A bio-inspired approach for data dissemination in wireless sensor networks

LUIS ENRIQUE PALAFOX-MAESTRE
JOSÉ ANTONIO GARCÍA-MACIAS

CICESE - Research Center
Computer Science Department/Applied Physics Division
Km. 107 Carretera Tijuana-Ensenada C.P. 22860
Ensenada, B.C., México
{palafox,jagm}@cicese.mx

Abstract. Recently, biologically-inspired algorithms have been presented as an alternative for designing many aspects of wireless ad-hoc and sensor networks. In this paper, we propose the adaptation of a bio-inspired algorithm called the "infection algorithm" for the energy-efficient dissemination of data from a sensor field to the sink node. Furthermore, we present a series of experiments with real data, gathered through the use of an agricultural monitoring application, and also simulations that validate the efficiency of our proposal.

Keywords: sensor networks, data dissemination, bio-inspired algorithms.

(Received January 26, 2006 / Accepted May 29, 2006)

1 Introduction

Many differences have been observed between wireless sensor networks (WSN) and more traditional data communication networks; consequently, the algorithms, protocols and techniques used in traditional networks are not adequate for WSNs. Perhaps the most important factor in these differences is the scarcity of resources in WSN nodes; this includes limitations in resources such as processing power, storage capacity, communication range and power availability.

We observe some interesting features in wireless sensor networks that suggest some similarities with networks composed of biological organisms:

- Small and simple nodes. The nodes on sensor networks are very simple devices that are commonly integrated by a processing unit, a sensing unit, a communication unit (usually using RF), and the energy source.

- Limited communications and processing capabilities. Given the fact that the nodes are very simple devices, these cannot handle complex algorithms nor have embedded sophisticated communications protocols. The typical operation of a sensor node is basically to read (sense) data from the environment, process this data in a simple manner and communicate its results to a neighbor node.

- Localized behavior. Considering the fact that sensor nodes only receive data from their neighbors, they only act based on local information. Furthermore, they cannot perform distributed actions such as the ones performed by more sophisticated data networks, e.g., remote procedure calls (RPC).

- Optimized use of resources. Given the fact that the available resources for these networks are very limited, the nodes in the network must use these resources in a very efficient way. Of particular importance is power management, due to the fact that sensor nodes do not have an unlimited power source; these sensors usually operate on the power supplied by small batteries, such as the ones commonly used in personal consumer electronics.

The previous sensor network features suggest some intriguing similarities between WSNs and biological networks such as the ones integrated by cells, ants, bees, and other insects communities. In these biological networks, the individuals can be seen as relatively simple entities whose actions are only based on local information, i.e. communicating with their neighbors, trying to be efficient in their work, etc. These similarities have been noted by researchers in the data networks field, and some ideas are starting to appear on recent publications showing proposals and even preliminary results.

This paper is structured as follows: section II gives an overview on the application of bio-inspired approaches for the design and modelling of several aspects of wireless sensor networks. In section III we introduce the infection algorithm, which is inspired on natural epidemics and was originally applied to the problem of stereographic images correspondence; then, section IV outlines how data dissemination in WSNs has been managed, and how we use the infection algorithm for this purpose. In section V we present results obtained from a series of experiments and simulations showing that the proposed algorithm estimates fairly accurate values that could be used on certain sensor networks applications while reducing power consumption. We conclude this paper with some final remarks and some general ideas of future work.

2 Bio-inspired approaches for WSNs

Given the stringent resource limitations in WSNs, it is not feasible to implement traditional algorithms and techniques on them, because these techniques do not make efficient use of available resources. Thus, new approaches have been explored, among them bio-inspired approaches [3].

An empirical study conducted by Ganesan et al. [7] introduces the concept of epidemic algorithms to describe the behavior of network protocols that allow fast data dissemination through the use of local interactions. In this work, a sensor network is built consisting of 169 nodes (13 x 13 grid), and many experiments were performed using a simple epidemic algorithm for data dissemination (using a gossiping technique). The interesting part of this work was the fact that such a simple algorithm would exhibit such a complex behavior, as the authors show in the results section of their report.

Biological automatons have been proposed to define the behavior and the interactions between nodes in a sensor network [3]. This was done through the design and implementation of a kOS (kilobit Operating System), which is a light version of an operating system that is executed by the nodes of a sensor network. The authors also present the results obtained in the SEA-COAS project [13], on which buoys containing sensor nodes with RF units (for data communications) were deployed to monitor sea bed movement.

