

# Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems

Guillermo G. Riva

Department of Electronic Engineering,

Universidad Tecnologica Nacional, Cordoba, Argentina

Tel: +54-3515986039 E-mail: [griva@scdt.frc.utn.edu.ar](mailto:griva@scdt.frc.utn.edu.ar)

Jorge M. Finochietto

Digital Communications Lab - CONICET

Universidad Nacional de Cordoba, Cordoba, Argentina

Tel: +54-3514334147 E-mail: [jfinochietto@efn.uncor.edu](mailto:jfinochietto@efn.uncor.edu)

Received: August 7, 2012 Accepted: November 11, 2012 Published: December, 2012

DOI: 10.5296/npa.v4i4.2206

URL: <http://dx.doi.org/10.5296/npa.v4i4.2206>

## Abstract

Monitoring spatio-temporal continuous fields using wireless sensor networks (WSNs) has emerged as a novel solution. An efficient data-driven routing mechanism for sensor querying and information gathering in large-scale WSNs is a challenging problem. In particular, we consider the case of how to query the sensor network information with the minimum energy cost in scenarios where a small subset of sensor nodes has relevant readings. In order to deal with this problem, we propose a Pheromone-based In-Network Processing (PhINP) mechanism. The proposal takes advantages of both a pheromone-based iterative strategy to direct queries towards nodes with relevant information and query- and response-based in-network filtering to reduce the number of active nodes. Additionally, we apply reinforcement learning to improve the performance. The main contribution of this work is the proposal of a simple and efficient mechanism for information discovery and gathering. It can reduce the messages exchanged in the network, by allowing some error, in order to maximize the network lifetime. We demonstrate by extensive simulations that using PhINP mechanism the query dissemination cost can be reduced by approximately 60% over flooding, with an error below 1%, applying the same in-network filtering strategy.

**Keywords:** Bio-inspired Networking, Computational Intelligence, In-Network Filtering, Monitoring System, Routing Algorithms and Protocols, Swarm Intelligence, Wireless Sensor Networks.

## 1. Introduction

A wireless sensor network (WSN) consists of a large number of small, low cost, and energy-constrained wireless nodes which can be deployed over an area to monitor physical phenomena such as temperature, air/water pollution, noise levels, etc. In this sense, monitoring spatio-temporal continuous fields using WSNs has emerged as a novel solution. However, sensor nodes have constrained energy, processing, storage and communication capabilities mainly due, among others reasons, to the fact that they are portable battery-powered devices [1], [2]. For this reason, an efficient resource management is a key issue in the design of WSNs [3]. Due to energy consumption in data communication, it is better to increase in-network computation tasks in order to reduce the message exchange (e.g. the data transmission of 1 kB is energetically equivalent to compute 3 millions of instructions approximately). However, the development of energy-efficient processing and routing strategies for query dissemination and data collection to minimize the data exchange and to maximize the network lifetime is a challenging problem, especially in environments where only a small subset of sensor nodes has relevant readings, and no information about the location of these nodes is available. Data monitoring requires not only an initial discovery stage, but also a continuous search for new relevant data due to field variations in time. One solution to this problem is to allow nodes to cooperate and jointly decide which data are relevant and their locations. Trails from sink to nodes with relevant readings can be generated in a distributed fashion, as members of insect colonies do using pheromone-based indirect communication.

In this work, we address the problem of monitoring spatio-temporal continuous fields using WSNs, where only a subset of nodes has relevant readings, but no information about the node's location is available a priori. The goal is to minimize the number of exchanged messages between nodes in the query routing, in order to prolong the network lifetime. Figure 1 shows the application of a WSN to obtain and monitor minimum values of a continuous field, where only a small set of sensor nodes has relevant information. Sink (red node) is located at the center of the WSN deployment. In this sense, we are interested in problems such as what is the top (i.e., maximum) reading?, or more generally, what are the 10% top readings?, and then monitor these points over time.

We give two examples of application of the proposed mechanism. In the first one, a WSN deployed over an oil spill in the ocean can efficiently obtain the points of maximum oil concentration, and track the spill's direction of displacement over time. If source nodes, based on a distributed algorithm for location [4], send also information about their geographic location to the sink, a map with the location of relevant information can be obtained. In the second one, a WSN can be applied to monitor the silted sea in a harbor. Minimum values of water depth are the relevant information to obtain, in order to avoid running aground large

and heavy ships. The main goal of this work is how to query the sensor network information, reducing the number of exchanged messages involved in the query dissemination.

