The part1 is designed as following:

For the server:
Server is first waiting for a new client request of a file and check if the path is valid and then fork the child process and create a new socket bind with new port for file transmission. And the server reads PACKET_SIZE bytes everytime from the file and store the WINDOW_SIZE packets in the sliding window. Server first sends the packets in the sliding window when window is full for the first time and uses a timeout to resend the initial window if all the packets are lost.
And then it does not send anything until it receives some ACKs from the client and window stay the same position. When server receives the negative ACK, it will simply send the missing packet inside the sliding window. When server receives a positive ACK, it will check if the ACK matches the current position sliding window and adjust the sliding window to correct position since server knows that the packets before the positive ACK are all collected by client. Server will reach the end of file when we cannot read anything from the file descriptor and if the we receive a positive ACK that equals the file length then we can tell that client got all the packets and server will send a packet that indicates end of file and wait for a timeout to end connection in case the lost of packet.

For the client:
Client will first send the file path and if it is valid, it will start waiting for incoming packets from server, and we use the timeout to resend the negative ACKs, it constantly check for the missing packets for the window and send the negative ACKs to server. Once it receive one packet, it check the packet number to see if the packet is the starting of the sliding window if so we can move the sliding window one position right and when this packet got lost, it will resend the positive packet in the timeout handler. The positive ACK means that until the ACK position, the previous packets are collected by client so that server can adjust the sliding window to correct position since server knows that the packets before the positive ACK are all collected by client.

For example:
Server: [0 1 2 3] 4
Client: 0 [x 2 3 4]
if positive ACK 0 is lost, then sliding window of client is move to right and 1 is lost from server then client will resend the positive ACK of 0 in timeout and negative ACK of 1. If server get 0 then it will become
Server: 0 [1 2 3 4]
Client: 0 [x 2 3 4]
Once client got 1 then it will shift to 0 1 2 3 4 [x x x x] and the timeout will send the missing packets and repeat since server only send packet when got client negative ACK so the packet lost by server will be handled by client timeout.