**RMI (Remote Method Invocation) vs UDP (User Datagram Protocol)**

A comparison between RMI and UDP mainly lies on the transport protocol used in each case. RMI uses TCP, ensuring a connection is established between the client and server and closes when all the messages have been sent. UDP on the other hand just sends the packets over the network without asking for acknowledgement, this means packets are often lost and arrive in a different order. From running the code, it was very interesting to see the theory proven correct in practice and some interesting network behaviour in the DoC labs.

Causes of messages being lost:

TCP (RMI) – Can be lost due to the network congestion, the buffers become full, meaning packets cannot enter the buffer and are lost. TCP does however have ‘TCP congestion control’ to mitigate this.

UDP – Also suffers greatly from network congestion, even more so as there is no congestion control unless built in and the messages are sent more rapidly so buffers get filled quicker.

All our tests were conducted in the DoC Computing labs. There were some interesting patterns for UDP but not for RMI as expected (all received on RMI).

For UDP (Figure 1), we tested on physical distance between lab computers (close vs far) and number of people in lab (empty vs busy). For up to around 300 packets sent, there were none lost. Above this, on BLUE eg, all packet numbers above 305 were lost, maybe giving some indication to buffer size or congestion control of the network by the admin. In the other cases the error rate tends to go up but is variable and there seems no correlation with distance or perceived activity (busy) in the lab. From this we can say there is a drop off point around packet number 300, and above this as the buffer capacity varies, the error rate changes but generally increases.

The reliability of RMI is 100% in our tests, i.e when the network, client and server have not crashed. This is as thought when looking at the theory as mentioned in the first paragraph. UDP however is unreliable, but that’s the way it is designed, to send packets immediately, quickly and without any care whether they were received or not and in what order. When measuring the time between the first and last message received up to packet number 400 (Figure 2), once can see that UDP is much quicker as there is no acknowledgements needed. The distance between machines does play a part on receive time when tested with RMI (blue vs orange plots).

In terms of programming, the easier one to program was RMI, being more intuitive, as it has a higher level of abstraction due to the use of the remote interface for the transmission of the message. With RMI we do not need to deal with ports or sockets anymore. Another reason why RMI is easier to program is that the code for UDP is larger so there are higher chances of making mistakes.

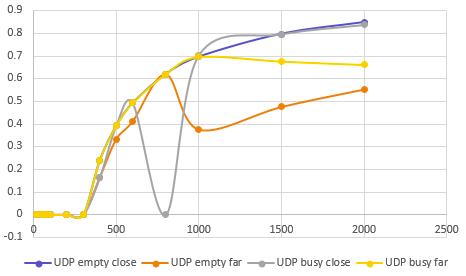
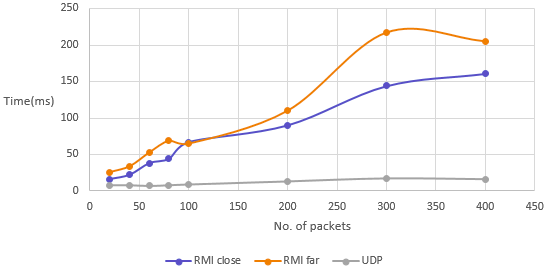


Figure 2 *:*

*Time between first and last message received by server(Y-axis) vs No. of transmitted packets(X-axis)*

Figure 1 *:*

*Error rate(Y-axis) vs No. of transmitted packets(X-axis)*