Wireless Sensor Network with Perpetual Motes for Terrestrial Snail Activity Monitoring

D. García-Lesta, D. Cabello, E. Ferro, P. López, and V.M. Brea

Wireless Sensor Network with Perpetual Motes for Terrestrial Snail Activity Monitoring

D. García-Lesta, D. Cabello, *Member, IEEE*, E. Ferro, P. López, *Member, IEEE*, and V.M. Brea

***Abstract*—Wireless Sensor Networks (WSN) are increasingly adopted in agriculture to monitor environmental variables to predict the presence of pests. Differently from these approaches, our work introduces a WSN to detect the presence of snails in the field. The network can be used to both trigger an alarm of early pest presence and to further elaborate statistical models with the addition of environmental data as temperature or humidity to predict snail presence. In this work we also design our own WSN simulator to account for real-life conditions as an uneven spacing of motes in the field or different currents generated by solar cells at the motes. This allows to achieve a more realistic network deployment in the field. Experimental tests are included in this paper, showing that our motes are perpetual in terms of energy consumption.**

***Keywords*—Wireless Sensor Networks, Agricultural pests, capacitive sensors, ZigBee, sensor applications**

I. Introduction

Several works have reported the huge damage that land snails cause to agricultural plantations (see [1, 2, 3, 4], among others). Currently, the best method to fight them is to regularly apply molluscicides to the whole plantation, whether or not pests exist, with the consequent environmental and economical impacts. Taking this into account, an early detection of the snails presence could help in the decision of when and where to apply molluscicides.

In this context, some approaches have been made with statistical models based on meteorological data to forecast the presence of snails. In [5], two different statistical methods on the Achatica Fullica Bowdich are used to both confirm their presence and yield a degree of sensitivity to their presence in South America. The work addressed in [6] provides the same information for smaller areas based on information gathered both from local weather stations and national weather agencies.

Snail sensors can be used either to collect data to iteratively improve statistical local models of snail presence forecast or to trigger alarms of snail early detection. In this regard, to the best of our knowledge, there is not any solution for pest prediction using local snail sensors gathering information of the actual presence of snails. This work aims at developing a Wireless Sensor Network (WSN) based on capacitive sensors to detect the presence of snails in the crop. This approach is conceptually different from the usual pest prediction methods that are based on statistical models and not on the physical sensing of the pest. Also, computer-based image analysis has been used for the pest detection, but with a high computational and power cost [7].

There are several works of WSN in precision agriculture. As an example, the work in [8] describes a WSN deployed in a crop to gather data about temperature and relative humidity, among other variables. These data are sent to the base station, which is communicated with a PC via WiFi. Another example of WSN in precision agriculture is the work reported in [9], where a WSN with GPRS or GSM modules is employed in large plantations to collect temperature and soil moisture data to optimize land irrigation. The data collected throughout the network are sent to a base station, from which they are issued to a web server using a GPRS module. Authors of [10] also address the irrigation task of a vineyard using a WSN of sensors and actuators. Local environmental data are collected and sent to the base station, where they are used together with weather data for the watering decision.

Another recent example of a WSN is described in [11]. Authors show an application in smart farming for feedlot animal health monitoring. Using directional antennas and 802.15.4 transceivers, the authors detected the presence of the animals at key places (as water and feed bunks) and sent this information to a web server through the routers. All the system was optimized to improve the lifetime of the network by simulating the routing algorithms and the wake-up periods of the sensor nodes.

This paper designs and tests a WSN for snail pest detection. The paper adds contributions to our previous work presented in [12]. First, as explained in Section II, the WSN motes are improved to both cut power consumption and minimize misdetections from interferences and erratic moves of snails over the sensor. Second, we design our own WSN simulator to account for more real-life conditions. Third, we measure the snail activity and the energy consumption of such a WSN for several weeks on a test field, checking out if the new type of operation designed for low-power consumption leads to perpetual motes.

II. Snail sensor mote

Terrestrial snails are gastropods which develop their activity mainly at night. Given that the rest of the day they look for a refuge to rest and protect themselves from their predators, providing them with shelters and measuring their occupancy levels is a good strategy to detect their presence in a crop. The shelter we have designed for this work is a PVC pipe allocated

This work has received financial support from the Consellería de Cultura, Educación e Ordenación Universitaria (accreditation 2016-2019, ED431G/08) and the European Regional Development Fund (ERDF) and TEC2015- 66878-C3-3-R.

Authors are with the Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, 15782, Spain. (e-mail: {daniel.garcia.lesta; diego.cabello; esteban.ferro.santiago; p.lopez; victor.brea}@usc.es).