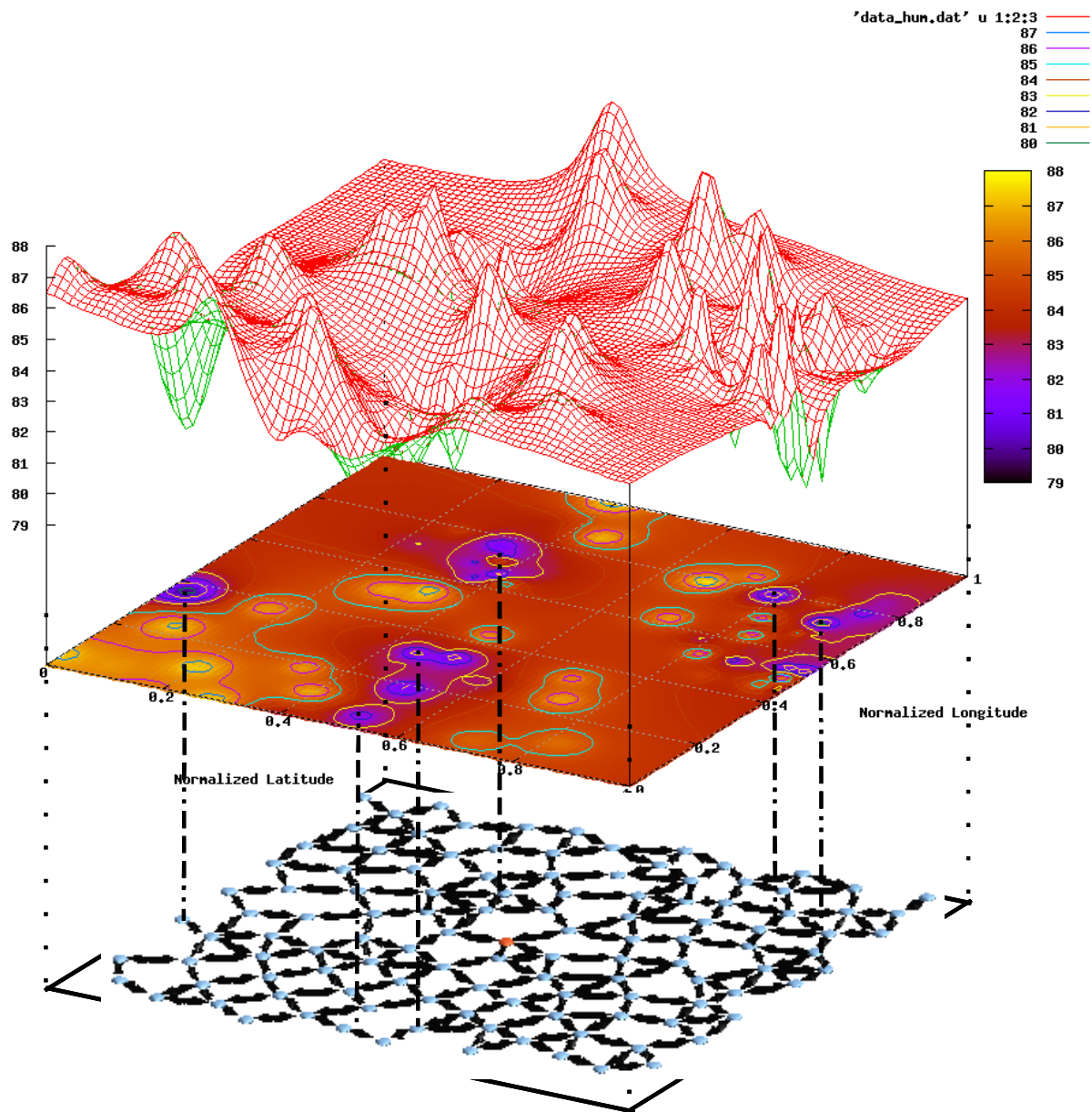


Figure 1. Monitoring of minimum values of a continuous field using a WSNs.

The simplest solution to deal with these problems is to let sensor nodes send their readings to the sink node which processes data in a centralized way (warehousing approach). A more efficient approach, especially in large-scale WSNs, is to process inside the network the readings and send to the sink node only those which are considered relevant. This last approach is known as *in-network processing* (INP).

