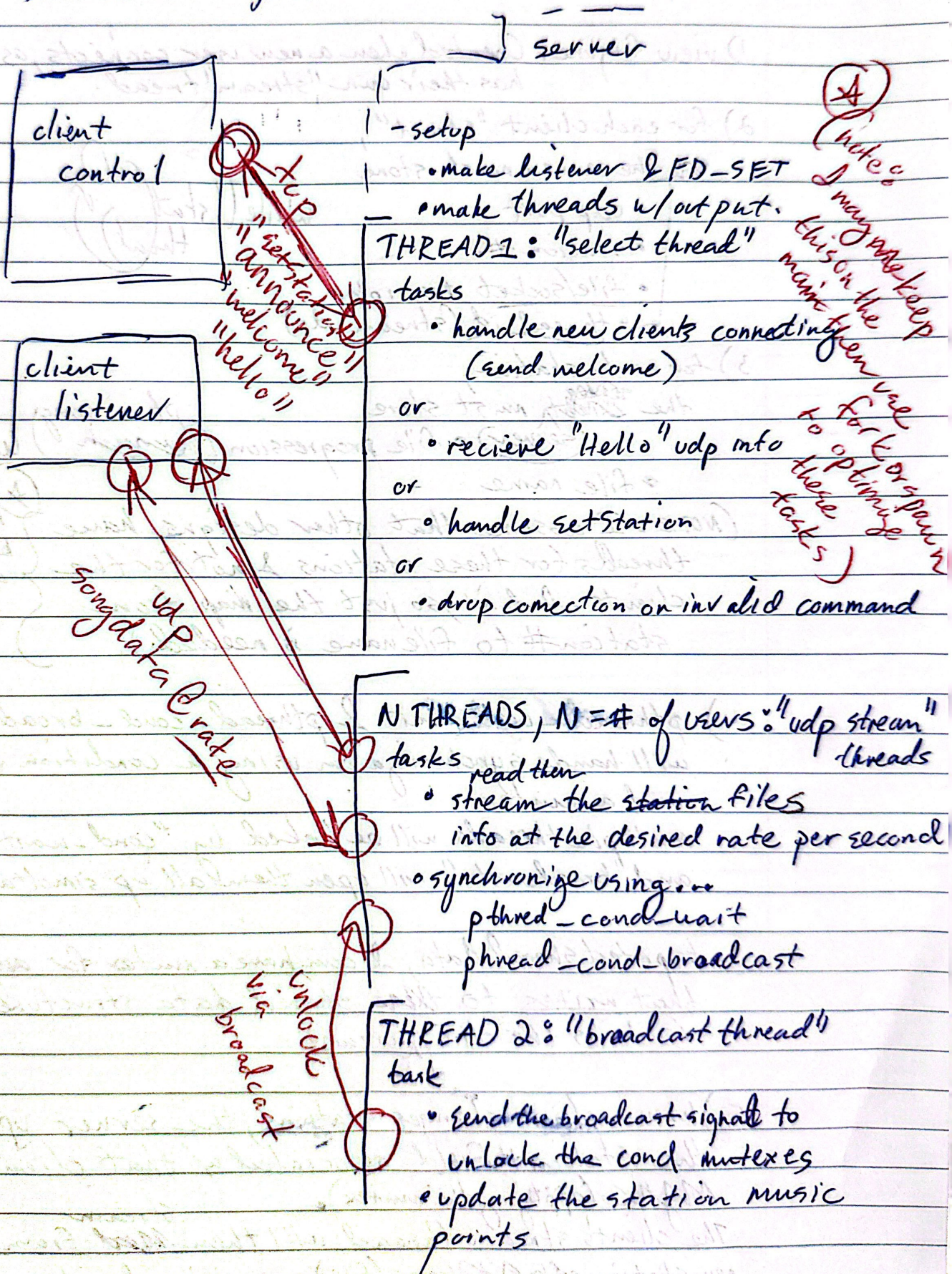


Snowcast design doc



Design Questions

1) view figure. Created when a new user connects, as each user has their own "stream" thread.

2) for each client "struct",
the server must store

- udp port
- station #
- file/socket descriptor
- thread id (stream udp)

while (!stat
thread)

3) for each station

the ~~server~~ ^{server} must store

- file progression (how much pt in song) WHY?
- file name

(NOTE: I assume that other designs have threads for these stations & not for the clients. I don't, so just the map from station# to filename is needed).

My way has better synchronizations

4) pthread_cond_wait & pthread_cond_broadcast will handle synchronization using a condition variable and a mutex.

multiple threads will be locked by "cond_wait" and "broadcast" will open them all up simultaneously.

to protect shared data, I can have a mutex for any operation that writes to these shared data structure (such as the client array)
data

5) when a client changes stations, the server updates the station# field associated w/ that client's ~~data~~ (applying the mutex).

The client's stream thread will then ~~read~~ ^{stream} from the new station ~~after it sleeps~~ (file) on next broadcast.