Fig. 1: Mote for snail pest detection based on capacitive sensors (left) and a snail entering the shelter (right).

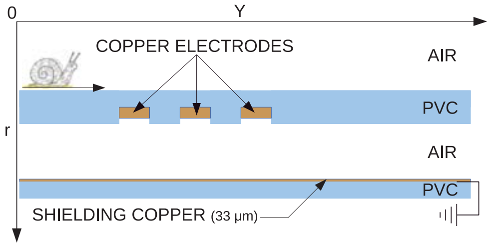

Fig. 2: Cross-section view of the lower part of the PVC pipe used for snail shelters (radial symmetry with respect to Y axis).

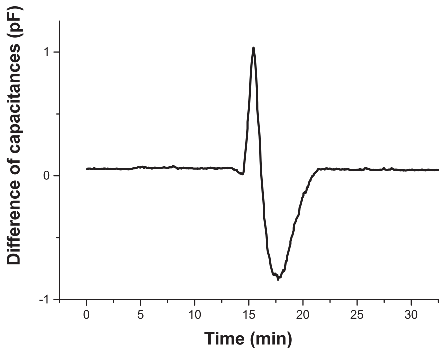

Fig. 3: Measured signal when a snail enters the shelter depicted in Fig. 1.

at tenths of centimeters above the ground with 25 cm height and 5 cm of diameter.

Previous work of the authors in [13] addressed the design of a mote with a photoelectric sensor that triggers detection by the reflection of infrared light from the snails. The system also comprises a solar cell to extend battery life. Although this system worked very well in darkness and cloudy days, natural infrared light saturates the sensor at sunny days.

Capacitive sensors are good candidates to circumvent the infrared radiation issue during daylight taking advantage of the high dielectric constant of the snails ($\varepsilon_r \approx 80$). A capacitive sensor with three copper electrodes around the pipe was designed by authors in [12]. A finite element simulator [14] was used to estimate the width, thickness and space values between copper electrodes to maximize sensitivity to snails within the input capacitive range of a 12-bit capacitive to digital converter (CDC) [15], which acts as conditioning circuitry.

Fig. 2 shows the aforementioned three copper electrodes inserted in three dents at the bottom part of the pipe. The system measures the difference of capacitance between the central and the bottom and top electrodes, respectively. The resultant capacitive sensor mote is described in [12], showing promising results. Fig. 3 is an example of a signal measured from our system with a snail entering the shelter. This signal is easy to process by a microcontroller as it features a maximum followed by a minimum. A snail leaving the shelter yields a signal with a minimum followed by a maximum. Snails standing over the electrodes provide noise, leading to misdetections.

The conditioning system to measure capacitance differences in our WSN motes is the 12-bit CDC AD7153 [15]. This chip features a differential input range up to ± 2 pF with a conversion time from 5 ms to 60 ms at low current (100 μA at 3.3V). The CDC is connected to the electrodes of the capacitive sensor through grounded coaxial cables. Configuration and measurement results are available via an I2C serial bus to Arduino Fio, which acts as microcontroller and feeds the system through a 2000 mAh LiPo battery and a solar cell of 100 mA and 5 V. Communications are carried out by an XBee board. A picture of the system can be seen in Fig. 4. All this circuitry is inside a plastic box located at the top of the shelter, which in this work has been replaced with an Al sealed box (see Fig. 1).

Finally, in terms of construction, it should be noted that capacitive sensors are sensitive to any material with a dielectric constant greater than the air's, so in order to avoid signals not caused by snails, e.g. rain, the pipe is surrounded by another PVC pipe sealed at the top and the bottom ends. Additionally, we have added a 33 μm width copper film shorted to ground at the inner part of the outer pipe (see Fig. 2).

Our former motes in [12] have been improved in this work to minimize both power consumption and experimental errors. Low power consumption is achieved by including radio sleep mode and local data storage and processing. Misdetections caused by snails standing on the copper electrodes are minimized with an IR light added to draw snails out of the copper electrodes. Finally, all the circuitry is enclosed in an Al sealed box shunted to ground, providing a much better effective shielding from EMI, RFI, and water leakages.

Capacitive sensors to detect animal activity were also reported in [16], where authors developed a capacitive structure to count the number of bees entering and leaving a beehive. In this case, the conditioning circuitry was an AC bridge. Our approach is more energy efficient, as it comprises a CDC as conditioning circuitry and a solar cell for environmental energy harvesting. Also, authors in [16] do not describe any WSN.

