

Assignment 3: Basic Socket Programing (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 TCP Socket

Steps:

1. Compile both tcpserver.c and tcpclient.c using command **make**.
2. Change directory to Server/. Run tcpserver using command **./tcpserver <port number>**
3. Open another terminal and Change directory to Client/. Run tcpclient using command **./tcpclient <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.

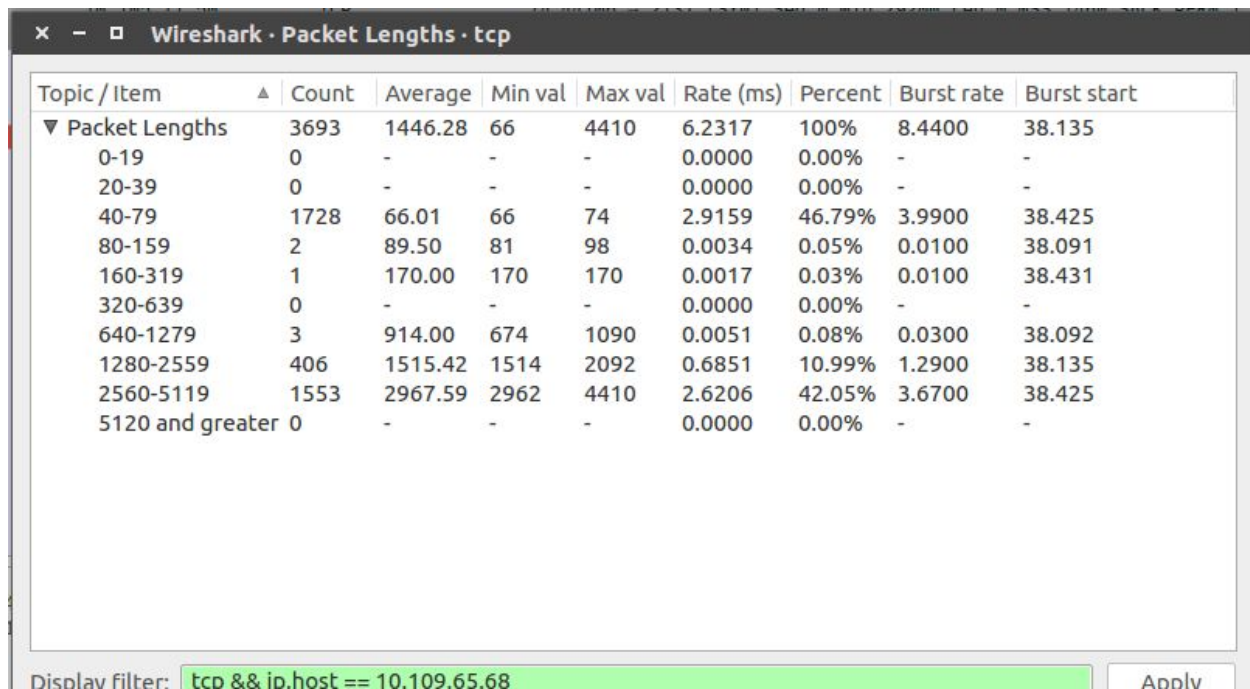
Protocol Working:

- a) The client first informs the filename and filesize to the server by sending the hello message.
- b) The server acknowledges the hello message.

- c) The client forwards the file data over the stream 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.

Observations:

1. Total number of segments received for tcp and their size distribution:



The screenshot shows the Wireshark 'Packet Lengths' window for TCP. The table displays the distribution of packet sizes, with the most frequent size being 40-79 bytes (1728 packets, 46.79% of total).

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
▼ Packet Lengths	3693	1446.28	66	4410	6.2317	100%	8.4400	38.135
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1728	66.01	66	74	2.9159	46.79%	3.9900	38.425
80-159	2	89.50	81	98	0.0034	0.05%	0.0100	38.091
160-319	1	170.00	170	170	0.0017	0.03%	0.0100	38.431
320-639	0	-	-	-	0.0000	0.00%	-	-
640-1279	3	914.00	674	1090	0.0051	0.08%	0.0300	38.092
1280-2559	406	1515.42	1514	2092	0.6851	10.99%	1.2900	38.135
2560-5119	1553	2967.59	2962	4410	2.6206	42.05%	3.6700	38.425
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Display filter: `tcp && ip.host == 10.109.65.68`

Total number of packets = 3693.

2. Total number of retransmitted segments for TCP = 0
3. Total time to receive the file for TCP:
38.658 - 38.065 = 0.593 sec

Justifications:

1. Small packet lengths are acknowledgement packets. We set the size of each chunk of data we sent to server as 1024 bytes, so when the data is attached with protocol headers, we get the size in the range of 1280-2559.
2. We have a closed connection between server and client and because of this there is less congestion in the network and no packet has been lost. Whereas in the case of previous assignment where a server is accessed by different clients, congestion is very high and packets have been lost.