Simulation Architecture for Data Processing Algorithms in Wireless Sensor Networks

Yann-Ael Le Borgne, Mehdi Moussaid, Gianluca Bontempi
ULB Machine Learning Group
Computer Science Department
Universite Libre de Bruxelles
1050 Brussels - Belgium
Email: {yleborgn, memoussa, gbonte}@ulb.ac.be

***Abstract*— Wireless sensor networks, by providing an unprecedented way of interacting with the physical environment, have become a hot topic for research over the last few years. As with any new technology, results from real experimentations using these networks are still scarce, as real deployments are either costly, or still unfeasible in the current state of technology. There is therefore an increasing need for simulation tools allowing the testing of different architectures, communication protocols or information processing algorithms in sensor networks. In this paper, we investigate a simulation framework for the testing of data processing in wireless sensor network applications. In a first stage, data is generated using partial differential equations, allowing the modeling of a large panel of physical phenomena. In a second stage, sensing unit operating system and network constraints are simulated using an instance of a versatile simulator to account for the platform characteristics. Insights provided by the proposed simulation frame are illustrated by a set of experiments on a heat source detection task.**

I. Introduction

Recent advances in low power microelectronics have lead to the development of small sensor modules, capable of sensing, processing, storing, and wirelessly transmitting information [1]. According to the technology trend, it is expected than in a few years time, the size and price of these modules will be small enough for them to be seamlessly deployed in large quantities over an environment, thereby providing information about the environment with an unprecedented level of spatial and temporal accuracy.

Applications for these networks are envisioned in a wide variety of domains, such as precision agriculture, civil engineering, scientific research, industry, medical health care, or defense. The potential of this technology has been clearly demonstrated, and this has lead an ever increasing number of research groups, both in academy and industry, to design and implement methods to efficiently operate wireless sensor networks (WSN) [1], [3]. Indeed, wireless sensor networks differentiate themselves from other types of ad-hoc networks, particularly in terms of very limited, and often irreplaceable, energy resources. The challenge therefore resides in designing energy efficient strategies to extract useful data from the network, from an energy-aware MAC layer at the network level, to a compact delivery of information at the end-user level.

The quality and reliability of data processing algorithms proposed for WSNs, such as in-network compression or clustering [1], [2], are however often difficult to estimate in real world conditions, as the cost associated to large real world deployments has remained prohibitive for most research institutes. Performance estimations of proposed methods are therefore commonly assessed by means of simulation systems, with simplified models that often overestimate the reliability of sensor modules. Furthermore, as few real world deployments have been so far realized, there is also a lack of real world data sets for testing WSN data processing algorithms. Data is therefore often simulated in a fairly simplistic way, such as linear functions of space and time, or multi-gaussian data distributions.

A number of recent publications on feedbacks of real world deployments have pointed out the fact that better tools were needed to simulate wireless sensor characteristics [4], [5]. Adequacy of the simulated environment to reality is a critical factor in the assessment of a data processing algorithm for WSN. Indeed, weak quality of modeling is likely to lead to methods or strategies that will not be transposable to real world deployments, thereby wasting part of research efforts.

In order to improve the quality of a WSN simulation for in-network data processing algorithms, we propose in this paper to couple a synthetic data generation system based on partial differential equations (PDEs) to the TOSSIM wireless sensor network simulator [6]. On the one side, PDEs allow to simulate many different natural physical phenomena, such as heat transfers, chemicals diffusion, wave propagation, or electrostatic fields. They can therefore bring an important contribution in the assessment of data processing algorithms for WSN, by improving the realism of data fed in the sensor network simulator. On the other side, TOSSIM is the simulator for TinyOS, a flexible and widely used open source operating system for sensor modules, for which many contributions from the research community have been made [7].

Overview of the architecture is presented in section II. Section III will cover the modeling richness of PDEs and section IV will discuss the simulation of sensor networks for data processing algorithms. Finally, in section V, we show, by the means of a high level prediction task, how the proposed simulation system can provide insights into tradeoffs between

network capacity and data processing algorithm accuracy.