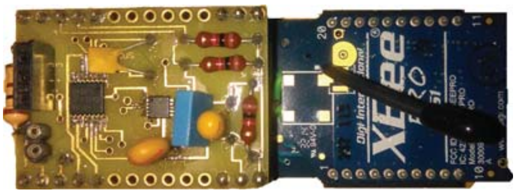

Fig. 4: Conditioning circuitry and XBee plugged into Arduino Fio of our WSN motes.

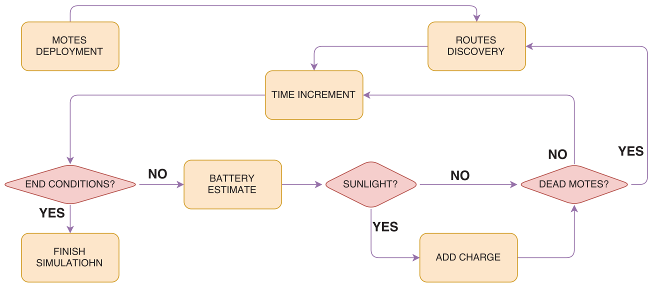

Fig. 5: Flow chart of the WSN simulator.

III. WSN System Design

Our system is a WSN with a base mote to gather all the relevant information and send it to a web server. WSN motes will be spaced 50 m and powered by batteries recharged with a solar cell with the goal of perpetual motes. Three different wireless technologies were considered: WiFi, Bluetooth and ZigBee. Low-power consumption, low data rate and a large number of nodes have led us to choose ZigBee [17]. ZigBee defines three types of nodes in a hierarchical way: a Coordinator that controls and configures the network, Routers that can act as bridges between nodes and End Devices.

A. *WSN Simulator*

Although there are many WSN simulators [18], we have created our own one to account for an uneven space of motes caused by field constraints and different currents provided by the solar cells at the motes[1]. The output of our simulator is threefold. First, it gives the network deployment. Second, it provides routes towards the base station/Coordinator. Finally, it yields the lifetime of the network.

Our simulator features the following global parameters:

- Router proportion: relation between the number of End Devices and Routers
- Mean distance between nodes: the motes will be deployed at a fixed distance plus a random distribution
- Radio range: maximum distance at which two transceivers are able to communicate with each other
- *timeInterval*: time between network synchronizations
- Battery capacity: all the motes have the same maximum charge
- I_m: nominal maximum intensity that each solar cell provides
- (c) Global coefficient: this coefficient accounts for the loss of the power provided by a solar cell when it is outside its maximum power point. This parameter is user-selected in the range [0,1] but could be made time dependant to take into account the varying sunlight intensity depending on the season.
- Current consumptions: current consumed at each stage by the motes

Also, each WSN mote in the simulator features local properties and parameters defined as:

- ID: mote identity
- Position: mote coordinates
- Battery: mAh that the battery holds
- Type: each mote could be an End Device, a Router or the Coordinator
- Neighbors: table of neighbors reachable from the mote of study
- Path: each mote will have a table with the route that a possible message will follow. End Devices will just have the ID of the parent router, while router motes will have the entire path until the base mote
- (*cHarv*) Harvesting Coefficient: each mote, depending on the position, orientation, and sensitivity of its solar cell, will have its own recharging power. To account for this, we have assigned them a coefficient, randomly selected following a Gaussian distribution with $\mu = 0.7$ and $\sigma = 0.3$, that weighs the amount of charge harvested
- (*cRand*) Variable coefficient at each mote: the amount of charge generated at each mote does not only depend on fixed parameters, but also on time-dependent variables, as temperature, shadows caused by clouds and others. *cRand* follows a constant random distribution within [0.5, 1.5] at each simulation step for each mote.

The default parameters of the simulator will be the number of motes (100) and their batteries (2000 mAh). Also, we will set 12 hours of sunlight and 12 hours of night. The maximum intensity that the solar cell provides is fixed at 100 mA. Finally, the radio range is set to 120 m, which is far below the range of XBee in open spaces (1.5 km).

The flow chart with all the processes run by our simulator are collected in Fig. 5. This flow of processes is implemented for an actual WSN deployment in the field. Next subsections describe such processes.

1) Network Deployment: The first step is to deploy the motes in the field. These are settled equally spaced, but with the addition of a random displacement in order to account for an actual distribution, as obstacles or constraints might not permit a WSN with motes evenly spaced. A random function sets whether a given mote will be a Router or an End Device, with the desired proportion selected by the user. The central mote is chosen as Coordinator. Also, the program assigns to each mote a harvesting coefficient, with the previous mentioned Gaussian distribution truncated in the range [0.1,1] to avoid cases better than the ideal one or non-working solar cells.

