

Module 3 -> Challenge 2

Authors:

2356546 - Fátima González

2198312 - Wenjie Zhao

Report:

For the second challenge of the Network Systems course, we designed and implemented a reliable transfer of data over a link which may arbitrarily delay or drop packets. The aim of this report is to discuss our approach and the information we learnt while performing the challenge. We worked closely with the website "networkingchallenges.ewi.utwente.nl", where our results were displayed and an analysis of them could be done easier and better.

Firstly, we started by changing our group token in the code. At the beginning we were getting errors, as the pictures that needed to be sent were not in the correct folder. As soon as we noticed and fixed the error, we got our first result for file number 1, a score of 574,1. The code was now working, so we started improving it according to requirements. Dividing the packet into different pieces was our first proposal. We managed to do it through making filePointer a variable, each time it will increase by adding the data length which has been sent. Also, we wrote a piece of code to figure out how many packets should be sent (totalACK) as while loop condition, and the variable fileLength is used as the while loop condition in receiver, which means when the receiver receives a whole file the while loop will stop receiving more packets. In file 2, we got a score of 1195,5. However, the transferring of file 2 was not stable, and we realised that the acknowledgments did not work properly.

We only managed to do it until the second file, because of a problem with the acknowledgments. In order to make the program work faster, instead of using the stop and wait function we could have used the sliding window algorithm. Also, our timeout took very long, and maybe it was not the correct way to do it.

To conclude, in this challenge we not only learned precious knowledge about data transferring but also a practical way to implement it through java. We tried multiple methods to improve our code in order to deliver all data as quickly and stable as possible. In the end, we enjoyed learning more about reliable data transfer, and got inspired by how to improve our code to make it work better.