

## **Performance Report - Group U**

For testing performance we wrote a few tests that send messages to the server and measure their latency between one another and the average across them all. Our system uses UDP packets to send messages from the client to the server and vice versa. Because of this, the latency for sending messages out from our client is relatively zero, unless restricted by the network interface itself. Our tests do not plan to test the limits of the networking interface, especially in a local environment where the network is a simple loopback.

As for sending updates from the server, our server receives messages off the queue, and updates the game state in the double buffer. Every 100ms the broadcaster will pick up the game state from the double buffer and broadcast this to the client. This means that at the very least, a user cannot guarantee they have received the new game state until 100ms after they have sent their move command (or other command).

This system is hard to measure performance of because we cannot match a single request to a single response and the server continually sends messages, as the design of the project was required. Due to the inherent nature of the broadcaster design and lack of request/response, we cannot measure the latency between a request and the time it takes for the server to process that message under any size of load (low, normal or high). This is to say that under the current broadcaster design of the system, the broadcaster feels no weight of the load due to high message volume, and will continue to broadcast messages on an interval we have specified.

The best that we can do in this case is spam the server with a lot of messages to mimic a high load, and sleep for a specified amount of time before checking the game state. If the game state is what we expected, then we can be reasonably sure that all the messages have been processed and the server is no longer under any high load. This is what our performance tests accomplish.

We have measured our overall performance of a certain scenario to have accomplished a throughput rate of 10000 messages per second. That is, sending 10,000 messages to the server, waiting 1 second for the new game state, and receiving the game state we expected, being reasonably sure that all messages have been processed. We did not push this limit further but we could have. We believe a 10000 message per second peak performance rate is reasonable and sufficient for the requirements.

The max, min and average values of message "latency" is printed out with the performance tests after they complete, however like discussed above, these numbers cannot measure the latency of the request/processing/response, so there is no merit on these numbers. They are simply measuring the time it takes to send the request, which may have no basis of analysis.

Using a protocol like TCP or mimicing a request/response messaging protocol on top of UDP would allow us to measure individual message latency & performance, but we believe that this is beyond the scope and in fact goes against the requirements of the project.