[1]The source code can be found at https://github.com/DLesta/WSNsimulator

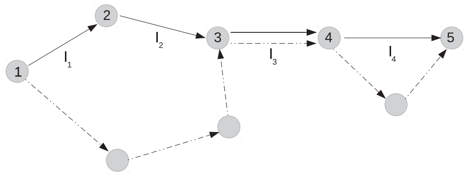

Fig. 6: Path-Cost analysis.

2) Routes Discovery: This process handles two subprocesses. The first one is to update the neighbors table of each mote. This simple subprocess just calculates the distances to the remaining motes and records those located within radio range. The second subprocess is to discover the route that a message will follow from each mote to the base [19, 20]. As the ZigBee standard defines, just the Routers and Coordinator perform the route discovery. End Devices just send the data to the closest router.

The length, L, of a path is defined as the number of devices in the path. In Fig. 6 two different paths are shown, one with L=5 and the other one with L=7. The connections between two nodes are called links. In Fig. 6 the links of the shortest path are numbered from l_1 to l_4.

Parameters such as link quality indication (LQI), energy conservation or number of hops can be used to decide the optimal path for each routing scenario. To simplify this, a *link cost* is associated with each link, and it is determined by the probability of successful packet delivery. The cost of each link for the ZigBee standard, as explained in [20], is defined as $C\{l_i\}$,

$$C\{l\} = \text{The lesser than 7 and } \textit{round}\left(\frac{1}{p_l^4}\right) \tag{1}$$

where p_l is the probability of successful packet delivery that we calculate as,

$$p_l = 1 - \frac{0.8}{\text{radio range}} \cdot distance + rand(-0.1, 0.1) \tag{2}$$

where *distance* is the distance between the motes and *rand(a,b)* is a value from an uniform random distribution between parameters *a* and *b*. Finally, the program calculates all the possible paths, choosing the one that minimizes the total path cost. If several paths have the same total path cost, the first one is chosen.

3) Battery Estimate: At each step of the simulation, the program adds to each mote the harvested charge. The amount of charge depends both on whether the time is at daylight and on the previous mentioned coefficients. Thus, the mean recharge to each mote is:

$$\begin{aligned} Q_{mean} &= I_m \cdot \text{timeInterval} \cdot c \cdot E[\textit{cHarv}] \cdot E[\textit{cRand}] \\ &= 0.7 \cdot I_m \cdot \text{timeInterval} \cdot c \end{aligned} \tag{3}$$

where I_m denotes the nominal maximum of intensity that the solar cell can supply and E[] is the expected value of the distribution, with E[*cHarv*]=0.7 and E[*cRand*]=1 .

Also, the program subtracts the charge consumed in this interval, plus the charge consumed in the transmission of the automatic message to allow the user to know which motes are alive, as will be explained in Section IV. Additionally, the program randomly selects two motes to send a message of detection, and it also subtracts the corresponding energy budget from the involved motes in carrying these messages.

4) Dead and Alive Motes Check: After these processes, the application looks for the motes with the battery value below zero, and deletes them. This forces the network to recalculate the routes. Isolated motes that lack a router to communicate with the base mote are removed too.

5) Program Finish: Finally, the program stops at three different scenarios: 1) the maximum simulation time is achieved, 2) the number of alive motes is under 10 or 3) the base is isolated from all the routers.

B. Simulation Results

The simulator can be configured for any time scheme. In our case we used the timing depicted in Fig. 7, which is the one selected for the implemented WSN, as it will be explained in Section IV. The network is synchronized for information delivery every two hours. This takes 20 s with a measured mean current consumption of 150 mA. The sensing process of snail detection at the capacitive sensor itself lasts 0.3 s with a current consumption of 4.1 mA. The sensing process is repeated every 3 seconds, with the motes in sleep mode for 2.7 s with 200 μA of current consumption.

1) Energy harvesting disabled: In this case, the system was simulated with the solar cells disabled by setting *c*=0. The estimate of the lifetime of every mote is straightforward through the synchronization and the frequency of information delivery. The sensing-sleep cycle is repeated n = 20 times/min = 28800 times/day. These numbers are included in (4), leading to a daily consumption of 24.16 mAh at every mote. As a consequence, a battery of 2000 mAh yields 82.78 days of lifetime, which agrees well with simulations depicted in Fig. 8, where each point is the mean of 20 simulation runs for each *timeInterval*. Longer *timeInterval* values could be chosen, e.g. 4 hours, increasing the lifetime in 20 days. Nevertheless, two hours is an appropriate value to control the snails activity in the plantation.

