

Report Challenge 3: Made by 2198312 - Wenjie Zhao and 2356546 - Fátima González

For the third challenge of the Network Systems course, we designed and implemented a medium access control protocol to share a time-slotted medium fairly and efficiently among four nodes. The aim of this report is to discuss our approach and understanding of medium-access mechanisms. We worked closely with the website "networkingchallenges.ewi.utwente.nl", where our results were displayed and an analysis of them could be done more facile and preponderant.

Firstly, we started by changing our group token in the code. At the commencement, we were getting a score of 26.29 in efficiency and 92.4 in fairness. We realised that efficiency was the first thing we need to ameliorate. We decided to improve the basic protocol to increase efficiency. After analysing the simulation trace, we found that following successful data sending, there is more likely to be another successful sending, and collisions are also more likely to be gathered together. That is the reason why we decided to change the probabilities according to the value of the previous medium state. After increasing the efficiency score to 37. We realized we should also make use of the other two variables "controlInformation" and "localQueueLength" to make the protocol more complex.

Firstly we created a switch condition with different cases. Idle state(node stops sending data), Test state(transition state between the other two states), Active state(node continually send data), in Idle state, each node has the probability of 60% to turned into Test state and send data, and when no collision happens, the node will turn into Active state and continue sending data until the count of occupying slots reaching the limitation. The limitation is to stop sending data and give room for other nodes. At first, we set the limitation as 20. With this implementation, we reached a genuinely high efficiency of 92.8 but our fairness went down to 23.6. A balance between fairness and efficiency was needed to improve our results. That's why we decided to modify our code again looking for an equilibrium between both and it's here where we achieved our highest score of 67.8 efficiencies and 68.9 fairness. In order to improve our current solution, instead of implementing Slotted Aloha as we did, we could have used another method that would be more efficient. We did not know at first that this was a possibility therefore, we implemented this method. If we were able to go back in time, we would have started using polling or CSMA/CD, the reason is that if we can distinguish each node, for example, give each node a unique ID, and assign different time slots fairly to the four nodes, we would be able to definitely boost our fairness score. Currently, we were only able to play with probability to find a higher score as we chose ALOHA method.

To conclude, in this challenge we not only learned precious knowledge about medium access control protocol but withal a practical way to implement it through java. We tried multiple methods to amend our code in order to increase both our efficiency and fairness. In the end, we enjoyed learning more about medium access control protocol and got inspired by how to improve our code to make it work better.