II. Overall architecture

Information retrieval from an environment on the basis of a wireless sensor network follows a sequence of different steps. This section presents the different aspects that need to be taken into account to provide an accurate modeling of the sensing process. A summary of the sensing process is given on figure 1.

At a high level, two independent processes can be identified,

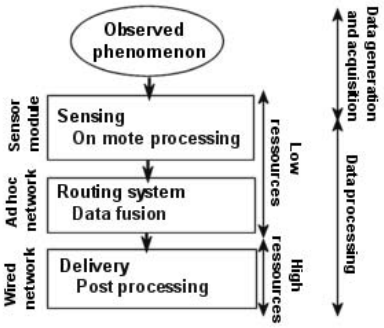

Fig. 1. Different stages of the data retrieval process.

the former being the data generation stage, that is captured by the set of sensor modules, and the latter being the data processing stage, which is shared among the different components of the network.

Modeling of the first stage involves simulating physical variations that occur in an environment. Solution proposed in this paper relies on PDEs, whose richness in modeling natural phenomena is illustrated in section III.

Once at the sensor level, data is routed and potentially processed through the network before its delivery to the end user. This stage involves a set of complex interactions between the different network components, and is also constrained by network component resources, which ought to be identified for accurate modeling. These aspects will be addressed in section IV which presents the network simulation component of the architecture proposed in this paper.

III. Realistic synthetic data generation

A wide range of variations in an environment, such as temperature, humidity, chemicals concentration, vibrations, pressure, and so forth, can be captured by sensors. In many cases, these variations can be modeled by sets partial differential equations, which consequently offer the possibility to improve the realism of simulated data.

We review hereafter some of the most versatile first order partial differential equations, by giving their mathematical expressions together with the type of natural phenomenon they can model.

A. Notations

In the following, let $V(\mathbf{x},t)$ be the value of a monitored variable V at point $\mathbf{x}$ and time t, where vector $\mathbf{x}$ is the coordinate vector of point $\mathbf{x}$.

To simplify the notation, we also use the Laplacian operator $\Delta V(\mathbf{x},t) = \sum_{i=1}^{k} \frac{\delta^2 V(\mathbf{x},t)}{\delta x_i^2}$, where k is the dimension of the space considered, and x_i the corresponding coordinate of point $\mathbf{x}$ in dimension i.

B. PDEs for natural phenomenon modeling

1) Laplace's equation: This equation is defined by

$$\Delta V(\mathbf{x},t) = 0$$

This equation can model the behaviors of electric, gravitational, and fluid potentials. It can be used to simulate static force fields.

2) Wave equation: Helmholtz's equation is expressed as follows:

$$\Delta V(\mathbf{x},t) - \frac{1}{c^2}\frac{\delta^2 V(\mathbf{x},t)}{\delta t^2} = 0$$

This equation can model wave propagation phenomena, such as sound, light, fluid, or electromagnetic wave propagation, as well as vibrations along different materials. The parameter c is here related to the wave speed. Example of propagation of a wave propagation phenomenon is given on figure 2. The wave source is situated at the right hand side of the diamond shaped box.

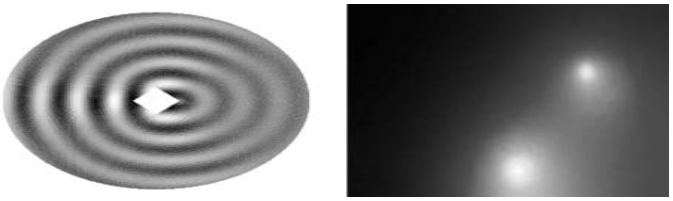

Fig. 2. Examples of screen shot for wave and diffusion processes.

