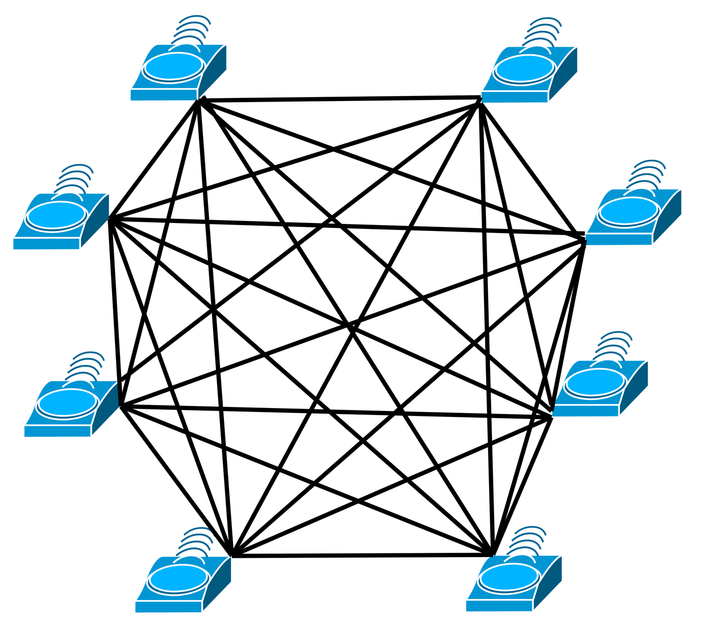
Implementation of Cooperative Information Storage on Distributed Sensor Boards

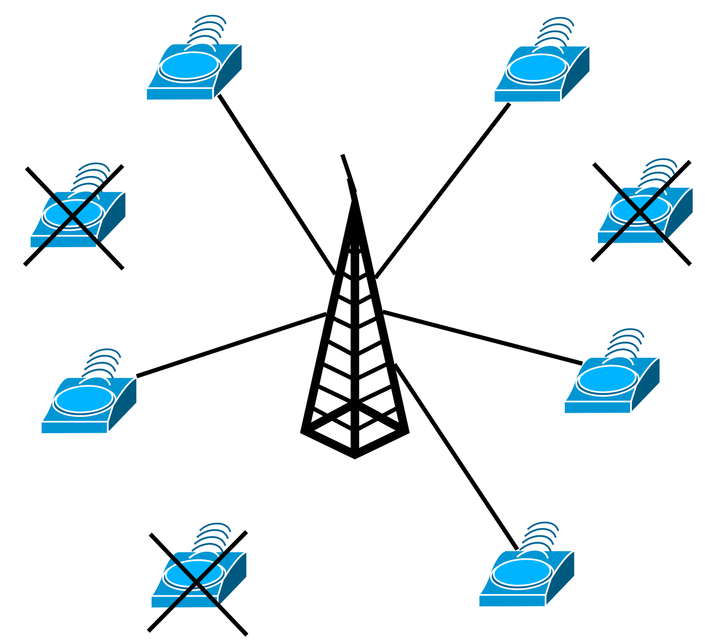
Anders Grauballe, Mikkel Gade Jensen, Achuthan Paramanathan, Janne Dahl Rasmussen, Tatiana K. Madsen and Frank Fitzek. Aalborg University, Department of Electronic Systems, Niels Jernes vej 12 A, 9220 Aalborg East, Denmark
mobiledevices.kom.aau.dk

Introduction

The basic scenario in this project is a cluster of sensors which are partially wirelessly connected with a gateway. A sensor is a small device (often called mote) which can measure environmental conditions. It is also able to communicate and relay messages to a control unit or gateway (see figures). Each sensor and the gateway is partially connected because they may not be active at the same time, they are e.g. out of range or powered off. The gateway will in this scenario not always be able to receive each measurement from the sensors.
Therefore the objective of this project is to develop and implement a distribution model for the OpenSensor boards[1] developed by Aalborg University, such that data from each sensors can be received by a gateway, even if not all sensors are online / in range.



All sensors are in range and partially connected.