Among INP techniques, *filtering* is the one that has less computational and memory requirements. Other techniques like data aggregation and data fusion typically demand more resources. Filtering can be implemented either at *data-level*, by deleting repeated data messages, due to the high spatial correlation, which is dependent of the resolution of the

sensor readings, or at *node-level*, by selecting only a subset of sensor nodes to send their readings to the sink. In this paper, we focus on the second case. Generally, node selection can be implemented by either *fixed* or *adaptive filtering* using static or dynamic thresholds, respectively. Static thresholds are used for searching readings within a specific range of values  $X$  [5], [6], while dynamic thresholds, for readings without a predefined range or when there is not information of the data range. In this last case, the threshold level is updated in the filtering process. Examples of threshold-based filtering are summarized in Table 1.

Fixed filtering is based on static thresholds so it can be implemented locally by each node despite the actual readings available at other nodes. On the contrary, adaptive filtering depends on these readings to update the thresholds [7]. Filtering can be implemented straightforward by simply disseminating queries all over the network (i.e., flooding). Even if the cost of disseminating a query is high in terms of message exchange, the cost of collecting data is much lower as only a few nodes do respond. This approach guarantees that all relevant nodes are selected for reporting their readings. When monitoring fields, the same query needs to be processed periodically. To avoid flooding the network at each query, one solution is to use proactive schemes based on disseminating a query only once but letting nodes to report their readings if data have changed. However, this scheme is not suitable for large WSNs monitoring dynamic continuous fields, where much data changes are expected.

Table 1. Information obtained using threshold-based filtering techniques.

fixed filtering (static threshold)	adaptive filtering (dynamic threshold)
max (value > $X$ ), min (value < $X$ ), range ( $X_1 < \text{value} < X_2$ ), iso-contour (value = $X$ ), count (number of nodes with value > $X$ ), multi-dimensional / complex (value <sub>1</sub> > $X_1$ & value <sub>2</sub> < $X_2$ )	max, min, top-k, top-porcentual, snapshot, maximal range, multi-dimensional / complex, skyline

In this work, we consider a reactive scheme where only the first query is flooded to all nodes, but the following ones are directed, based on an iterative process, towards areas where relevant nodes are located. In this way, we aim at disseminating the same query periodically but at a minimum cost. Besides reducing the cost of data gathering through filtering, it is worth noting that query dissemination may require much message exchange. This work investigates how to decrease the cost of query dissemination by integrating *computational intelligence* (CI) techniques on nodes that can help to direct queries and avoid flooding the network. CI is a set of nature-inspired computational methodologies to address complex problems of the real world in which traditional methodologies are ineffective or infeasible. A detailed survey of CI paradigms and their potential applications in WSNs is given in [8], in which two bio-inspired paradigms for energy-aware routing are highlighted: *swarm intelligence* (SI) (such as ant colony optimization, pheromone-based strategies, etc.) and *reinforcement learning* (RL). Both paradigms have low computational and memory requirements, high flexibility, and reach near-optimal results. A detailed survey and future directions of SI-based routing protocols is treated in [9].

In this context, we propose PhINP, a pheromone-based in-network processing mechanism using broadcast communication model, which combines advantages of *in-network filtering*, *pheromone-based* and *reinforcement learning algorithms*. The proposal is able to implement both fixed and adaptive in-network filtering by means of a directed query dissemination strategy. For simplicity, we focus the discussion on finding maximum values, which is a particular case of adaptive filtering. Robustness and adaptability of the routing mechanism are emphasized due to the variability of conditions in WSNs (i.e failure of nodes, introduction of new nodes, packet loss, etc). In order to obtain a good performance of PhINP mechanism for different environments, the sink node, applying reinforcement learning and a configuration policy, adapts the behavior of the mechanism to the dynamic of the physical magnitude and/or network state.

Additionally, a motivation for this work was to develop cooperative schemes to decentralize the control of the sink node. Therefore, a sink node that requires information in a moment can become a sensor node in a later time (role changing). This scheme enables to extend the network lifetime due to nodes in the neighborhood of the sink have high energy requirements due to they cooperate by forwarding many responses from other nodes.