3) Diffusion equation: The general form of the diffusion equation is of the form:

$$k\Delta V(\mathbf{x},t) - \frac{\delta V(\mathbf{x},t)}{\delta t} = 0$$

This equation governs a diffusion phenomenon, and k is the thermal diffusivity in the case of heat propagation, or diffusion coefficient in the general case, of the medium. Commonly modeled phenomena using this equation include diffusion of gases or fluids, heat propagation, or erosion processes. Example of two heat sources in a rectangular environment with dissipation at the boundaries is given on figure 2.

4) Other types: Examples given above are far from exhaustive, and only aimed at giving an overview of the potential modeling power of partial differential equations for use in a sensor network simulation frame. Other examples of interest could be the advection equation or the Euler Tricomi equation, for transport of dissolved product in water or airflow over supersonic aircraft, respectively [8].

IV. Sensor network simulation

As an emerging research field, there currently exists a large number of projects proposing different hardware and operating systems alternatives for designing and operating sensor networks [1]. In this section, we first review the main current sensor modules, and then review the possibilities for simulating data processing and routing in sensor networks.

A. Sensor platforms

A sensor module typically consists of five components - sensing hardware, memory, battery, embedded processor and trans-receiver. While sensor module of the millimeter scale have already been designed in research laboratories (SPEC, deputy dust [1]), these are still research prototypes, and will likely not be found on the market before a few years time.
Characteristics of the main currently available wireless sensor platforms on the market are detailed in table I. Sizes of these

TABLE I
Main sensor modules available on the market.

Type	Features
MICA2 (Crossbow)	8MHz ATmega128L microcontroller 4KB EEPROM, 128KB Flash, 512KB data storage 38.4kbps radio
BTnode (ETH Zurich)	8MHz ATmega128L microcontroller 4KB EEPROM, 64KB RAM, 128KB Flash Bluetooth radio + 800, 900, 2.4GHz radio
ESB (Univ. Berlin)	8MHz TI MSP430 microcontroller 64KB EEPROM, 2KB RAM 19.2kbps radio
Tmote (MoteIV) TelosB (Crossbow)	8MHz TI MSP430 microcontroller 10KB RAM, 48KB Flash, 512KB data storage Integrated sensors. 250kbps 2.4GHz radio

platforms mostly depend on their battery, typically 2 AA batteries. In terms of energy efficiency, platforms based on the TI MSP430 microcontroller are in the current state of technology the most advanced ones, draining a current of about 30mA in their fully active mode and less than 1μA in their idle mode.
Other platforms, while not commercialized (for example Imote (Intel), or Eyes nodes (Eyes european project)) also exist, and have about the same characteristics as those detailed in table I.

B. Simulating wireless sensor networks

For testing data processing algorithms, we argue that key factors that need to be taken into account are of two types.
First, the simulation frame should be able to assess the feasibility of the algorithm by imposing a real implementation, that can be compiled and run on existing sensor modules. We here argue that this aspect is crucial as actual implementation can reveal important flaws of algorithms such as memory needs or computability, as sensor modules have very scarce resources. An example of this is the fact that no sensor platform has yet the ability to execute floating point computations, and implementation would in this case reveal the overhead induce by finding a work around in the design of the algorithm for its practical use.
The second key factor that needs to be considered is the communication between modules. Communication induces two critical aspects that may impair the theoretical validity of a data processing algorithm in a sensor network. First, one can not suppose network connectivity to be perfect and link variability should be a feature of the simulation frame. Second, synchronization aspects, from code execution to network communication should also be considered, as they play a major role in the real deployment of a sensor network.
There currently exists a wide variety of simulation systems for sensor networks, from sensor module micro controller to network communication. For example, from the network point of view, NS2 [10] is the prominent simulator, allowing accurate simulations of wired and wireless networks, up to thousands of nodes. While benefiting of a wide panel of standard and experimental communication protocols and network architectures, it however lacks simulation of code execution on network's nodes. ATEMU, and more recently AVRORA [11], offers cycle accuracy in machine code level simulations for sensor module platforms based on the AVR assembly language, and provides network simulation up to several hundreds of nodes. It is however limited to AVR micro controllers, and cannot simulate code execution for other types of micro controllers. Other projects, such as EmStar [12] or SENS [13], allow to simulate interactions in heterogeneous networks, involving low-resources sensor modules and higher resources microserver platforms, by focusing on higher level interoperability issues.
To our knowledge, the best candidate that currently allows the simulation, with reasonable accuracy, of mote code execution and network communication is TOSSIM [6], the simulator for TinyOS. TinyOS is an open source event driven system for programming on the main current types of sensor modules, features a great number of program contribution, and benefits of a large community of users [7]. Program language is the NesC, a variation on C, and can be compiled both for TI MSP430 and ATmega128L AVR microcontrollers, which are currently the most widely used in sensor module technology. TOSSIM simulates through a virtual clock at a 4MHz rate the actual code that would run on a sensor module. It also reproduces the complete network stack of TinyOS at the bit level allowing, through a simple mechanism, to simulate network contention or packet corruption. Finally it offers, although simplified, easy generation of network models for connectivity, and a model for energy consumption.