$$\begin{aligned} Q &= \frac{1h}{3600s}(28800 \cdot (4.1\,\text{mA} \cdot 0.3\,\text{s} + 0.2\,\text{mA} \cdot 2.7\,\text{s}) \\ &\quad + 12 \cdot 150\,\text{mA} \cdot 20\,\text{s}) \\ &= 24.16 \ \text{mAh} \end{aligned} \tag{4}$$

2) Energy harvesting enabled: We have also simulated the network with different values of energy harvesting to find the minimum *c* for a perpetual network. First, simulations to assess the number of alive motes as a function of time were performed for a network of 100 motes. Fig. 9 shows the evolution of the network for a recharged mean value of Q_{mean} =4.9 mAh, which corresponds to *c* = 0.035 at sunny hours for each *timeInterval*. Fig. 9 displays the worst, best and mean cases of 20 simulation runs for every mote. Although some motes die, after two or three years, the solar cells recharge sufficient

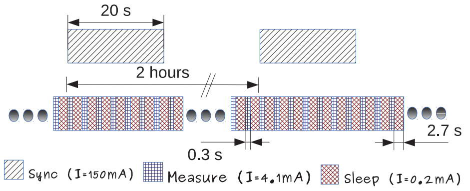

Fig. 7: WSN for snail activity monitoring operation scheme.

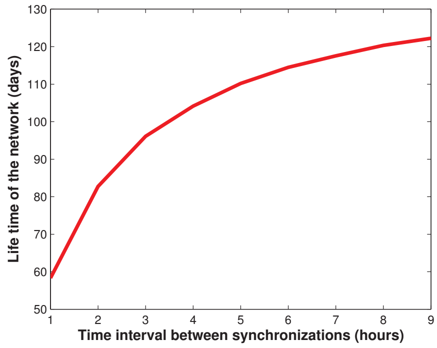

Fig. 8: Lifetime of the network as a function of the time interval of synchronization for the non-harvesting network.

energy to lead to a constant number of motes alive. Fig. 10 shows the results for the mean value of alive motes with different Q_{mean}. Three different conclusions can be drawn from these data:

- Q_{mean} =2.8 mAh (c = 0.02) the lifetime is a little bit increased, but it is still under the required level to allow any mote to survive more than half of a year.
- Q_{mean} = 3.5 and 4.2 mAh (c = 0.025 and c = 0.03, respectively) leads to a network with perpetual motes, but the motes with less recharge die in just a year, leaving the network with only 40 motes. In the next four years around ten additional motes also die.
- Q_{mean} =6.3 mAh recharged each *timeInterval* leads to a practical perpetual network (corresponding to c = 0.045). In this case around 20 motes die within 1.5 years and then the number remains constant.

A similar information is extracted from Fig. 11, which shows the mean battery of all the motes in the network. As seen, the mean charge of the network tends to be constant after some years, meaning that the most of the alive motes at certain time will reach a stable battery value.

These simulations have been obtained with a very conservative approach, as Q_{mean} =6.3 mAh each *timeInterval* with daily conditions amount to only c=4.5% of maximum capacity of the solar cells, which is even low for the case of a mote in the shade. Moreover, in these simulations all the motes awake their radio transceiver to send their status to the base. This is done only for testing purposes, but in a real case they would just need to awake when information have to be sent, or with a much lesser frequency to resynchronize their timer. This, according to (4), would decrease by almost a half the energy consumption. All these reasonings and data show that achieving perpetual motes for snail detection is feasible.

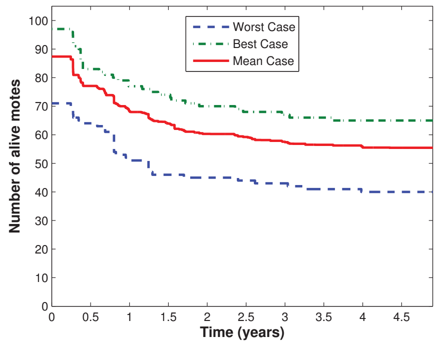

Fig. 9: Number of alive motes as a function of time with solar cell enabled (Q_{mean} =4.9 mAh).

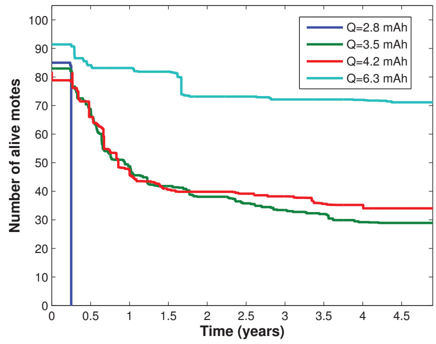

Fig. 10: Mean number of alive motes for different values of Q_{mean}.