The main contribution of this work is to propose a simple, efficient and distributed routing mechanism for query dissemination and information gathering, which can reduce the exchange of messages in the network and increase the network lifetime, by allowing some minimal error.

The rest of the paper is organized as follows. Section 2 discusses both related work on pheromone-based routing and the main differences with our proposal. Section 3 introduces the PhINP mechanism and the network model, while query dissemination, data gathering, and pheromone-level update are described in Subsections 3.1, 3.2, and 3.3 respectively. The pheromone configuration policy is described in Subsection 3.4. Section 4 analyzes results obtained by simulation and discusses main trade-offs. Finally, Section 5 concludes the work.

## 2. Related Work

*Bio-inspired networking and communication protocols and algorithms* have emerged as efficient and robust solutions for routing problems in WSNs. They are based on biological systems, as result of millions of years of evolution, with characteristics such as adaptability to environmental conditions, resiliency to failures and damages, collaborative operation and self-organization. By looking the nature as a source of inspiration, we can deal with large scale networks in which *the absence of centralized control* and *unattended resolution of potential failures* are key factors [10], [11].

*Swarm intelligence* is a relatively novel field which focus on the collective behavior of multi-agent systems using decentralized control and self-organization. In this kind of distributed and collaborative intelligence, an individual member has not the ability to perform an action efficiently. However, each member, using a kind of indirect communication named *stigmergy*, contributes to perform a task in an efficient way. In this sense, pheromone-based



routing mechanisms mimic the cooperative foraging behavior of ant colonies using two phases to improve information discovery and gathering. In the exploration phase, ants lay down pheromone trails. Upon finding food, ants return to their colony reinforcing the trails with more pheromone. Other ants tend to follow these paths and to reinforce them releasing more pheromone. Over time, pheromone tends to evaporate erasing unused paths. In order to avoid *stagnation*, in which all ants tend to use the same trails, losing the ability to discover new information sources, a certain probability of exploration of new trails may be considered.

This pheromone-based approach has inspired several routing mechanisms, which can be classified by applying three criteria. 1) In terms of how the data sending process is started, three approaches can be defined, *proactive* [12], [13] (sources report values to sink after an event), *reactive* [14] (sources report values to sink on-demand based on queries), and *hybrid* (a combination of proactive and reactive). These approaches are shown in Figure 2. Reactive schemes typically have better scalability than proactive ones because they require less control overhead [15]. Therefore, they have a good performance in scenarios where data changes quickly. 2) In terms of the communication model, most works have considered unicast communication model but broadcast and hybrid models have been also proposed [14]. The first one does not take advantage of the broadcast nature of wireless communication (a transmitted packet is received by all neighbor nodes in the communication range). In addition, broadcast both enables multi-path routing and increases the robustness regarding packet loss and failed nodes, which are very common in WSNs. 3) Finally, in terms of the initiator of the process, three approaches have been proposed, source-initiated [12], [15], [16], [17], sink-initiated [18], and hybrid approaches.

This work proposes a reactive data-driven routing mechanism which takes advantage of local pheromone levels in sensor nodes to direct query messages to nodes with relevant information. Key features that differentiate the proposed mechanism from existing ones are: (i) both data exploration and query dissemination are integrated in one phase, (ii) the pheromone level is neither transported in the query message nor exchanged between neighbor nodes, this is a local parameter of each node, (iii) the pheromone level of a node is a simple value indicating the probability to continue relaying the query rather than using a pheromone matrix, (iv) most pheromone-based routing mechanisms are address-based, instead our proposal is data-based, which is more suitable for in-network processing, and (v) the list of visited nodes is not included in the query message, which is very common in pheromone-based algorithms.

From the point of view of the knowledge of a node, we have three information domains. In the first one, a node has a  $m$ -by- $m$  matrix, where  $m$  is the number of nodes in the network, with the probability (or cost) to send messages from each node to another. In this case, the node has a global view of the network. In the second one, a node have a  $1$ -by- $n$  vector, where  $n$  is the number of neighbor's nodes, with the probability to send messages to each neighbor. In the last one, a node has only a local parameter which represents the probability to send a message to all neighbors (i.e. broadcast). We focus on the last case, which is less memory demanding and energy-efficient, especially in dense WSNs. In the two first cases, sensor nodes incur in high energy consumption due to communication overhead for matrix / vector

updating.