V. Case study

In this section, we present a complete simulation of a wireless sensor network for a high level data prediction task application. The purpose of the application is to predict, given a set of temperature readings collected by the sensor network, how many heat sources are present in the environment. The proposed simulation frame will allow to put into evidence

trade offs in the number of sensors used in this application.

A. Environment modeling

1) Problem description and modeling: The studied environment is a box, in which an unknown number of heat sources can appear or disappear, and the problem is to predict the number of heat sources contained inside the box through the temperature variations observed on the top of its surface. Applications for such a problem could be, for example, a piece of machinery in an industrial context, or a burrow in which one would aim at detecting the presence of animals. We also suppose in this environment an opening inducing a heat dissipation process, accounting, for example, for a ventilation system or an access to the environment.
Modeling of the temperature variations due to conduction on one surface of the box can be obtained by using the Fourier partial differential equation, governing heat diffusion process:

$$k\Delta V(\mathbf{x},t) - \frac{\delta V(\mathbf{x},t)}{\delta t} = f$$

and by associating Neumann conditions $V'(\mathbf{x},t) = g$ to each edge of the surface, except for the portion modeling the operture where a Dirichlet condition imposing a fixed temperature $V(\mathbf{x},t) = u$ is defined. Actual values chosen are mostly qualitative as the purpose of the modeling was to obtain a realistic diffusion propagation phenomenon. Parameters were chosen so as to observe temperature variations of about 5 degrees in a five minutes period, namely $k = 0.1m^2.s^{-1}$ for thermal diffusivity, $f = 100W$ for a heat source, $u = 30$ for the Dirichlet condition, and $g = 0$ for the Neumann condition. The box was 80cm long, 50cm wide, and 20cm high. Illustration of a snapshot of the modeled surface in the case of two heat sources is given on figure 3.
Potential appearances of heat sources took place at random

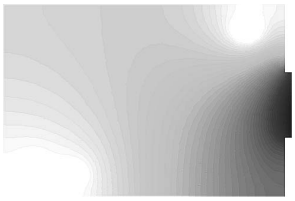

Fig. 3. Example of two heat diffusion processes observed in the simulated environment. Access point is on the right side.

spots in the environment. The number of concomitant heat sources varied from one to three. Ten tuples (1, 2 or 3-tuples) of heat source appearance spots were randomly generated. For each of these configurations, heat sources were left for 400 seconds in the environment, and then removed. Data was collected at every time instant, at 659 different randomly locations on the surface, for the 400 seconds during which the source was present, and for another 400 seconds to collect data when the surface cooled down. All in all, 800 time instants by 10 position sets by 3 conditions, i.e. $N = 24000$ observations were collected.
In this problem, we also assume that optimal sensor location can not be decided in advance. The 659 spots at which temperature were collected serve as a basis in the rest of this section to determine the a priori number of sensors needed to achieve the prediction task within some accuracy. The data set of generated values (inputs) are stored together with the corresponding number of heat sources (output), and is referred to as D_N.