Also, feedback loops have been introduced for the autoconfiguration of sensor nodes [6]. In this work, the behavior of the human blood pressure auto-regulation process is used to model network auto-configuration, which the authors posit that has the structure of a closed feedback loop. This type of behavior is used to automatically configure robot assisted mobile sensor networks such as the ROSES (Robot Assisted Sensor Networks) platform which is described briefly in the paper [6].

In the work presented in [15], the authors use the spontaneous synchronization process of fireflies and propose the *Reachback Firefly Algorithm* (RFA) which is used to implement time synchronization in sensor networks. Furthermore, this algorithm takes into account real world radio effects such as packet loss and network latency.

As evidence suggests, the use of algorithms based on biological systems has generated very interesting results so far. We believe that these type of algorithms constitute a tool that presents a great opportunity for research in the design of sensor network technology.

3 The Infection Algorithm

The "*infection algorithm*", presented by Olague et al. [12], is based on the concept of natural epidemics and was applied for searching corresponding points in stereo images, while reducing the number of processing operations required to do so compared to the traditional exhaustive search method.

Transition rules were used for correspondence searching, similarly to cellular automata. The rules entries depended on the current state of surrounding neighbors (pixels). The neighborhood considered in this work was 25 neighbors (9 close neighbors and 16 external), these 25 neighbors where contained in a 7x7 window that was centered on the point of interest.

The infection process in this case evolved over the image according to the previously mentioned set of rules that changed the current state of the pixel depending on the state of its surrounding neighbors. Four states were defined for this algorithm:

- Healthy individuals (Not-exposed). Nothing has been decided yet for that pixel.
- Sick individuals (Exposed). The pixel has been computed using constraints of dense stereo matching.
- Infected individuals (Proposed). The value of the pixel is guessed based on the state of its neighbors. Some conflicting information from various neighbors prevent to fix its status at this time.
- Immune individuals (Automatically allocated). All the information from the neighbor is coherent and the guess value has been set.

Summarizing, the algorithm was defined as follows:

1. All pixels in the image were initialized to the *not-exposed* state.
2. The maximum interest pixels were extracted. And their state was set as *exposed*.
3. The transition rules were applied to every pixel in the image, except to those whose state was *automatically allocated* or *exposed*.
4. While there are still pixels that are not in the *automatically allocated* nor *exposed* states, go to step 3.

The main goal of this algorithm is to find the maximum number of correspondences according to the defined rules. The authors mentioned [12] that these rules were defined on a case by case basis, but the criteria for setting these rules was not specifically mentioned.

An interesting aspect of this proposal is the fact that, compared with the use of exhaustive search, the proposed algorithm can deliver savings of up to 50 percent in the number of operations (in some cases of up to 99 percent). We think that some of the general ideas of this algorithm could be very useful for wireless sensor networks. For instance, the algorithm acts in its entirety based on local information, estimating values in the neighborhood of one pixel; this can be translated to important savings in processing, which in WSNs would result in energy savings as well. In WSN applications where there are heavily populated networks with nodes that sense some environmental parameter such as temperature or humidity, it is highly probable that the sensed data of a node is very similar to the sensed data of a neighbor node. Therefore, it is very appealing to use the infection algorithm to read only a small number of nodes and estimate the values of their neighbors, resulting in significant amounts of energy savings.

4 Efficient Data Dissemination in Wireless Sensor Networks

One of the main problems in the application layer in the wireless sensor networks protocol stack is the efficient dissemination of data, typically from the sensor field to the sink(s) [1]. There is some work related with that problem such as Cougar [5][16][17] and TinyDB [10][9][11], these papers present the use of an SQL-like declarative language to query data from sensor networks. However, in these papers the authors do not go into details on how the queries are processed by the sensor networks.

There has been also some work related to data aggregation in wireless sensor networks and the use of estimation techniques for reducing power consumption. Particularly, in [2] the authors present a distributed algorithm that uses correlation functions to estimate an aggregated value and reducing power consumption in relation to the snapshot aggregation approach, but in this case the authors only seem to consider scalar aggregated functions such as *max*, *min* and cannot be applied to functions such as *average* nor *count*.

