Challenge 3 – Medium Access Control

There are two extremes when realising a medium access control, both of which optimize for either efficiency or fairness. To optimize for efficiency, once a node managed to successfully send one packet, it keeps sending until its local packet queue is empty. Other nodes must wait until the medium is idle again, resulting in high latencies for them.

The other extreme would be that a node never uses two adjacent time slots, giving every other node a chance to be the next one to send. Depending on the specific collision management used, this approach distributes the available time among the nodes about evenly. However, collisions may occur after every packet, significantly reducing overall efficiency.

For our solution, we first implemented a generalization of both approaches. Once a node successfully transmits a data packet, it receives a token. While the node keeps the token, all other nodes remain silent. This is the case until the node transmitted a certain percentage of all the packets currently in its local queue. For percentages of 100% or 0%, this would result in the extremes described above. Every other percentage would realize a compromise, which is what we first went for. However, after experimenting a bit, a randomized approach gave better results. Now, the probability of keeping the token after a transmission starts at 100% and is reduced by 5% for every packet sent since the token was first acquired. If there are no more packets in the local queue, the token is never kept.

After this calculation, the node informs the other nodes in the system of whether it intends to keep the token after successfully sending a packet by transmitting control data of 1 (keeping the token) or 0 (giving up the token). This also eliminates the need for one time slot of silence between two nodes transmitting, because the control data signals whether the medium will be clear in the next slot. After a node indicated to give up its token, that node may not immediately try to send data again. It must wait until another node gave up its token or at least one time slot of silence occurred.

In the case of a collision (which can only occur between nodes, because other nodes don’t interrupt the node with the token), all nodes which wish to send data have a certain probability of trying to send in the next time slot. If another collision happens or all nodes remain silent, this is repeated until one node successfully transmits and receives the token.

Jonas Becker (428292) & Daniel Beckmann (416096)