In this work we increase the analysis of PhINP mechanism of a previous work [19] both analyzing the parameter setting to obtain good performance (i.e., good trade-off between error and cost) and comparing the performance of PhINP with Gossip and Flooding mechanisms, applying the same in-network filtering strategies in all cases.

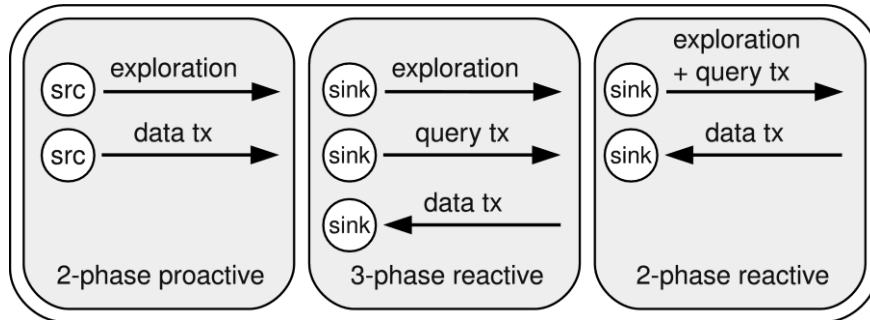


Figure 2. Classification of pheromone-based routing mechanisms in wireless sensor networks.

### 3. PhINP Mechanism

We consider a network conformed by sensor nodes disseminated over a square area, in which there is not a defined network topology. It is established in an ad-hoc manner based on local interactions (broadcast radio links), which are function of both node positions and communication range. In this context, one node takes the role of sink when it is accessed by a user requiring information. The surface, representing the physical magnitude to be sensed, changes continuously in time. Nodes are tasked with storing data obtained from sensing. Sink node, initiating a query process, has no clue about the information's location. PhINP mechanism makes use of two phases, *query dissemination*, and *data gathering*, followed by *pheromone-level update*, which are described in the following subsections. The *pheromone configuration policy* is described in subsection 3.4.

#### 3.1 Query Dissemination

This phase is based on iterative search space reduction, adapting the query dissemination to the data distribution. It efficiently combines both a *pheromone-based strategy*, to direct queries to nodes with relevant readings, and *in-network filtering*, to limit no relevant nodes to answer. The query message is conformed by type (query or data message), identification, and threshold value (e.g., maximum value). It is worth mentioning that, in order to implement the *maximum function*, nodes need to update the query threshold if their sensed value is larger (i.e., dynamic threshold). In order to reduce the communication cost, each node can relay the same query message only once.

On receiving a query message, each node must make two decisions, *a) if to answer the query, reporting its data to the sink after a period of time*, and *b) if to continue relaying the query to its neighbors*. The first decision is based on both the query's threshold value and the node's reading, and the second one is based on the node's pheromone level. These two

decisions generate four different kind of behaviors, which are detailed in Figure 3, in case of searching for maximum readings. First, the node checks whether it has already received the same query or not by reading the message identification (id). To limit overloading the network, a received query is discarded if it has already been processed. This lets each node to forward a query message at most once. Next, the node reads the threshold value of the query message to determine the relevance of its reading. The information gain  $\Delta E$  represents the difference between this threshold and the local sensed value. If  $\Delta E < 0$ , the reading is considered relevant and the node self-configures to answer the query after a period of time (Figure 3 a,c). Otherwise, the node has not relevant readings for the requirement (Figure 3 b,d). Finally, the node must decide if to relay the query to its neighbor nodes based on its pheromone level ( $\lambda$ ). As higher  $\lambda$ , more likely to relay the query. If the node's pheromone level is low, the probability to continue relaying the query is low (Figure 3 c,d), otherwise, the query is likely relayed (Figure 3 a,b), updating the threshold value of query message if there is information gain (Figure 3 a), and increasing the hop level of the query message. Algorithm 1 describe all the node operations in query dissemination phase, and Figure 3 shows the node behavior in opposite conditions ( $\lambda = 0.1$  and  $\lambda = 0.9$ ).

