***Computer Networks and Distributed Systems Assessed Coursework: RMI and UDP (by Tanay Verma & Calvin Chan)***

Our project investigated client-server pairs that are used to send messages over the internet. The two types we looked at were implemented using Java Remote Method Invocation (RMI) and User Datagram Protocol (UDP). By sending varying numbers of messages, we will analyse the performance, efficiency and ease of set-up of each type of system.

***UDP***

There are three main ways that messages can be lost; whilst they are being sent, whilst they are in transit or whilst they are being received.

When packets are sent via UDP, they are first placed in a ‘socket send buffer’, which is part of the networking hardware. If this becomes full, or cannot output packets as quickly as they come in, there will a loss of packets. However, in our case this will probably not be the main cause of data loss, as the lab computers’ hardware usually deals with data transmission much more demanding than this exercise.

Most of our packets are probably being lost in transit, due to congestion on the network. This can be caused by an overloaded router, which is receiving data faster than it can send it. At the time we ran our tests the lab was quite busy, which means the load on the internet was higher than usual. This most likely resulted in most of our losses.

Similar to whilst they are being sent, whilst they are received at the server, the data packets go into a ‘socket receive buffer’, which can get overloaded. Again, this is probably not a significant factor.

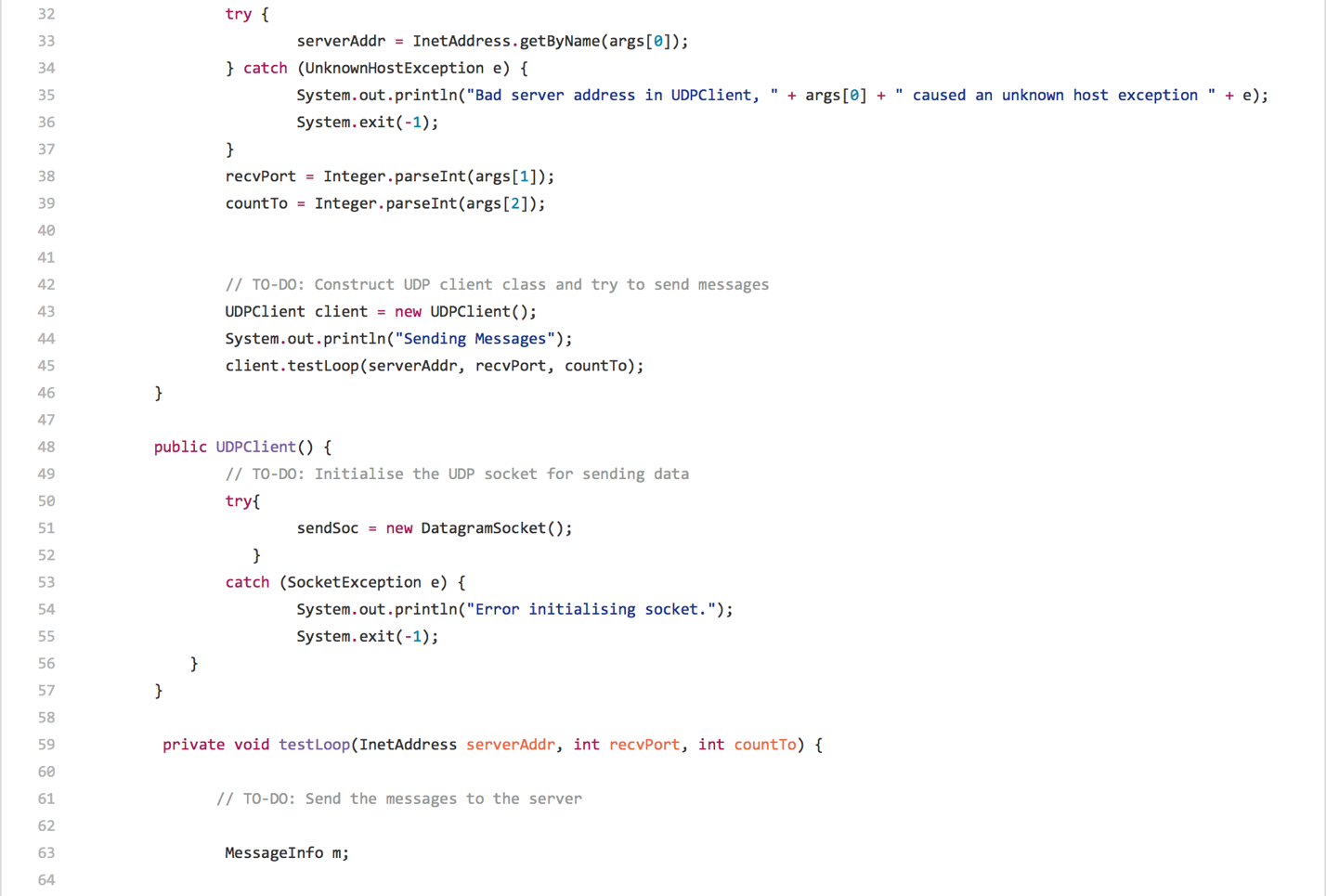
The graph below shows the average failure rate (failed packets/total packets sent \* 100%) for each number of messages sent. It seems that as the number of messages increases, the failure rate decreases, until about 500 messages, after which it stabilises.