2) Prediction task and assessment: The problem tackled here is a prediction task with four possible outputs, corresponding to the number of heat sources present in the environment. Solution to this problem requires identifying a prediction model

$$y = h(x, \alpha_N)$$

i.e. a mapping h with parameters α_N, that transforms *an input vector* x, providing information about the environment state, into an *output value* y, here the number of heat sources. A prediction model is entirely determined by the knowledge of its parameters α_N, and identification of these parameters is performed by means of a supervised learning algorithm based on an observation data set, referred to as D_N, of N samples. There exists a wide variety of prediction models, such as decision trees, neural networks or K-nearest neighbors, to which are generally associated different learning procedures. As the purpose of this paper is not to discuss the best model for this specific heat source detection problem, we chose among the family of possible prediction techniques the lazy learning approach, which exhibits interesting features for sensor network contexts, by handling missing values and on line learning [14].
This technique stores all observed data during supervised stages (i.e when the relation between inputs and output is known) in an observation data set, and postpones the design of a prediction model until a prediction query is made. When, given a set of input data from the environment, a prediction is asked to the lazy learning system, a statistically relevant number of neighbors to the query are retrieved from D_N, and a local linear model is built to determine the output corresponding to the given input.
For each the following experiments, assessment of the generalization ability of the prediction model was based on a 10-fold cross validation procedure. As we made no assumption about a predefined placing for the sensors, we generated twenty different sets of sensor positions for each network size. Reported prediction accuracies are averages of the 10-fold cross validation procedure described above over the twenty random sensor arrangements.

B. Interdependence of learning and network design

In this section, we first assess the best strategy to use for this prediction problem independently of the network constraints. We then consider its implementation and simulation in TOSSIM, to put into evidence limitations in the expected accuracy of the algorithm due to packet collisions.

1) Data aggregation: The first approach to the problem is to consider a central server, such as a desktop PC, that collects at regular time instants all readings from the sensor networks, and achieves the prediction task by associating each sensor reading to an input of the prediction model. Let this be the *strategy 1*. As illustrated on figure 4, left, prediction accuracy decreases for more than five sensors in the simulated problem considered here, because of the well known problem of a too-high input space dimensionality [15], that prevents the learning algorithm from finding the right parameters for the prediction model (results are not given for network sizes superior to 40 sensors due to the computational limits).
A way to avoid this effect is by aggregating input variables. In the present case, we alternatively investigated the use of sensor readings' mean and variance as inputs to the prediction model, thus bounding to two the number of input variables to the prediction model, while still capturing essential information about the number of heat sources in the environment. Let this be the strategy 2, whose results are reported on figure 4, left. As the number of input variables is not dependent on the number of sensors, prediction accuracy increases together with the number of sensors used, and yields from 5 sensors better prediction accuracies (more than 90%) than the optimum of strategy 1.

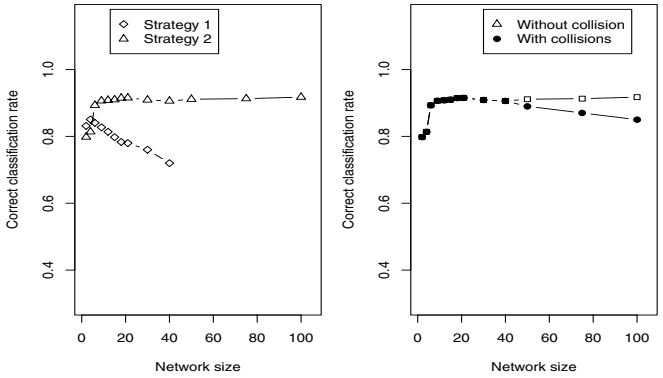

Fig. 4. Left: Comparison of prediction accuracy between strategies 1 and 2 for varying network sizes. Right: Collision impact on prediction accuracy as network size increases.