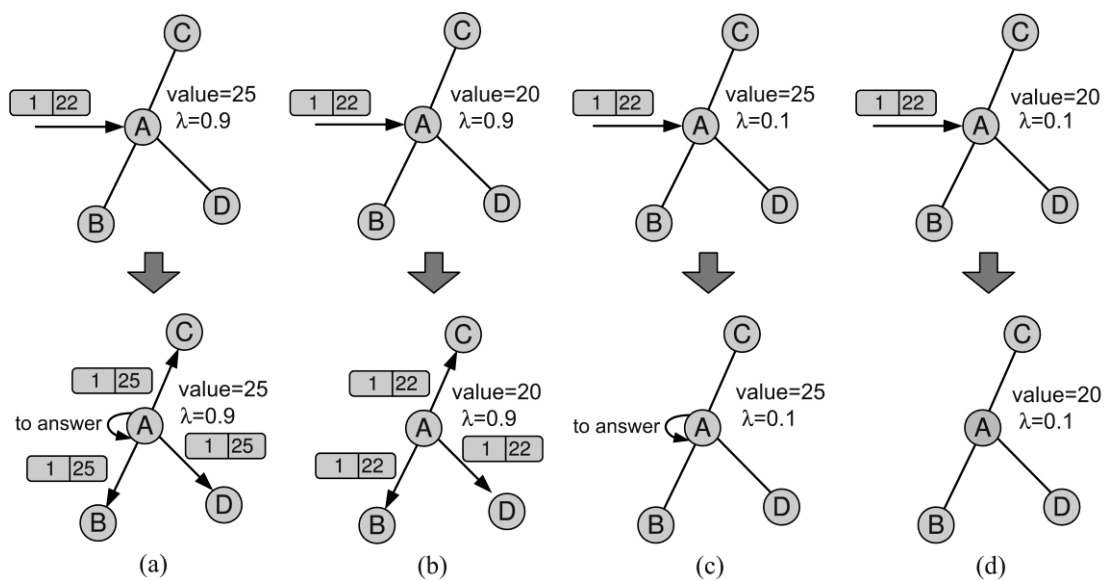


Figure 3. Query dissemination phase: (a) answer and relay query, (b) only relay, (c) only answer, and (d) no action.

### 3.2 Data Gathering

At the end of query dissemination phase, each node learns its shortest hop-distance to the sink node (i.e. *hop level*), from multiple copies of the same query message received by multiple paths. The response phase is triggered by all nodes selected after query dissemination phase. The triggering time is inversely proportional to the hop distance between sink and the selected node plus a random small delay. This mechanism allows to disable (i.e. filter) the answers of previously selected nodes near to the sink node. In data gathering phase, a multi-path routing strategy limited by hop level is applied to guarantee that



sensor readings arrive at the sink, providing low latency and robustness to node failures and unreliable links. After receiving a data message, each node, based on the hop level information, decides whether to relay the message to sink node, if is so, it puts its hop level in the message to relay. This is done when the hop level of the receiving node is less than that included in message by sending node. Otherwise, the message is discarded. This process is detailed in Figure 4 and Algorithm 2.

---

**Algorithm 1** - Query dissemination algorithm

---

```
1.  $id \leftarrow getId(queryMsg)$ 
2.  $hops \leftarrow getHops(queryMsg)$ 
3.  $thresholdValue \leftarrow getThreshold(queryMsg)$ 
4.  $\Delta E \leftarrow (thresholdValue - sensedValue)$ 
5.
6. // Discard already forwarded queries
7. if  $idTable[id] == true$  then
8.    $delete(queryMsg)$ 
9. else
10.   $updateIdTable(id)$ 
11. end if
12.
13. // Answer decision
14. if  $\Delta E \leq 0$  then
15.   $dataMsg \leftarrow createMsg(sensedValue)$ 
16.   $scheduleSend(dataMsg)$ 
17.   $answer \leftarrow true$ 
18. end if
19.
20. // Query relaying decision
21. if  $\lambda > rand()$  then
22.  if  $\Delta E \leq 0$  then
23.     $updateMsg(queryMsg, sensedValue)$  // Update threshold value
24.  end if
25.   $updateMsg(queryMsg, hopCount)$  // Update hop count
26.   $send(queryMsg)$ 
27. else
28.   $delete(queryMsg)$ 
29. end if
```

