Homework 4

Riccardo Pezzoni 10575577

May 16, 2022

1 Simulation

In *RunSimulationScript.py* a couple of settings are necessary. First of all some debug channels are specified. The ones I made use of are: timer, boot, radio, radio_send, radio_ack, radio_rec.

Then the two created motes are set to boot at 0 s and 75 s respectively.

2 Message

The message object *my_msg_t* is defined to contain three fields:

- *msg_type*: containing value 1 for requests and 2 for responses
- *msg_counter*: increasing counter of the sent messages
- *value*: sensor data, only used in responses.

3 Configuration

In *Challenge4AppC.nc* components and interfaces are specified. The defined components are MainC and Challenge4C as App, AMSenderC(AM_MY_MSG), AMReceiverC(AM_MY_MSG), TimerMilliC(), TimerMilliC(), TimerMilliC() with the following connections.

```
App.Boot -> MainC.Boot;
App.Receive -> AMReceiverC;
App.AMSend -> AMSenderC;
App.SplitControl -> ActiveMessageC;
App.MilliTimer -> TimerMilliC;
App.Packet -> AMSenderC;
App.Read -> FakeSensorC.Read;
```

4 Program Logic

The logic of the program in entirely contained in *Challenge4C.nc*.

5 Results

Looking at the printed log in Challenge4.log we can see that node 1 boots immediately and starts to send Requests without receiving any ACK. At 75s node two boots. Due to the timer imprecision and the time it takes to boot the first requests that receives an ACK is the one with counter 77. From that moment on every request made by node 1 is acknowledge and so is every response (which contains the sensor data and the counter which it refers) made by node 2. Upon the receipt of the 8th ACK, node 1 stops the timer and after the last response, the simulation finishes.