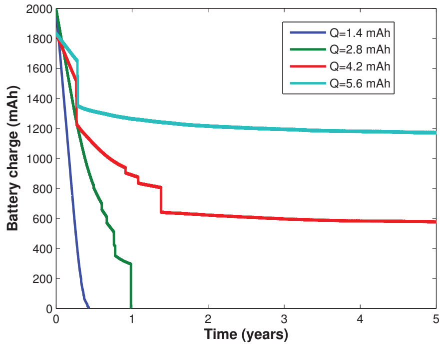

Fig. 11: Mean battery of the network for different values of Q with the solar cell enabled.

Fig. 12: Tests performed in Santiago de Compostela in April 2016.

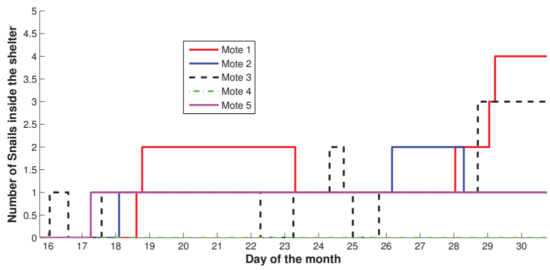

Fig. 13: Number of snails inside each shelter as a function of the day in a 2 weeks experiment.

IV. WSN System Implementation

With the same time scheme of Fig. 7, signals are locally processed at each WSN mote and only the relevant events are sent. Snails are detected with a succession of a raising and a falling edge over a mean value which accounts for the offset caused by environmental issues as temperature and humidity. Unpredictable behavior caused by snails standing over the copper electrodes is circumvented by finishing the algorithm if a falling/raising edge has not been measured for 15 minutes.

For mote-server communication two different possibilities have been developed. The first one and most suitable for large agricultural areas without infrastructure is the GPRSbee module [21], which is a GPRS/GSM board based on the SIM900 chip. The other solution was to send the information directly to a PC through a different XBee and to use the PC to upload the data to the server.

Communications between nodes were implemented within XBee S2B PRO modules. These modules are ZigBee compliant and they allow a point-to-point communication scheme. All the motes send the information to the base in sequences as the one shown in (5), where the events can be an input/output or noisy signal begin/end, and it is later uploaded to the server.

$$\{\underbrace{\text{mote ID}}_{\text{1 byte}}, \underbrace{\text{time stamp}}_{\text{2 bytes}}, \underbrace{\text{event}}_{\text{1 byte}}, \underbrace{\text{time stamp}}_{\text{2 bytes}}, \underbrace{\text{event}}_{\text{1 byte}}, \ldots\} \quad (5)$$

For this purpose, the XBee of the base mote is configured as a Coordinator, and it will automatically handle the association requests and network formation while the microcontroller will manage the rest of the functions. It first requests the actual time to the web server, then it performs the capacitive sensor calibration to finally send the begin operation message to all the motes. With this command, the whole network sets the XBee in sleep mode and begins to process the incoming signal, storing the events detected until the next network synchronization. When this time is reached, the base sends a request command to all the motes, and it gathers all the information and uploads it to the server.

End devices operation is simpler than the base operation. They perform the calibration when they are turned on, waiting for the base mote to send the start command. As this command is the same whether the base has been initiated at this moment or a data collect process has finished, an End Device can be incorporated to the network at any time. When this command is received, the device begins to measure values and process them, storing the result with a time stamp if some detection is achieved. Once the elapsed time exceeds *timeInterval*, the mote under study waits the time required to set the network along with the request command. After that, it sends the events detected along with the command to make the base know that it is alive. Finally, it waits for the start command and begins again the same process.

V. Experimental Results

As a proof-of-concept, five motes were constructed and tested in Santiago de Compostela (Spain). The network was deployed in a space of 4 m^2, in a small lettuce plantation (see Fig. 12). This plantation was surrounded by a plastic with a special type of lubricant at the top to avoid snails leaving the area. The first day 25 snails were released into the enclosure and their activity was recorded for two weeks between April 16 and 30, 2016. Also, among this field validation of the system, the WSN was shown in a test configuration at [22], with all the radios always on for real time events detection.

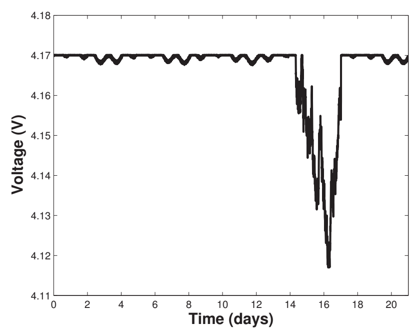

Fig. 14: Battery voltage in a three weeks experiment.