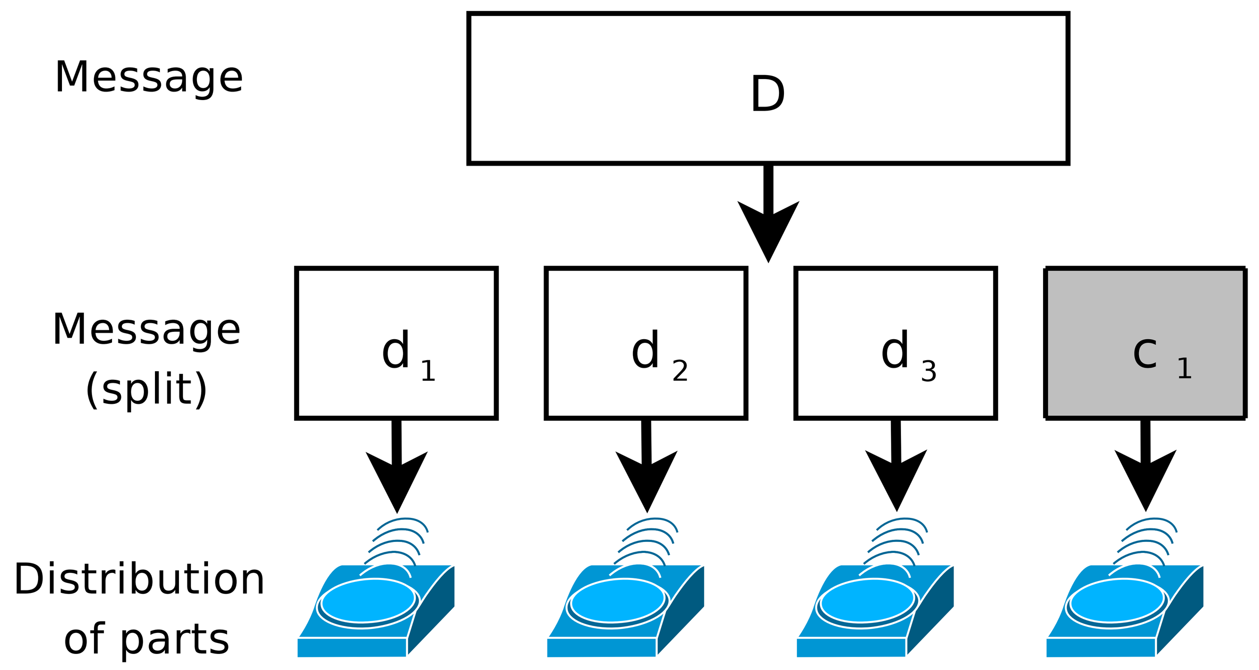
Not all sensors are available when the gateway is present

Distribution and Reconstruction

Each mote collects the data individually. When the data is collected, it is encoded and distributed to each of the motes in the cluster. The distribution of data from a sensor to other motes in the cluster is done in the following way applying Reed-Solomon (RS) coding in the system:

1. Data is split into $l = n - m$ parts.
2. m redundant parts are generated using RS on the data parts resulting in a total of n parts.
3. The parts are distributed to n motes (including itself), one part for each mote.

The steps in the algorithm are illustrated below



The data (message) is split and distributed. d is data parts and c is checksum

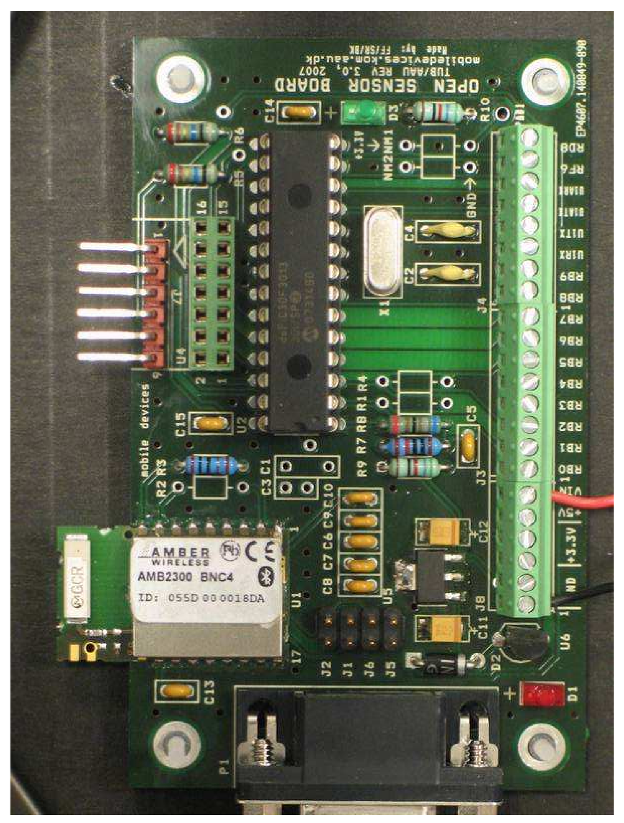
When the gateway arrives to collect data, it is done in the following way:

1. The gateway broadcast a signal to announce its presence
2. Online motes stay online throughout the gateway session
3. Gateway requests data from each mote. If a timeout occurs on request, the mote is considered offline
4. With the gateway request each mote receives a new session number
5. Gateway receives data from each online mote
6. When the gateway has received data from all online motes, it leaves the cluster
7. The online motes will then measure and distribute new data
8. When offline motes become online, they receive the new session number with new data and deletes old data. Then they will measure and distribute new data

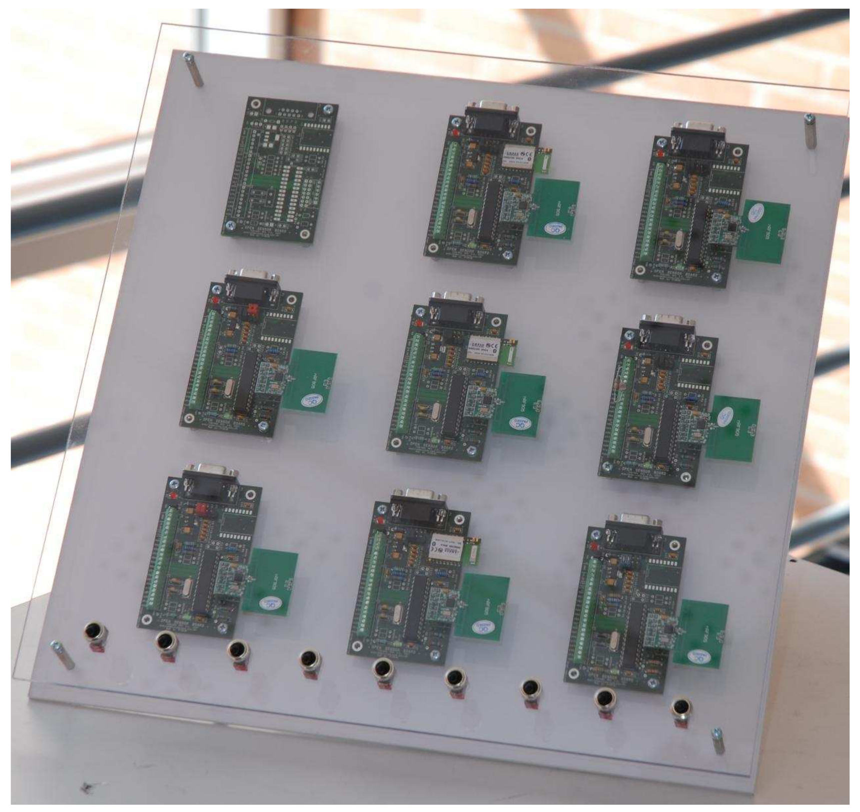
Sensor Platform

Each OpenSensor v3.0 mote contains a microprocessor, communication module, sensor module and power supply. All of this is contained in a box typically about the same size as a small mobile phone. This mote is the one being used for the implementation and it has the features described in the following. [1]

- Li-Poly battery interface which can be used for power supply to the mote
- RS-232 serial port interface
- dsPIC30F3013 microprocessor for controlling the mote
- 22.1 MHz oscillator as external clock source for the dsPIC
- ICD 2 debugger/programmer interface
- Programmable LED
- Bluetooth module for wireless communication with mobile phones or PCs, connected with serial RS-232 connection to the microprocessor
- nRF905 (Nordic Semiconductor) transceiver for communicating via the ISM band
- Removable loop antenna (433 MHz) for the RF transceiver