2) Network contention: Through an aggregation strategy, the prediction task considered here will gain in accuracy as the number of sensors increases, by incorporating more information in the aggregate variables. However, if increasing the number of sensors improves the accuracy of the prediction task, it also impairs the quality of the network transmissions due to collision effects. To study this effect, the radio model of TOSSIM can provide insight into this parameter. Simulation results of packet collisions were run in TOSSIM for varying number of sensors, using the radio model of the MICA2 sensor platform. In this simulation, we required each sensor to send its value every second. Results obtained are summarized on figure 4, right, showing that above 40 sensor modules this effect starts impairing the prediction accuracy.

VI. Conclusion

The key point of this paper was to stress the need for a simulation frame for data processing algorithms in sensor networks, from data generation to network simulation. The proposed simulation framework mainly emphasized the coupling of a data generator to a WSN simulator. Regarding data generation, we suggested PDEs as they can provide models for many environmental phenomena that will typically be monitored by WSN. Concerning WSN simulation, we suggested TOSSIM as an interesting simulator for assessing the implementability and the performances of a data processing algorithm. While other solutions, as was discussed, may however be considered regarding WSN simulators, experiments driven in the case study showed that the proposed framework could identify some of the trade offs that must be considered between sensor module constraints and the network characteristics, and performance and feasibility of a data processing algorithm.

Acknowledgment

This work was supported by the **COMP^{2}SYS** project, sponsored by the Human Resources and Mobility program of the European Community (MEST-CT-2004-505079). Authors would also like to acknowledge anonymous reviewers for their helpful comments.

References

[1] M. Ilyas, I. Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, 1st edition, CRC Press, 2005.
[2] F. Zhao, L. Guibas, *Wireless Sensor Networks, An Information Processing Approach*, Morgan Kaufmann, 2005.
[3] Y. Le Borgne, G. Bontempi, "Round Robin Cycle for Predictions in Wireless Sensor Networks". In: *Proceedings of ISSNIP 2005*, December 2005, Melbourne, Australia.
[4] J. Polastre, R. Szewcyk, A. Mainwaring and D. Culler, "Lessons from a Sensor Network Expedition.", In: *Proceedings of European Workshop on Sensor Networks*, Berlin, Germany, January 2004.
[5] J. Thelen, D. Goense and K. Langendoen, "Radio Wave Propagation in Potato Fields", In: *Proceedings of WNM workshop (co-located with WiOpt 2005)*, Riva del Garda, Italy, April 2005.
[6] P. Levis, N. Lee, M. Welsh, Woo, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications", In: *Proceedings of ACM SensSys*, Los Angeles, California, USA, 2003.
[7] http://www.tinyos.net
[8] A.D. Polyanin, *Handbook of Linear Partial Differential Equations for Engineers and Scientists*, Chapman Hall/CRC, 2002.
[9] J. Polastre, R. Szewczyk, D. Culler, "Telos: Enabling ultra-Low power wireless research", In: *Proceedings of the 4th International Conference on Information Processing in Sensor Networks*, Los Angeles, USA, 2005.
[10] http://nsnam.isi.edu/nsnam/index.php/Main_Page
[11] BL Titzer, DK Lee and J Palsberg, "Avrora: Scalable sensor network simulation with precise timing", In: *Proceedings of IPSN*, Los Angeles, California, USA, April 2005.
[12] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, "EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks", In: *Proceedings of the USENIX Technical Conference*, Boston, MA, June 2004.
[13] S. Sundresh, W. Kim, and G. Agha, "SENS: A Sensor, Environment and Network Simulator", In: *Proceedings of 37th Annual Simulation Symposium*, Arlington, VA, USA, 2004.
[14] G. Bontempi, Y. Le Borgne, "An adaptive modular approach to the mining of sensor network data", In: *Workshop on Data Mining in Sensor Networks*. SIAM SDM April 2005, Newport Beach, USA.
[15] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.