In this paper we propose an alternative method inspired by the infection algorithm mentioned in the previous section; this method represents an interesting alternative to data dissemination guided by the savings in power consumption, which is the main design factor for protocols in the different layers of the wireless sensor networks protocol stack [1].

The original approach of the infection algorithm [12] was to reduce the number of processing operations required. However, in the case of sensor networks, the main priority is the efficiency in power consumption; for this reason the main objective of our proposal is to reduce the amount of the operations involving the

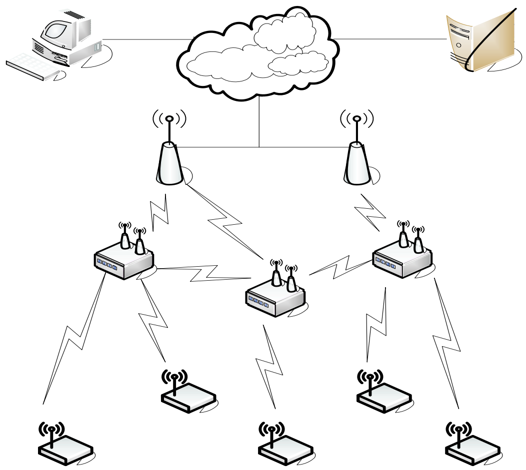

Figure 1: A hierarchical wireless sensor network.

RF unit and not necessarily to reduce the amount of processing operations. This, mainly because we have observed the technological trend to increase processing power in sensor networks platforms in the near future [8].

One way of reducing the amount of communications operations is, at the time of making a data request to a cluster of nodes in a wireless sensor network, not to pass the query to all of the nodes that belong to such a cluster. Thus, in brief, our proposal consists on selecting a subset of nodes that belong to a hierarchical WSN as shown in Fig. 1, where each one of the nodes has a direct link to a cluster leader (also called cluster head). In the selected subset of nodes, an on-demand read would be made through which an estimation of the remaining nodes would be possible using the infection process.

The entity responsible for performing the selection process must be the cluster head, this cluster head typically must have more power and processing resources than the rest of the nodes. It is important to note that the selection criteria is very flexible and it can comply with the specific need of a particular sensor network. Some typical criteria for the selection can be, for instance, the nodes with the higher power level in the cluster, the nodes that are located in a strategic position to infect a greater number of nodes in a smaller number of iterations, or even a random selection of nodes.

Undoubtedly, the most critical part of the proposed infection algorithm for WSNs is how the infection process takes place; this process is based on the information belonging to the neighboring sensor nodes. To estimate the value of an unknown node we use the correlation with its neighbors, this correlation is calculated and stored in the cluster head node from previous readings. Specifically, in our proposal we calculate a correlation matrix for each one of the eight neighbor nodes surrounding the one in particular (we could consider more than eight neighbors for a better estimation in exchange for a higher computational cost and more storage space required). This correlation matrixes must be stored in the cluster head node, and this node must also calculate its initial values at startup by performing a initial reading on all of the nodes that belong to the cluster, and down the line, this cluster head node must also be able to update the stored matrixes using the values of future readings belonging to contiguous nodes in the same cluster of the WSN.

The correlation matrixes N, S, E, W, A, B, C, D correspond to the neighbors to the north, south, east, west, northwest, northeast, southwest and southeast directions respectively of a sensor grid of m rows and n columns. Every single one of the nodes that belong to this grid must have a direct link to a special node called cluster head as has been noted above. Additionally, $r_{(i,j)}$ represents the data read at the node located in row i and in column j.

In equation 1 we have matrix N, note that in the first row of matrix N we have basically a vector or zeroes, this is because the sensor nodes corresponding to the first row of the grid do not have any neighbors to the north:

$$N = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{r_{(2,1)}}{r_{(1,1)}} & \frac{r_{(2,2)}}{r_{(1,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(1,n)}} \\ \frac{r_{(3,1)}}{r_{(2,1)}} & \frac{r_{(3,2)}}{r_{(2,2)}} & \cdots & \frac{r_{(3,n)}}{r_{(2,n)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{r_{(m,1)}}{r_{(m-1,1)}} & \frac{r_{(m,2)}}{r_{(m-1,2)}} & \cdots & \frac{r_{(m,n)}}{r_{(m-1,n)}} \end{bmatrix} \tag{1}$$

