Concurrency Changes from Iteration 2 to Iteration 3

During the process of modifying the elevator simulator system, changes were made to ensure that the system can run in a concurrent fashion. This meant that the elevator subsystem was expanded to enable multiple elevator threads running at the same time. The scheduler was reworked to accommodate this. This required refactoring code which originally functioned in a 1-1 fashion to a many to 1 design. One scheduler would communicate with multiple elevators. Each elevator thread runs concurrently and would communicate with the synchronized components in the scheduler. This was achieved using an RPC model with UDP datagram packets. By using the UDP datagram packets, the access to shared resources was synchronized between the various threads. This allowed for the scheduler to gather information on the different elevators and determine if the request received can be serviced by the available elevators. The scheduler then creates a datagram packet to communicate with the available elevator. This communication between the scheduler, floor and elevator threads allows the scheduler to get a request from the floor program and then process it to be serviced by a given elevator. Once the scheduler gets a packet from the floor or elevator it will send a reply to acknowledge that it has accurately received it. Due to the addition of the UDP implementation of RPC, the system is able to synchronize the different process handled by the scheduler allowing the implementation to move away from a sequential set of events that the elevator would handle. The scheduler can effectively determine the current positions of changing threads to figure out which elevator is best fit. This results in the system having the closest elevator moving in each direction to service the floor. This differs to the implementation in iteration 2 in that the older implementation had a single elevator sequentially service each elevator without worrying about the time of request and the direction of movement.