---

### 3.3 Pheromone-level Update

After data gathering phase, each sensor node updates its pheromone level, which changes its probability of forwarding new query messages. The update rate is setup by the query process and, in general, it is assumed equal to the monitoring frequency (i.e., the frequency at which queries are disseminated). The first query message encourages all nodes to participate in this phase by setting their maximum pheromone level ( $\lambda = 1$ ). This guarantees that all nodes receive the query (i.e. flooding) and that all nodes with relevant readings are properly selected to answer. After each iteration (i.e. query dissemination and data gathering phases), all nodes decrease their pheromone level by  $\lambda_{\text{dec}}$  up to a lower bound given by  $\lambda_{\text{min}}$ . Only nodes which have been either *selected to answer the query* (i.e., send sensed data) or *relayed data back to the sink*, increase their pheromone level by  $\lambda_{\text{inc}}$  up to an upper bound given by  $\lambda_{\text{max}}$  (such as in MAX-MIN Ant System [20]). In our proposal, the  $\lambda_{\text{max}}$  level is limited to 1, and the  $\lambda_{\text{min}}$  level can be adjusted based on the application requirements. In this last case, it is a trade-off between query cost and error, because for low  $\lambda_{\text{min}}$  values the query cost tends to be low but the error increases, and vice versa. The goal of this phase is to reinforce those paths directed towards nodes with relevant data, which is described in Algorithm 3. After a period of time, and in a synchronized and iterative way, all nodes are updated with a pheromone level of 1, and the process of space reduction search is initiated again. This mechanism avoids to lose new relevant readings from nodes with low pheromone level and distant from the formed trails.

---

**Algorithm 2** – Data gathering algorithm

---

```
1. hops  $\leftarrow$  getHops (dataMsg)
2. thresholdValue  $\leftarrow$  getThreshold (dataMsg)
3.  $\Delta E \leftarrow$  thresholdValue – sensed
4.
5. // Filter selected nodes to answer
6. if  $\Delta E > 0$  then
7.   unscheduleSend (dataMsg)
8.   answer  $\leftarrow$  false
9. end if
10.
11. // Discard already received data messages
12. if idTable [id] == true then
13.   delete (dataMsg)
14. else
15.   updateIdTable (id)
16. end if
17.
18. // Data message relay decision
19. if hops > minHops then
```

---

```

20.  Msg ← setHops (minHops - 1)
21.  send (dataMsg)
22.  relayDataMsg ← true
23.  else
24.    delete (dataMsg)
25.  end if

```

---

### 3.4 Pheromone Configuration Policy

In order to obtain a good performance of PhINP mechanism for different environments, the sink node, applying reinforcement learning and a pheromone configuration policy, adapts the behavior of the PhINP mechanism to the dynamic of the physical magnitude. That is, the sink node can change the time of convergence of PhINP. In this sense, if the sensor readings collected by sink node are the same after successive iterations, it realizes that the magnitude does not change, or that is very slow, and it applies a policy of configuration of the pheromone decrement and increment levels more conservative, and vice versa.

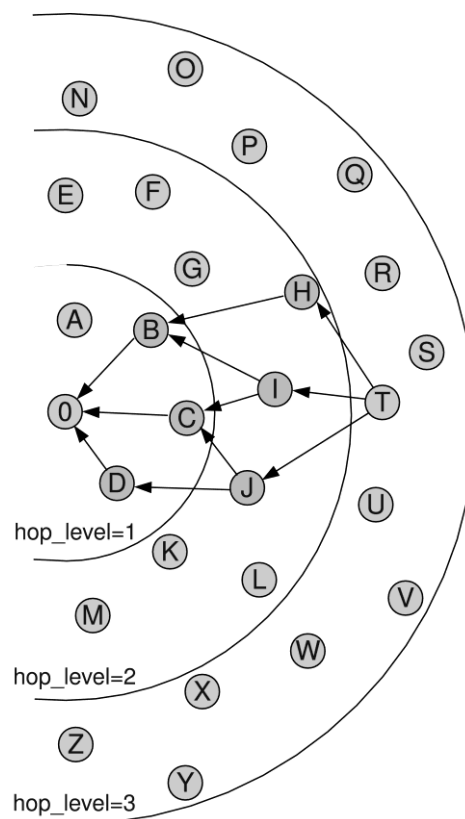


Figure 4. Data gathering phase: data messages are relayed to the sink based on the node's hop level

---

#### Algorithm 3 - Pheromone update algorithm

---

```

1. if  $\lambda > \lambda_{\min}$  then                                // pheromone evaporation