similarly, in equation 2 we have matrix S, on which the last row of the sensor grid do not have southern neighbors, therefore the correlation matrix has all zeroes on the last row:

$$S = \begin{bmatrix} \frac{r_{(1,1)}}{r_{(2,1)}} & \frac{r_{(1,2)}}{r_{(2,2)}} & \cdots & \frac{r_{(1,n)}}{r_{(2,n)}} \\ \frac{r_{(2,1)}}{r_{(3,1)}} & \frac{r_{(2,2)}}{r_{(3,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(3,n)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{r_{(m-1,1)}}{r_{(m,1)}} & \frac{r_{(m-1,2)}}{r_{(m,2)}} & \cdots & \frac{r_{(m-1,n)}}{r_{(m,n)}} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{2}$$

in equation 3 we show the correlation matrix for the eastern neighbors:

$$E = \begin{bmatrix} \frac{r_{(1,1)}}{r_{(1,2)}} & \frac{r_{(1,2)}}{r_{(1,3)}} & \cdots & \frac{r_{(1,n-1)}}{r_{(1,n)}} & 0 \\ \frac{r_{(2,1)}}{r_{(2,2)}} & \frac{r_{(2,2)}}{r_{(2,3)}} & \cdots & \frac{r_{(2,n-1)}}{r_{(2,n)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{r_{(m,1)}}{r_{(m,2)}} & \frac{r_{(m,2)}}{r_{(m,3)}} & \cdots & \frac{r_{(m,n-1)}}{r_{(m,n)}} & 0 \end{bmatrix} \quad (3)$$

in equation 4 we show the correlation matrix for the western neighbors:

$$W = \begin{bmatrix} 0 & \frac{r_{(1,2)}}{r_{(1,1)}} & \frac{r_{(1,3)}}{r_{(1,2)}} & \cdots & \frac{r_{(1,n)}}{r_{(1,n-1)}} \\ 0 & \frac{r_{(2,2)}}{r_{(2,1)}} & \frac{r_{(2,3)}}{r_{(2,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(2,n-1)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{r_{(m,2)}}{r_{(m,1)}} & \frac{r_{(m,3)}}{r_{(m,2)}} & \cdots & \frac{r_{(m,n)}}{r_{(m,n-1)}} \end{bmatrix} \quad (4)$$

the equation 5 shows the correlation matrix A for the northwestern neighbors:

$$A = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{r_{(2,2)}}{r_{(1,1)}} & \frac{r_{(2,3)}}{r_{(1,2)}} & \dots & \frac{r_{(2,n)}}{r_{(1,n-1)}} \\ 0 & \frac{r_{(3,2)}}{r_{(2,1)}} & \frac{r_{(3,3)}}{r_{(2,2)}} & \dots & \frac{r_{(3,n)}}{r_{(2,n-1)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{r_{(m,2)}}{r_{(m-1,1)}} & \frac{r_{(m,3)}}{r_{(m-1,2)}} & \dots & \frac{r_{(m,n)}}{r_{(m-1,n-1)}} \end{bmatrix} \quad (5)$$

the equation 6 shows the correlation matrix B for the northeastern neighbors:

$$B = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \frac{r_{(2,1)}}{r_{(1,2)}} & \frac{r_{(2,2)}}{r_{(1,3)}} & \cdots & \frac{r_{(2,n-1)}}{r_{(1,n)}} & 0 \\ \frac{r_{(3,1)}}{r_{(2,2)}} & \frac{r_{(3,2)}}{r_{(2,3)}} & \cdots & \frac{r_{(3,n-1)}}{r_{(2,n)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{r_{(m,1)}}{r_{(m-1,2)}} & \frac{r_{(m,2)}}{r_{(m-1,3)}} & \cdots & \frac{r_{(m,n-1)}}{r_{(m-1,n)}} & 0 \end{bmatrix} \quad (6)$$

in equation 7 we show the correlation matrix C for the southwestern neighbors:

$$C = \begin{bmatrix} 0 & \frac{r_{(1,2)}}{r_{(2,1)}} & \frac{r_{(1,3)}}{r_{(2,2)}} & \cdots & \frac{r_{(1,n)}}{r_{(2,n-1)}} \\ 0 & \frac{r_{(2,2)}}{r_{(3,1)}} & \frac{r_{(2,3)}}{r_{(3,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(3,n-1)}} \\ 0 & \vdots & \vdots & \ddots & \vdots \\ \vdots & \frac{r_{(m-1,2)}}{r_{(m,1)}} & \frac{r_{(m-1,3)}}{r_{(m,2)}} & \cdots & \frac{r_{(m-1,n)}}{r_{(m,n-1)}} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad (7)$$

