Scientific report – *Selective Code-dissemination for Wireless Sensor Networks*

The present document describes the result of my two-weeks long visit to the University of Trento, under the supervision of Gian-Pietro Picco. My collaborative partner is Luca Mottola from the Univeristy of Milan (while working in Trento), and our proposed system is a selective code-dissemination protocol for mobile wireless sensor networks (WSN). These networks are usually applied to wildlife monitoring, where the sensor devices are attached to animals in some remote areas. It is then very difficult both to recharge the batteries of the sensors, and to connect the sensors to computers to install new programs, therefore there is a need for an energy-efficient code dissemination protocol. There is also a lack of support for sharing an already deployed sensor network, while resources might permit this. The proposed protocol is designed to allow dynamic code-upload on a specific subset of a sensor network. The document is structured as follows: first, a description of the initial protocol is given as before the visit, then the motivation, and in the last part, the main achievements of the visit are described.

Most of the existing work on code dissemination for WSN is based on flooding the network with the code-update. Although this achieves a fast delivery to the nodes, it also encounters a lot of overhead in terms of unnecessary message transmissions. Since WSN are very energy-constrained systems, there is a need for a smarter code-dissemination protocol. The protocol creates a logical network on top of the physical topology based on the social interactions of the nodes, i.e. there is a link between two nodes, if they have a social link between them (they meet often enough). The protocol can recognize social groups, thus can update groups using broadcast messages instead of updating the group-members one by one.

After some initial evaluation of the base-protocol, we wanted to extend it to allow for selective code update. The idea is to let the user decide the characteristics of the nodes to be updated, and the underlying protocol would find an energy-efficient route to the target nodes. An example is to update only the nodes reading temperature over 10 ^{o}C. To achieve this, we need to provide programming primitives to define the constraints and allow for dynamic code-update. Sensor reprogramming has been an intense area of interest of Luca Mottola, while I am focusing on the networking side, so we looked for ways to collaborate. After some initial discussions and planning, I paid a visit to the University of Trento to discuss and describe the details of the system and start the implementation of the networking layer needed.

The outcomes of the visit are as follows (this is a high-level view!):

- definition of the programming primitives to identify the target set of nodes: these include primitives to define node attributes such as temperature, humidity, and constraints to target a specific subset of the nodes to be updated. A simple constraint would be *constraint(avg_temperature > 10)* to target nodes sensing a temperature greater than 10 ^{o}C

- definition and implementation of the underlying protocol to implement these primitives: to efficiently deliver the code update to the target nodes using only the necessary forwarders, we use the logical network of the base protocol. An aggregated value of each group is spread around of the network, and used by the nodes to decide whether they need to take an update or not
- a plan for the implementation of the programming model, and its combination with the routing layer, and finally how the system is going to be evaluated

The visit clarified many of the details of both protocols and sparked new ideas, which would not have otherwise been possible to come up with (such as the possible types of attributes and constraints supported by the system). The visit was also fundamental in planning the next steps to complete the system. These are happening now, as we are preparing for Luca's visit to Cambridge in early August. His visit will allow us to finish the implementation of the programming model, and combine the two systems. The program code will be made publicly available after the visit. The system is then to be evaluated using realistic simulations, and the results are to be published at a relevant conference such as the ACM International Middleware Conference (Middleware09) or ACM Conference on Embedded Networked Sensor Systems (Sensys09).