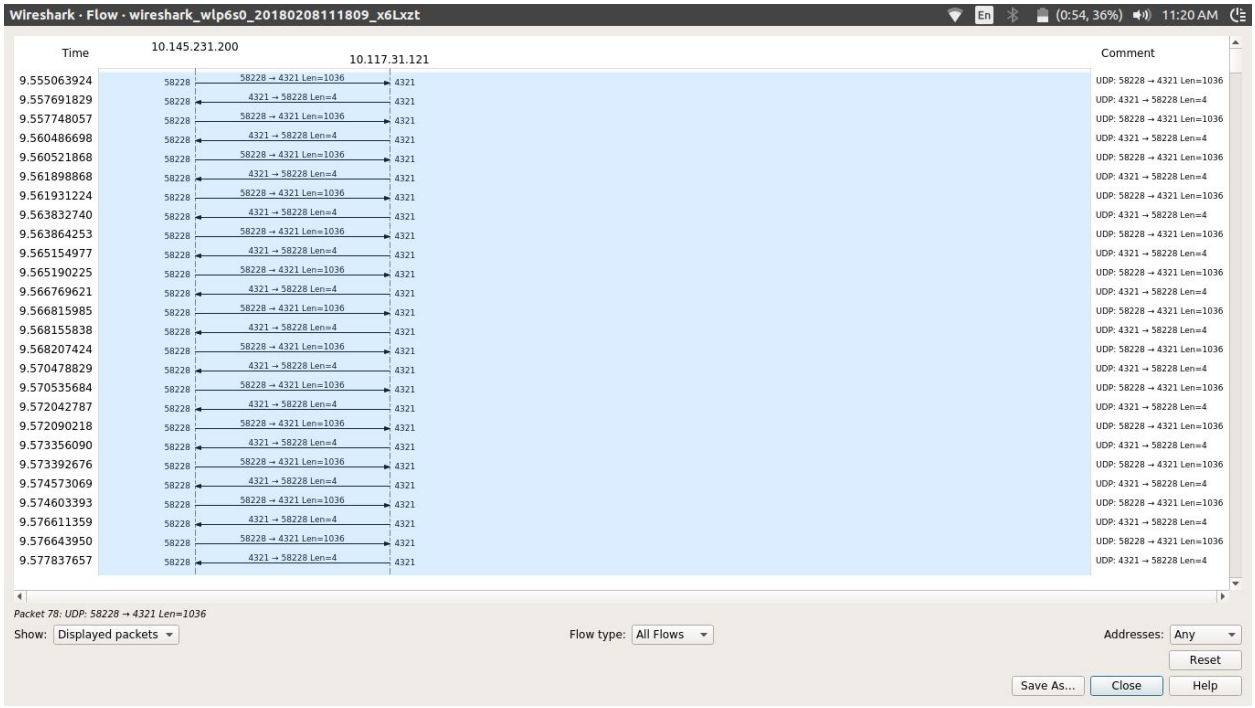
1) Packet size distribution



2)

Total number of retransmitted packets = 0

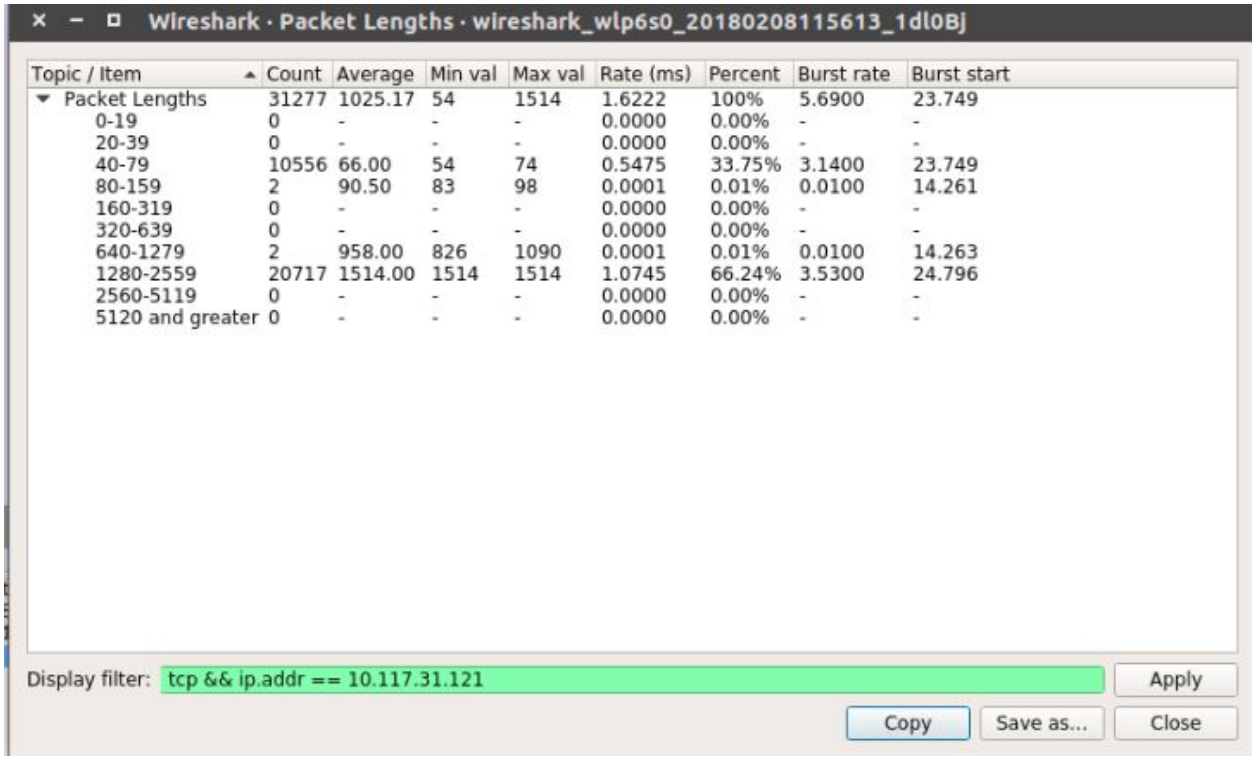
3)



Total time to receive the packet = 58.75068 - 9.55506 = 49.19562s

Comparisons with TCP : (* input - 30 mb random text file)

- Packet distribution for TCP :



- Number of retransmitted packets = 0

- Total Time taken = 25.214 - 5.933 = 19.281s

Justifications:

- Total time taken for UDP is higher than TCP as we are using Stop and Wait ARQ protocol. As next packet is only sent after it receives an acknowledgement, a lot of time is wasted waiting for ACK.
- Packet size distribution varies as the header field attached by UDP(at application layer) and TCP are of different size, UDP being the smaller.
- In UDP, we are sending ACK for every packet, so count of ACKs and Data packets are same(precisely 1 more data packet because of last md5sum sent by the server). Where as in TCP, ACK can be jointly given for 1 or more data packets. So, count of ACKs and Data Packets are far apart.