and finally, in equation 8 we show the correlation matrix D for the southwestern neighbors:

$$D = \begin{bmatrix} \frac{r_{(1,1)}}{r_{(2,2)}} & \frac{r_{(1,2)}}{r_{(2,3)}} & \dots & \frac{r_{(1,n-1)}}{r_{(2,n)}} & 0 \\ \frac{r_{(2,1)}}{r_{(3,2)}} & \frac{r_{(2,2)}}{r_{(3,3)}} & \dots & \frac{r_{(2,n-1)}}{r_{(3,n)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \frac{r_{(m-1,1)}}{r_{(m,2)}} & \frac{r_{(m-1,2)}}{r_{(m,3)}} & \dots & \frac{r_{(m-1,n-1)}}{r_{(m,n)}} & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (8)$$

Additionally, if $\sigma_{M(i,,j)}$ is the correlation of the node located at row i and at column j corresponding to matrix M. We could estimate the value of a node $\hat{r}_{(i,j)}$ as an average of previous readings and/or estimations from their neighbors. Thus for every node (i,j) the estimated value is given by equation 9:

$$\hat{r}_{(i,j)} = \frac{\begin{array}{c} \sigma_{A(i,j)} r_{(i-1,j-1)} + \sigma_{N(i,j)} r_{(i-1,j)} + \\ \sigma_{B(i,j)} r_{(i-1,j+1)} + \sigma_{W(i,j)} r_{(i,j-1)} + \\ \sigma_{E(i,j)} r_{(i,j+1)} + \sigma_{C(i,j)} r_{(i+1,j-1)} + \\ \sigma_{S(i,j)} r_{(i+1,j)} + \sigma_{D(i,j)} r_{(i+1,j+1)} \end{array}}{L} \quad (9)$$

where L is the number of neighbors on which a reading and/or previously estimated data exist, and additionally $r_{(i,j)} = 0$ for values of i, j that do not belong to the intervals $1 \leq i \leq m$ and $1 \leq j \leq n$.

Evidently, to estimate the value of a node we need to know the value of one of its eight contiguous neighbors in the specific case of our previously described topology. However, if we know in advance the values of a greater number of neighbors we could anticipate that the resulting estimation would be more accurate. On the contrary, if we don't have a reading or estimation of any of the surrounding neighbors of the node, the estimation of its reading would not be possible under this scheme.

An important factor for the outcome of the infection algorithm for WSN is the infection threshold value λ, which is defined as the minimum number of neighbors required (i.e. the neighbors' data must be known either by a previous estimation or by a physical reading) to perform the estimation of a given node. This parameter can be set arbitrarily, however, its value would dictate how fast the infection process evolves in the sensor network. This is because we can establish intuitively that with a small value, the infection propagates faster and therefore the number of required operations for our algorithm would be smaller, but if this value is small we are also sacrificing accuracy in our estimations. In contrast, if we set λ to a larger value, we would need

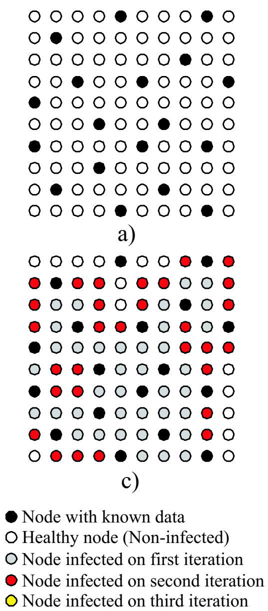

Figure 2: Example of the infection process in a WSN.

a greater number of iterations for the infection process to propagate throughout the entire network, but consequently, we would obtain more accurate estimations of the nodes. For obvious reasons, the possible values for λ must be between one and the maximum number of neighbors (eight in our case).

In Fig. 2 we present an example that shows how the infection process propagates in a sensor network consisting of a grid of 10x10 sensor nodes, in this example the value of λ is set to 2, indicating that we require to know at least two of the values of the surrounding neighbors nodes in order to perform the estimation of the current node. Note that in this particular case we only require three iterations of the infection algorithm in order to infect all of the nodes in the network.

