

## **Lab 8**

*Robin Meyler*

### **1) What is the major differences between TCP client and TCP server implementations?**

- The Server and Client have different sockets, the server has to listen for the clients to react with a listen socket and keep track of the clients with sockets while the client just needs a connection to the server
- The server sets up a TCP server with the bind() command with IP address information
- The server then listens for connections to be send from the client(s) It will wait in this state unless told not to until a connection is made with Listen() call. In the example provided, it will stop here and wait.
- The Client attempts to connect to the address of the server until it receives back a result confirmation. In the given example it loop till it finds a socker it can connect to.
- The server will keep track of each connection from the clients with IDs and address information, it is the servers responsibility to accept messages from all the clients and process this information then send the information as well as any implications of this information to the other clients so they can update their own version of state.
- The client merely has to keep track of it's internal state, any local changes are sent to the server.
- In the given example for reading the server waits to receive messages with recv(). It takes the information as a message, the message length and the connection that is sending it, sends a message back to the client with similar information.

### **2) What is the major differences between UDP server and TCP server implementations?**

- TCP socket uses a 3 way handshake to establish correct information transfer between he server and the clients by establishing Acknowledgements
- UDP does not use this extra level assurance. Instead, UDP clients throws information at the server and it's the sole responsibility to catch the packets of information and the clients does not look for nor expect a confirmation of this information's transfer.
- UDP Server/Client setup suits certain applications. For example, if you were sending something to the server such as a soccer match stream, not every single packet sent to the server about the footage is essential. The server can send the stream to however many people, if you're watching something like this a lost packet or 2 does not affect the experience and you might not even notice. The advantage of this approach means that it is much faster for the users and is ideal for things like streaming.
- TCP instead will resend a packet if it receives back information that it was not caught by the server. While this would make some applications much worse from a user stand point, other applications such as sensitive information of banking and server/client relationships to do with money or medical use. In cases like these, it is very important that every packet of information is sent and it is usually important that the information is sent in order