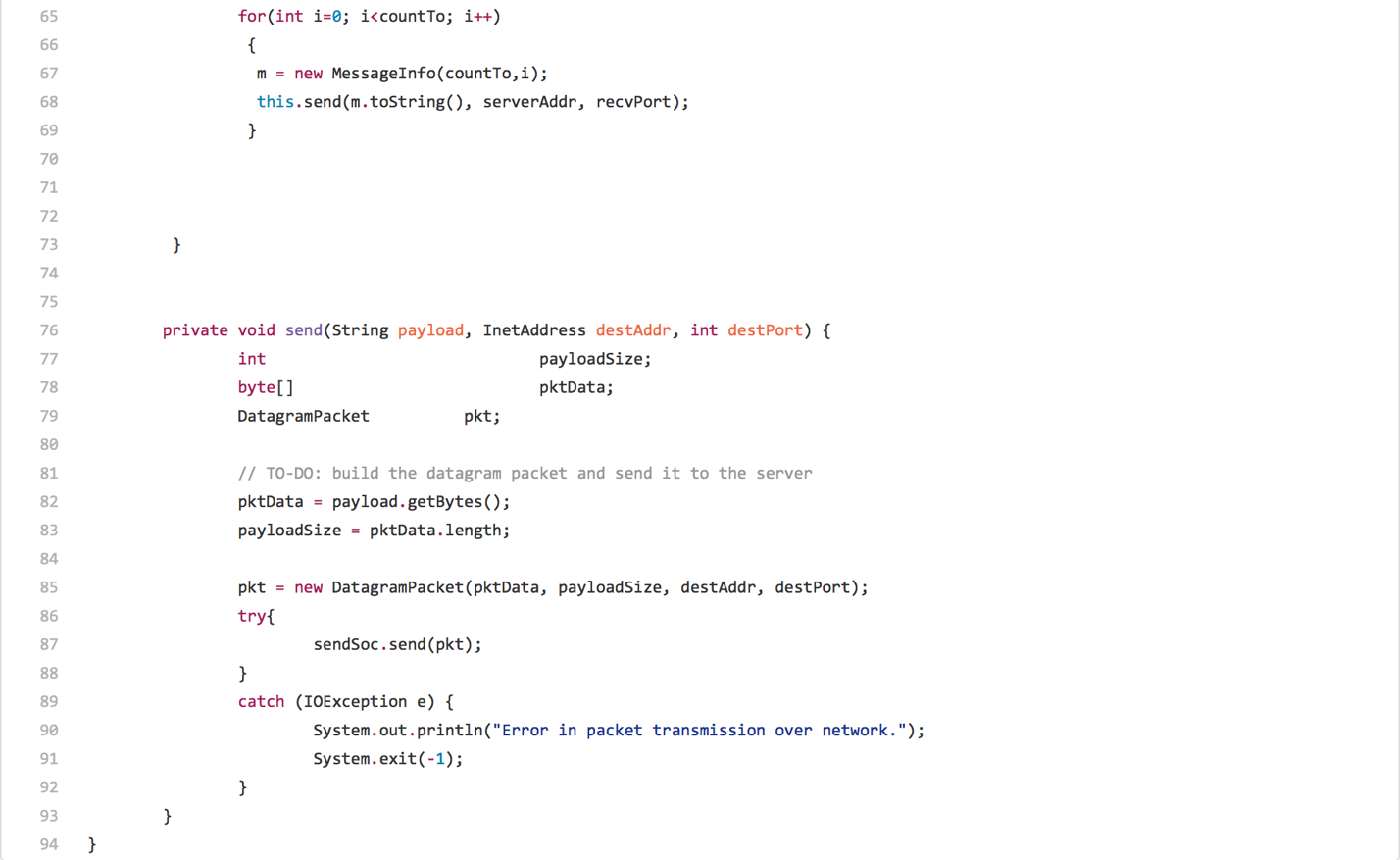


***UDP Screenshots Whilst Running***



***UDP Client Code***



******

***UDP Server Code***