In the example presented in Fig. 2, as we can see clearly in Fig. 2d, the entire sensor network was infected, which implies that all of its nodes' values are either known or estimated. However, this is not always the case, because depending on the value assigned to λ and the position of the nodes read initially, we could have a network where at the end of the execution of the algorithm we would still have some healthy nodes, called *immune nodes*. For this reason, we would need to find a suitable value for λ and a proper selection criteria that minimizes the number of immunes nodes. The value selection criteria for this parameter and the selection criteria for the initially read nodes are beyond the scope of this paper and could pose an interesting topic of research in the area.

5 Experiments and simulations

We now present the results obtained through a series of experiments and simulations to evaluate the proposed data dissemination algorithm for WSN. Through them, we measure the accuracy of the predicted data, as compared with the actual data gathered with our monitoring application; we also estimate the power savings that results from reduced radio communications.

5.1 Experiments

The site on which the experiments took place was a greenhouse where tomato plants were being cultivated. The dimensions of the greenhouse are 22 meters wide, 8 meters long and 4 meters tall. Nine sensor nodes were deployed in the greenhouse (Berkeley motes class MICAz model MPR2400 with a sensor board model MTS310). Ideally, we would have liked to perform the experiments with a more densely populated sensor network, but unfortunately due to a lack of resources it was not possible. However, in the simulations section we will present the results obtained on higher density sensor networks.

Temperature readings were captured in the greenhouse during a period of approximately two hours with a sampling frequency of one reading every eight seconds. Several experiments were conducted while changing the number of sensor nodes physically being read; the selection criterion that we applied was to select first the nodes that reported the higher level of remaining power in their batteries. The algorithm was executed using a λ value of 2 for 2 nodes, then for 3 nodes and so on up to 9 nodes (obviously with 9 we have no error because we query the entire sensor field explicitly).

In Fig. 3 we present a graph containing the average of the RMSE (Root Minimum Square Error) obtained from each experiment of our algorithm and the resulting power saved (as a percentage) for each case. The error comes from comparing the actual data read in the field, with the data that our algorithm predicts. The percentage of power savings represents how many nodes were actually used for processing and communicating

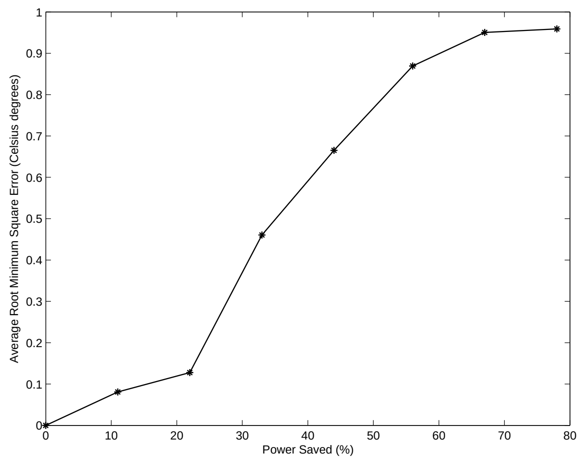

Figure 3: Plot of the percentage of power saved by the algorithm vs. average RMSE of the resulting estimations.

data, i.e., 90% savings would mean that only 10% of the nodes were actually used for reading, calculating, and communicating values, and the rest were sleeping. Note that as we would have intuitively expected, with a larger number of selected nodes we would minimize the average RMSE of our estimations but the power saved would be also minimum as opposed to selecting a smaller number of nodes.

Another important fact observed in the experiments is that for a larger number of selected nodes (greater than four in this particular scenario) we required a smaller number of iterations (only one in our case) as opposed to selecting just a few nodes (less than five) on which case we needed two iterations of the infection algorithm.

5.2 Simulations

The simulations were conducted considering 100 sensor nodes in a 10x10 grid. The temperature values used in the simulations were generated through a pseudo-random function; values ranged from 26.018 to 36.569 degrees Celsius (according to those observed in our greenhouse experiments).

