Imperial College London
Department of Computing

February 18, 2015

Computer Networks and Distributed Systems

Assessed Coursework - RMI and UDP

name Maciej Olejnik
username mo1712

1 Summary

1.1 Lost Messages

Message loss is a bigger issue in case of using UDP as it is a protocol with no guarantee of delivery, ordering, or duplicate protection. The most obvious reason for message loss might be failure of the network - the bigger the distance between client and server, the bigger the probability of dropping a message. Network failures include network congestion, faulty hardware or drivers. Another reason for the message loss is buffer overflow at the receiver's end. The server machine has a buffer of certain size which is used to store incoming messages. If the messages are coming at a faster pace than the server is processing them, the buffer will get full and some messages will have to be dropped. In case of RMI, the abstraction should take care of network failures and buffer overflows by adjusting the pace at which it send messages or retransmiting them if necessary. So the only way RMI can fail to deliver the message is if the server is down or there is no physical connection between the client and the server (eg ethernet cable being out-of-order).

1.2 Observed patterns

Observations indicate that the buffer overflow is a serious issue when using UDP. When more than 310 messages are sent by a client, lost messages emerge. Initial roughly 300 messages are almost always successfully delivered but it seems that after that the buffer becomes full and drops any subsequent messages. If enough messages are sent after the buffer becomes full though, some messages are successfully delivered. The reason for that is that the server managed to process a message and it emptied some space in the buffer, so another message could be stored. The pattern is the following: 300 messages delivered successfully, and then every k^{th} message delivered, where k is between 10 and 40 in most cases. There was no pattern observed in case of RMI, because of extremely high reliability of this protocol.

1.3 Relative reliability of RMI and UDP

In short, RMI is way more reliable than UDP. The reason for that is that RMI is a high-level abstraction and takes cares of any failures that might occur during message transmission (see section 1.1) and takes appropriate actions to deal with those failures.

1.4 Easiness of programming

RMI was way easier to program for me. Firstly, there was less code to write. Secondly, it was more intuitive as it involved creating an object (object reference actually) and calling a method on it, which is typical way to program in Java, rather than sending an array of bytes to some other computer. Finally, RMI abstracts away the details of the communication between two machines over the network, takes care of marshalling/unmarshalling the arguments for you and hides the details like port number and sockets.