

## Protocol Requirements Documentation

### DFA Node locations in code:

Client waits for server pong	Client.cpp 200-217
Client waits for version number	Client.cpp 235-243
Server waits for version selection	Server.cpp 167-179
Server awaiting Client Info	Server.cpp 182-196
Connected	Client.cpp 342-403
Chat server mode	Client.cpp 408-550

### DFA edge locations in code:

Client initiates connection with server by sending a PING to the server	Client.cpp 175-197
Server responds to this ping with a PONG	Server.cpp 131-142
Server sends highest VERSION_NEGOTIATION	Server.cpp 147-156
Client sends VERSION_NEGOTIATION	Client.cpp 247-283
Timeouts at any of the first 4 nodes send the DFA state back to IDLE	Client.cpp 220-227, 328-336   Server.cpp 110-115,
Client sends CLIENT_INFO	Client.cpp 285-319
Client sends JOIN_CHANNEL	Client.cpp 374-403
Client sends SEND_MESSAGE	Client.cpp 472-492
Client sends QUERY	Client.cpp 499-510
Server sends RECEIVE_MESSAGE	Server.cpp 275-288
Client sends LEAVE_CHANNEL	Client.cpp 446-472
Client sends QUIT	Client.cpp 347-372, 421-446

### Miscellaneous protocol requirements:

- Runs on TCP port 29200 Client.cpp 109, Server.cpp 341
- Usernames may not be longer than 10 characters Client.cpp 289-304
- Channel names may not be longer than 50 characters Client.cpp 385

The standard message format for PDU's are a 256 byte prefix, 4 byte command, and 2KB command parameters [Client.cpp 164-167](#) [Server.cpp 88-91](#)

Server stores a running history of all chat messages [Server.cpp 31-39, 245-253](#)

Accounting for byte-ordering [Server.cpp 170-172 \(not met\)](#)

Server relays chat messages to all recipients [Server.cpp 242-289](#)

However, server does not send the same chat messages back to the clients who sent them

[Server.cpp 251](#)

The client ignore chat messages from channels it has not joined [Client.cpp 538](#)

If a client sends the !quit command, the server closes the socket used with that client

[Server.cpp 213-221, 298-304](#)

*Extra Credit was not completed.*