The OpenSensor v3.0 platform



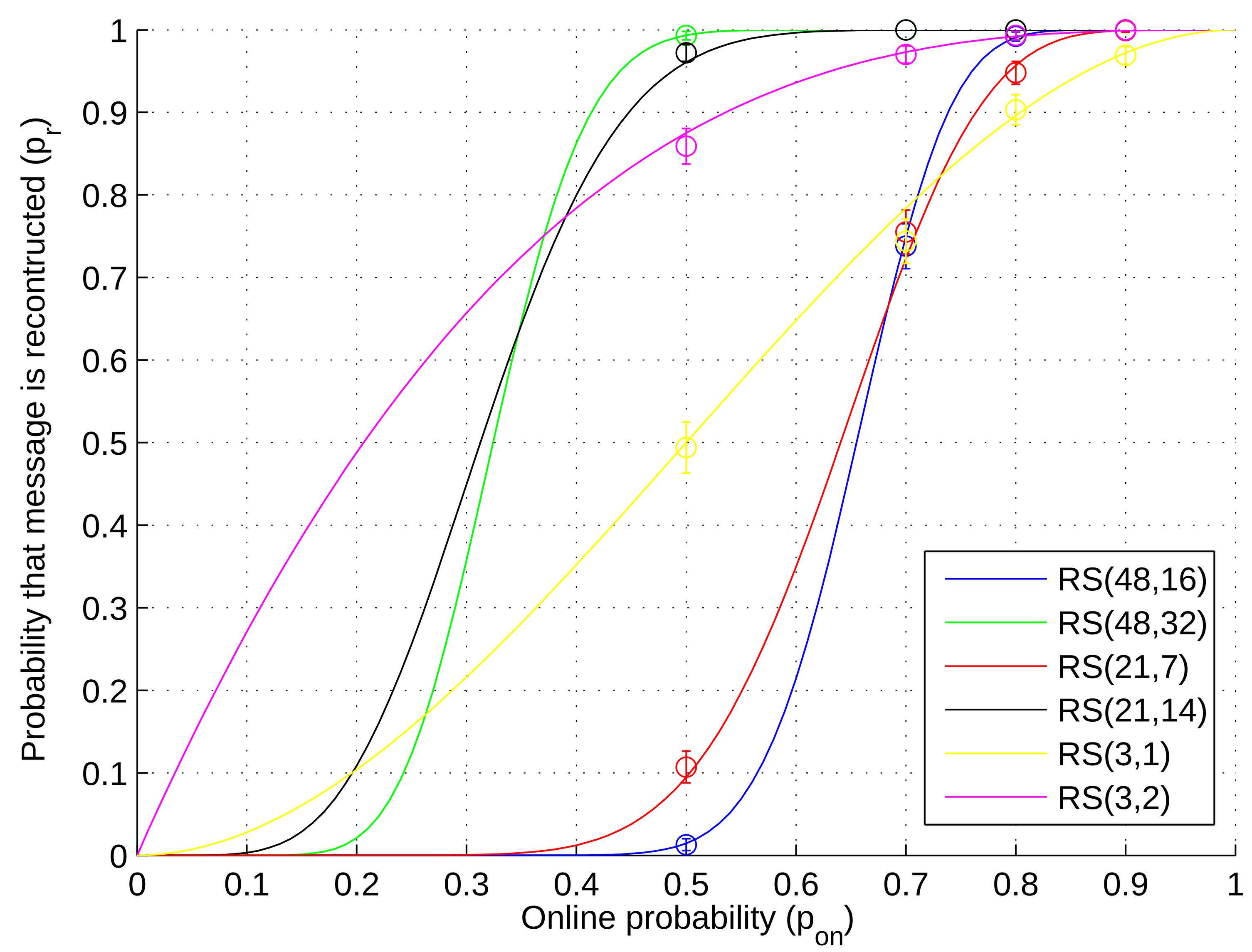
Sensor demonstrator

Results

Due to the limitation in the number of motes in the prototype, it was not possible to test the generel performance of the proposed protocol in a real life scenario.
In order to understand how the choice of the coding scheme influences the data retrieval reliability, we have investigated the system performance by means of a simulation and an analytical approach. We have implemented a simulation of the system in Java both to investigate the performance of the system with different number of devices and to illustrate the different states that each mote would be in at a given time. The parameters that can be varied are:

- Total number of motes
- Number of checksum motes
- Online probability

To ensure that the motes are independent regarding state shift, each mote runs in a separate thread. Furthermore, an ideal MAC protocol is assumed, i.e. each sent packet is received without any errors. Each of the following cases: RS(3,1), RS(3,2), RS(21,7), RS(21,14), RS(48,16), RS(48,32) has been tested with online probability $= \{0.5, 0.7, 0.8, 0.9\}$, RS(n, m) being a total of n motes with m of them being checksum. The results can be seen in the graph below where the lines are an analytical model and the circles are simulation results.



Probability of reconstruction

As seen the simulation results verifies the analytical model in the six cases and it is assumed that this will also be the test results of a full scale implementation.

References

[1] Ben Krøyer and Frank Fitzek. *Mobile Devices - OpenSensor.* http://mobiledevices.kom.aau.dk/opensensor/, 2007