Issues

1. The format of command-line arguments, with the exception of the client name, are not confirmed. Can be solved by iterating over the command-line arguments and using C's built-in functions such as isascii(), isdigit(), isspace(), etc.
2. Server assumes that the messages it receives are in the correct format. Can be solved by adding a function that checks the format, as was done on the client side.
3. There's no signal handler for handling sudden crashes. Memory is not freed nor sockets closed if the program crashes midway. This can be solved by implementing a signal handler.
4. Currently the server is closed by sending the message "QUIT SERVER" to the server, from a client. This was done for simplicity as I only needed to add a line or two of code. A better way to do it would be to either implement a signal handler or use select() and listen on STDIN for a line such as "QUIT SERVER" on the server-side.

send_packet.c modifications

set_loss_probability() : divide the input number by 100.0f, also call the function srand48().
send_packet() : This function actually doesn't work by itself, which is not stated in the task paper. drand48() which is used as the RNG requires srand48() to be called beforehand. This is done in my modification of set_loss_probability().

The program - upush_server.c

Line 175: Start of main()
Line 176-216: Setup
Line 217: Start of the main loop.
Line 218-230: Receive input and tokenize it.
Line 232-238: If it's a registration message.
Line 240-252: If it's a lookup message.
The server uses a linked list of client structs to store information about each client, this was done for simplicity. Whether or not clients have gotten too old is checked when a lookup of that client is received. This avoids unnecessary checks. select()'s timeout could've been used if avoiding clients taking up unnecessary space is more important.

The program - upush_client

Line 709: Start of main()
Line 710-789: Setup
Line 790: Start of the main loop.
Line 797-840: If select() detects user input (STDIN).
Line 842-869: If select() detects a message was received (socket).
Line 870-875: If select() times out.
The blocklist is a linked list. The message queue is a linked list of client structs, and each client struct contains a linked list of messages. The select() in the main loop has a timeout of 1 second. Every timeout it will check whether or not any messages need to be resent or given up on and if it needs to send a heartbeat. Having a 1 second timeout in the main loop makes it so that the actual timeout from the command-line or the 10 second heartbeat won't be delayed. This also gives the server enough time to respond when the timeout is 0.