The proposed algorithm was programmed in Matlab and the initial reading values were assigned arbitrarily from the pool of values obtained with the pseudo-random function. The main tasks performed in the simulation were:

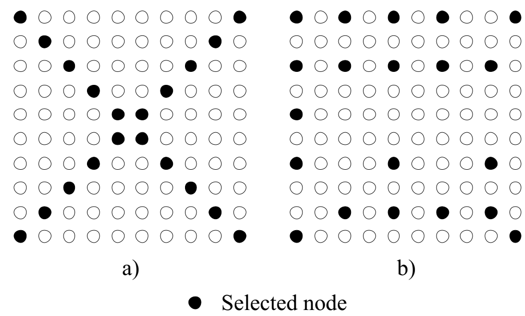

Figure 4: Pattern of selected nodes for the simulations.

- First, the program calculates the correlation matrixes of the generated temperature values.
- Next, a subset of nodes is selected. In this case we made an arbitrary selection of nodes, however, in a real setting some selection criteria (e.g., power level) could be used.
- Finally, we estimated the values of the remaining nodes in the sensor grid.

Analogously to the experiments, for determining the accuracy of predicted values we compared with the pool of randomly generated temperature data. We ran the algorithm with different values of λ and selecting different numbers of nodes to simulate physical readings.

In our first simulation we selected a subset of 20 nodes that follow a pattern consisting of two diagonal lines across the grid (refer to Fig. 4a) and executed our algorithm with λ values of 2, 3 and 4.

Fig. 5 shows that with a smaller λ value we need fewer iterations of the algorithm in order to compute the entirety of the estimations. However, intuitively, we can state that with a larger value of λ we can obtain more accurate estimations because we are using more neighborhood data in order to estimate each value. One drawback of using a large value of λ is the fact that there is a certain upper bound of that value on which we cannot estimate the entire grid field. In our case, the upper bound value is 4, because with the pattern that we used for node selection there are no nodes in the grid that have 4 neighbors with known values. therefore, in this case we cannot estimate any values.

In our second simulation, we selected the same number of nodes (20), but we selected more strategically

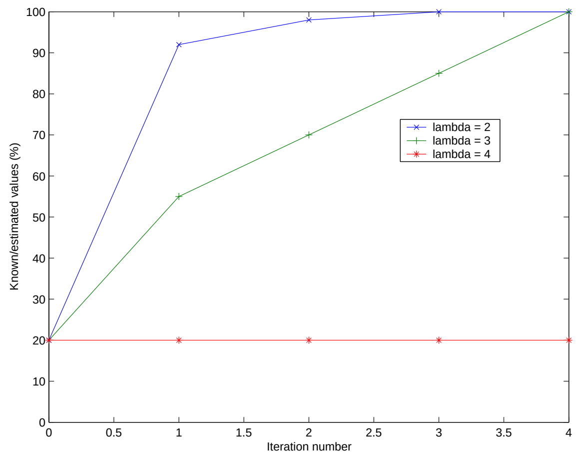

Figure 5: Plot of percentage of estimated values for the first simulation.

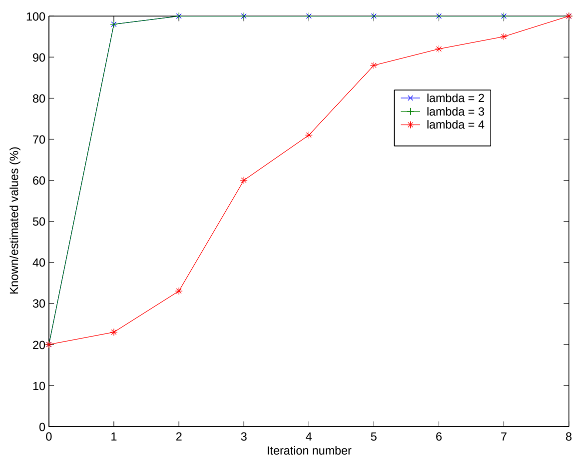

Figure 6: Plot of percentage of estimated values for the second simulation.

located nodes (shown in Fig. 5b) and executed the algorithm using the same λ values as in the previous simulation. In Fig. 6 we show that, in contrast to our previous simulation, with a λ value of 4 we can estimate the entire sensor grid, the drawback of this scenario is that the algorithm estimates the values slower (i.e. requires more iterations), but in return, intuitively, we can say that we have more accurate estimations. Note that the plots of $\lambda = 2$ and $\lambda = 3$ are superposed (i.e. the algorithm estimates the values with the same number of iterations), so in this particular scenario there is no additional computational cost of raising the λ value from 2 to 3, therefore, in practice, it would be more cost-effective to use the higher value (3 in this case).