A. Signals detected

Fig. 13 shows the experimental results plotting the number of snails inside each shelter as a function of the day. As can be seen, there are motes with low levels of occupancy, e.g. mote 4 which does not show activity at all and mote 5 with just one input on day 17, while others as motes 1 and 3 show high levels of activity. Signals corresponding to snails standing over the electrodes are not shown in this graph. Such signals were captured at mote 4 on day 19 for 6 hours and at mote 1 on days 22 and 26 for 8 and 6 hours respectively. As there is no way to know whether the number of snails changed during this time, the previous occupancy level was assumed not to change in this model. Finally, the status of the motes (whether they are alive or not) was sent every 2 hours to the server with the conclusion that all the motes remained alive during the experiment, as it was expected from our WSN simulator.

B. Power consumption

Regarding the power consumption of the motes, according to simulations they should have a stable amount of remaining battery. To check if the assumption of battery recharged per day was realistic, the voltage supplied by the battery (which is the best indicator of the remaining battery) should be stable. This was measured during 3 weeks with different weather conditions (see Fig. 14). The results show that just around one hour of sun irradition per day is enough for the battery to supply its maximum again (4.17 V). The minimum measured voltage was 4.115 V after three days of cloudy and rainy conditions, which corresponds to a remaining battery over its 95%. Also, it can be seen how the slight voltage decay observed during the night is rapidly replenished at dawn, which is shown as ripples in Fig. 14.

C. Scalability

The scalability of the system is guaranteed because of the communication protocol selected, as its MAC layer already implements strategies for handling many accesses to the medium. If the number of motes were bigger than the maximum number of simultaneous accesses, which is unlikely, different time requests could be implemented.

A different source of scalable problems could be that, even when the protocol manages correctly the access to the medium and all the information arrives to the base, this one might not handle all the information due to saturation in its input buffer, leading to information loss. The solution for this would be to change the processor unit of the base mote by other with higher computational power. The consequent increase of power consumption would be fixed with solar cells and batteries of larger capacities.

VI. Conclusions

This work has shown the design of a ZigBee WSN with perpetual motes for terrestrial snail activity monitoring. The design has been made with an own WSN simulator which accounts for real-life conditions as an uneven space of motes in the field, or different currents collected by the solar cells of the motes. Simulations have been used for energy estimates in a small WSN with five motes. Field tests in such a WSN for several weeks show that the network is perpetual in terms of energy consumption, validating simulations. An upscaling to larger plantations is feasible in terms of energy. In terms of communications, an upscaling to larger plantations would oblige to include more power computing and accordingly larger batteries and solar cells in the base/Coordinator mote to handle more information. The resultant network could be used both to trigger an alarm of early snail presence or to further elaborate statistical models of snail detection with environmental variables as temperature or humidity.

References

[1] S. Karthiga, V. Jegathambigai, M.D.S.D. Karunarathne, A. Svinningen, and G Mikunthan. Snails and slugs damaging the cut foliage, cordyline fruticosa and use of biorationals towards their management. *Communications in agricultural and applied biological sciences*, 77(4):691698, 2012.

[2] H. Hussein, D. Al-Rajhy, F. El-Shahawi, and S.M. Hashem. Molluscicidal activity of pergularia tomentosa (l.), methomyl and methiocarb, against land snails. *International Journal of Pest Management*, 45(3):211–213, 1999.

[3] G.H. Baker. Helicidae and Hygromiidae as pests in cereal crops and pastures in South Australia. *Molluscs as Crop pests*, pages 193–215, 2002.

[4] J. Castillejo, I. Seijas, and F. Villoch. Slug and snail pests in spanish crops and their economical importance. In *Slug & snail pests in agriculture. Proceedings of a Symposium, University of Kent, Canterbury, UK, 24-26 September 1996.*, pages 327–332. British Crop Protection Council, 1996.

[5] R. Vogler, A. Beltramino, M. Sede, D. Gutiérrez Gregoric, V. Núñez, and A. Rumi. The giant African snail, Achatina fulica (Gastropoda: Achatinidae): Using bioclimatic models to identify South American areas susceptible to invasion. *American Malacological Bulletin*, 31(1):39–50, 2013.

[6] A. Ramos A. Bugarín, M. Lama and J. Janeiro. *METIS*. https://citius.usc.es/transferencia/oferta-tecnoloxica/listado/metis-tecnoloxia-soporte-decision-agricultura, Accessed May 2017.

[7] S. Azfar, A. Nadeem, and A. Basit. Pest detection and control techniques using wireless sensor network: a review. *Journal of Entomology and Zoology Studies*, 3(2):92–99, 2015.

[8] A. Baggio. Wireless sensor networks in precision agriculture. In *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden*. Citeseer, 2005.

