Open Sensor Networks
P2P communication and Cosm

Alejandro Andreu Isábal
alejandro.andreu01@estudiant.upf.edu

I. Final arquitecture design

Using Arduino and ZigBee is very convenient for the sake of this project since they are two very mature and easy-to-use solutions. However, although Arduino allows very fast prototyping it lacks computational power as well as Internet capabilities. This is why an external component is necessary, a server.

A server with a *XBee* module attached to it is powerful enough to allow an entire radius around it to upload environmental data. Independent nodes distributed in the streets or other kinds of public places can be deployed easily and it reduces as well the final cost for many users if they are willing to share their data –only a node is necessary–. These days small servers are no longer expensive and can be bought for less than $100.

The *Raspberry Pi* is the perfect choice since it is an *open-source hardware* device capable of running many *ARM-based Linux distributions* which costs $25. Hence the design can be summarized graphically as follows:

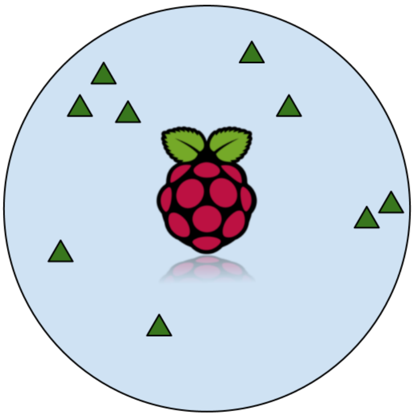

Fig. 1. Arquitecture diagram. Each triangle represents a different node.

Depending on which ZigBee protocol and version we are using the circle will vary its radius. For instance, with the *PRO* version *Series 1* we can theoretically reach any ZigBee node 7km far away from the coordinator. This means a city like Barcelona with an urban area of $803km^2$ can be covered with less than 10 micro-servers.

II. Data uploading

The first open data portal that must be compatible with this network is *Cosm* (formerly Pachube), since it provides excellent documentation and API. Also, it very well-known worldwide and their ease of use cannot be compared with any other website of this kind.

In this platform, every user can have many *feeds* –or sensor nodes ideally–, each one with its own *datastreams*. That is, every variable that is being measured at a given time. Cosm then elaborates complex graphs representing the information and makes it available through their API. At the same tame, every user can *tag* the information so it's easier to find when searching for specific feeds. This could lead to a `osn` tag for the *open sensork network* pilot to better identify its participants.

III. P2P communication

Achieving P2P (or *point-to-point*) communication between two ZigBee compliant devices is very easy. The only requirement is to configure them to work in the same network identifier, and also there shall be one *coordinator* and one *router* –these are examples of ZigBee roles, each one with its own unique functionality–. Whenever the router "says" something, the coordinator receives it inmediately. In this case, the Raspberry Pi (*RPi* from now on) is the coordinator and the Arduino behaves as the router.

The *XBee Explorer* is a breakout board that has an USB interface and that works with any XBee module. Thus it can be connected to the server waiting for nodes to inform about their sensor status. Any Linux distribition will recognize this board as a regular device.

IV. Current status

At the time of writing, communication and data uploading work smoothly. This is achieved by several programs, which are explained below:

- Arduino program: It gathers any information available and prints it through its serial interface. That is, the XBee module.
- Python script: Thanks to the `pyserial` library, any Python script can read and print information from a serial port. It reads different lines and stores them in temporary variables. After that, it serializes the data into a *JSON* file and writes it to local storage. Finally, a *bash* script is called.
- Bash script: Cosm provides a very easy way to upload data in **NIX* systems, that is, through a program called *cURL*. It takes as an argument the previous JSON file and uploads it to their servers.