Finally, after observing the results of the experiments and simulations, we can conclude that selecting the input parameters of our algorithm (selected subset of nodes and λ value) is a complex optimization problem that involves accuracy, computational cost and required energy for querying the sensor grid. This problem is not addressed in this paper, but will be the subject of future work.

6 Final Remarks and Future Work

We have proposed the use of a biologically-inspired algorithm for the energy-efficient data dissemination in wireless sensor networks. The results, obtained through empirical tests and simulations, show that important power savings can be made and that the accuracy of predicted data is adequate. Therefore, we consider that this approach holds great promise for the design and development of low-power and processing-efficient protocols. Regarding the applicability of our proposed algorithm, it should be noted that data should be uniform and not show "holes", discontinuities or great local variations. For instance, typical monitoring applications (temperature, humidity, etc.) are good candidates for using our algorithm; on the contrary, fire detection for instance is not a good application.

After these encouraging results, we are now working on some outstanding aspects, such as refining the criteria for determining the convergence of data physically read by the sensor nodes; this will then be used to initiate the phase where only some selected nodes will actually read the data. More accurate power savings estimations are needed as well. We are also trying to deploy a larger scale platform in an open agricultural field to run long duration experiments and gain further insight regarding the applicability and possible limitations of our proposal.

References

[1] Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.*, 40(8):102–114, 2002.

[2] Boulis, A., Ganeriwal, S., and Srivastava, M. B. Aggregation in sensor networks: an energy-accuracy trade-off. *Ad Hoc Networks*, 1(2-3):317–331, 2003.

[3] Britton, M., Shum, V., Sacks, L., and Haddadi, H. A biologically-inspired approach to designing wireless sensor networks, 2005.

[4] Culler, D. E. and Hong, W. Wireless sensor networks - introduction. *Commun. ACM*, 47(6):30–33, 2004.

[5] Demers, A., Gehrke, J., Rajaraman, R., Trigoni, N., and Yao, Y. The cougar project: a work-in-progress report. *SIGMOD Rec.*, 32(4):53–59, 2003.

[6] Dressler, F., Kruger, B., Fuchs, G., and German, R. Self-organization in sensor networks using bio-inspired mechanisms, 2005.

[7] Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., and Wicker, S. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical report, Intel Research Labs @ Berkeley, 2002.

[8] Hill, J., Horton, M., Kling, R., and Krishnamurthy, L. The platforms enabling wireless sensor networks. *Commun. ACM*, 47(6):41–46, 2004.

[9] Madden, S., Franklin, M. J., Hellerstein, J. M., and Hong, W. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[10] Madden, S. and Hellerstein, J. M. Distributing queries over low-power wireless sensor networks. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 622–622, New York, NY, USA, 2002. ACM Press.

[11] Madden, S., Szewczyk, R., Franklin, M. J., and Culler, D. Supporting aggregate queries over ad-hoc wireless sensor networks. In *WMCSA '02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, page 49, Washington, DC, USA, 2002. IEEE Computer Society.

[12] Olague, G., de Vega, F. F., Pérez, C. B., and Lutton, E. The infection algorithm: An artificial epidemic approach for dense stereo matching. In Yao, X., Burke, E. K., Lozano, J. A., Smith, J., Guervós, J. J. M., Bullinaria, J. A., Rowe, J. E., Tiño, P., Kabán, A., and Schwefel, H.-P., editors, *PPSN*, volume 3242 of *Lecture Notes in Computer Science*, pages 622–632. Springer, 2004.

[13] Seacoas project website. http://www.ee.ucl.ac.uk/secoas/secoas/.

[14] Toh, C. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, 2001.

[15] Werner-Allen, G., Tewari, G., Patel, A., Welsh, M., and Nagpal, R. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, pages 622–622, New York, NY, USA, 2005. ACM Press.

[16] Yao, Y. and Gehrke, J. The cougar approach to in-network query processing in sensor networks. *SIGMOD Rec.*, 31(3):9–18, 2002.

[17] Yao, Y. and Gehrke, J. Query processing in sensor networks. In *CIDR '03: Proceedings of the First Biennial Conference on Innovative Data Systems Research*, Asilomar, Ca, USA, January 2003.