[9] J. Gutiérrez, F.J. Villa-Medina, A. Nieto-Garibay, and M.Á. Porta-Gándara. Automated irrigation system using a wireless sensor network and gprs module. *IEEE Transactions on Instrumentation and Measurement*, 63(1):166–176, Jan 2014.

[10] F. Viani, M. Bertolli, M. Salucci, and A. Polo. Low-cost wireless monitoring and decision support for water saving in agriculture. *IEEE Sensors Journal*, PP(99):1–1, 2017. Early access.

[11] H. Wang, A. Fapojuwo, and R. Davies. A Wireless Sensor Network for Feedlot Animal Health Monitoring. *IEEE Sensors Journal*, 16(16):6433–6446, 2016.

[12] D. García-Lesta, E. Ferro, V. M. Brea, P. López, D. Cabello, J. Iglesias, and J. Castillejo. Capacitance-based wireless sensor mote for snail pest detection. In *Sensors Applications Symposium (SAS),IEEE*, pages 1–6, April 2015.

[13] E. Ferro, V.M. Brea, D. Cabello, P. López, J. Iglesias, and J. Castillejo. Wireless sensor mote for snail pest detection. In *SENSORS, 2014 IEEE*, pages 114–117. IEEE, 2014.

[14] D. Meeker. *Finite Elements Methods Magnetics 4.2*. http://www.femm.info Accessed May 2017.

[15] Analog Devices. *AD7153 Datasheet*, 2016. www.analog.com, Accessed May 2017.

[16] J.M. Campbell, D.C. Dahn and D.A.J. Ryan. Capacitance-based sensor for monitoring bees passing through a tunnel. *Measurement Science and Technology*, 16(12):2503, 2005.

[17] J.S. Lee, Y.W. Su, and C.C. Shen. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In *Industrial Electronics Society IECON 33rd Annual Conference of the IEEE*, pages 46–51. IEEE, 2007.

[18] B. Musznicki and P. Zwierzykowski. Survey of simulators for wireless sensor networks. *International Journal of Grid and Distributed Computing*, 5(3):23–50, 2012.

[19] ZigBee Alliance et al. Zigbee specification, 2006.

[20] S. Farahani. ZigBee and IEEE 802.15.4 Protocol Layers. In *ZigBee wireless networks and transceivers*, chapter 3, pages 90–94. Newnes, 2011.

[21] *GPRSbee official site*. http://gprsbee.com/, Accessed May 2017.

[22] D. García-Lesta, E. Ferro, V. M. Brea, P. López, D. Cabello, J. Iglesias, and J. Castillejo. Live Demonstration: Wireless Sensor Network For Snail Pest Detection. In *International Symposium on Circuits and Systems (ISCAS), 2016 IEEE*, May 2016.

Daniel García-Lesta received the Physics degree from the University of Santiago de Compostela (USC), Spain, in 2014 and he received his Master's Degree in Electronics Systems for Information and Communication from the National Distance Education University (UNED), Spain, in 2016. Currently he is working toward his PhD at Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), University of Santiago de Compostela. His research topic is smart CMOS vision sensors.

Diego Cabello (M'96) received the BSc and PhD degrees in Physics from the University of Granada, Granada, Spain, and the University of Santiago de Compostela, Santiago de Compostela, Spain, in 1978 and 1984, respectively. Currently, he is a Professor of Electronics at Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), University of Santiago de Compostela, Spain. He has been the Dean in the Faculty of Physics between 1997 and 2002, and the Head of the Department of Electronics and Computer Science between 2002 and 2006, both in the University of Santiago de Compostela. His main research interests lie in the design of efficient architectures and CMOS solutions for computer vision, especially in early vision.

Esteban Ferro received the Physics degree from the University of Santiago de Compostela (USC), Spain, in 2012 and he received his Master's Degree in Microelectronics from the University of Seville and Microelectronics Institute, Spain, in 2015. Currently he is working toward his PhD at Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), University of Santiago de Compostela. His research topic is micro-energy harvesting.

Paula López (M'02) received her PhD degree in Physics in 2003 and is currently an associate professor at the University of Santiago de Compostela. Her research interests include the design of mixed signal integrated circuits for image processing applications and the physical modeling of electronic devices, particularly CMOS image sensors, as well as the translation of these models into hardware description languages. She is author or coauthor of more than 50 research papers.

Víctor Manuel Brea received his PhD in Physics in 2003. Currently he is an Associate Professor at Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), University of Santiago de Compostela, Spain. His main research interests lie in the design of efficient architectures and CMOS solutions for computer vision, especially in early vision, as well as micro-energy harvesting.