Computer Networks, Fall 2020
Instructor: Jitendra Bhatia

Socket Programming

Task 2: Simple file transport protocol over UDP

Modify the file transport protocol over UDP which is already shared as fileudpserver.c and fileudpclient.c.

The protocol between the client and server is as follows.

1) Server will take the port on which it will listen as a command line argument.

2) The client program is started (server IP and port is provided as a command line argument) then Client sends a request packet (RRQ) to the server. The request will contain the file name that it wants to fetch from the server.

3) Serve should send the data packet where Each DATA packet will have a sequence number (starting from zero) and will be fixed size (512 bytes). Only the last packet will have a size less than 512 (it could be zero if the file size is a multiple of 512 bytes). This last packet will signal the end of file.

4) In case the server cannot transfer the file (e.g., file doesn't exist), the server will send an ERROR message (in response to the RRQ message from the client).

Note:

Using UDP Sockets

TCP and UDP have many similarities but some key differences too.

● You will no longer use SOCK_STREAM socket type, use SOCK_DGRAM instead.

● You will use recvfrom() and sendto() instead of recv()/read() and send()/write().

● Unlike TCP which may break your message into smaller pieces (or combine smaller

pieces into large ones), UDP will preserve message boundaries. So two writes (sendto())

calls on the sender will require two reads (recvfrom()) calls at the receiver. This will

actually simplify your message processing (compared to TCP).