```

```
2.    $\lambda = \lambda - \lambda_{\text{dec}}$ 
3. end if
4.
5. if answer == true or relayDataMsg then // pheromone reinforcement
6.   if  $\lambda < \lambda_{\text{max}}$  then
7.      $\lambda = \lambda + \lambda_{\text{inc}}$ 
8.   end if
9. end if
```

---

#### 4. Performance Evaluation

PhINP mechanism was evaluated using a discrete event network simulator developed on Omnet++ [21]. *Query error*, *query cost* and *active nodes* metrics were considered in order to compare the performance of this proposal with respect to existing ones. The first metric represents the gap between the best reported data to the sink after a query iteration and the actual optimal one (i.e., the relative error). The second metric considers the number of nodes on average involved in forwarding the query message. Note that the maximum query cost is equal to 1, which means that all nodes relayed the query once (i.e., flooding). This cost is proportional to energy consumption, as power requirements in sensor networks are dominated by wireless communications. The latter metric depicts the proportion of active nodes in the network, that is, nodes with residual energy to keep running.

The performance of PhINP was compared with broadcast-based flooding and gossip mechanisms. However, the same in-network filtering strategies were applied in all cases. The main difference between them is how the query message is disseminated through the sensor network. *Flooding* is a deterministic approach, where the probability of each node to relay the query message is equal to one. As a consequence, queries are disseminated to all nodes. *Gossip* is a probabilistic approach, where the probability of each node to relay the query is based on the gossip level, which is fixed and equal in all nodes. This parameter can take values from 0 to 1. A low gossip level means that the query has low probability to be relayed for each node and, as a consequence, few nodes participate in the query dissemination, and vice versa. Gossip mechanism, setting a gossip level equal to one, behaves as flooding. Finally, *PhINP* is an adaptive approach, in which the probability to relay the query is based on node's pheromone level. This parameter is individually adapted between  $\lambda_{\text{min}}$  and  $\lambda_{\text{max}}$  in each iteration, based on data distribution, in order to obtain paths or trails between sink and sensor nodes with relevant information. These levels are limited considering error-cost requirements.

The simulation scenario is conformed by nodes uniformly deployed over a square surface with a constant node density of  $2.5 \times 10^{-3}$  nodes/m<sup>2</sup>, assuming a circular communication range of 40 meters. Therefore, if the network size is increased, the coverage area is extended, maintaining the same density. For example, we consider a deployment of 25 sensor nodes in a surface of 100 x 100 m<sup>2</sup>. The sensed field is modeled as the sum of  $S$  decreasing exponential functions with different values of position, amplitude, and attenuation. Sensor readings depend on node positions in the field. The field model assumes multiple independent sources

such as air conditioners, lights, speakers, etc. Our simulations consider the case of  $S = 160$  and report the average results between 2000 and 5000 random simulations. In this sense, the higher the number of nodes in the network, higher resolution.

In the followings subsections, we analyze the convergence and sensitivity of PhINP, the performance compared with flooding and gossip mechanisms, and the robustness of PhINP respect to packet loss and failed nodes obtained through simulations.

#### *4.1 Convergence and Sensitivity*

Since the proposed scheme first floods the whole network, the error remains null in the following iterations if there are no changes in the sensed field. In fact, the error remains near to 0 due to a low probability of disconnected nodes in some deployments. Under these conditions, the convergence of the algorithm can be analyzed. At each iteration, paths to relevant data are reinforced, decreasing the query cost. This behavior is shown in Figure 5 for different network sizes. Note that the cost is almost independent of the network size. Figure 5 shows that, for a low  $\lambda_{\min}$  level, the algorithm convergence is slower (12 iterations instead of 6) than for the high ones, but with a lower query cost (40% instead of 65%). As a consequence, there is a trade-off between convergence time and minimum cost.

Since monitoring implies to be able to track changes in the sensed field, we evaluate the sensitivity and the dynamics of the proposed scheme to these changes. Ideally, once the scheme converges to some minimum query cost after a few iterations, it should be able to still detect new relevant data sources with a low error probability. The sensitivity to field changes depends on the minimum level of pheromone  $\lambda_{\min}$  that nodes can have. We evaluate this sensitivity for  $\lambda_{\min} = 0.1$ , considering a network with 400 nodes. To this end, after convergence (12 iterations), the amplitude of each data source is changed randomly up to 90%. Figure 6 shows, that after changing the amplitude of data sources, a peak in the error occurs, which is minimized by network adaptation in the following iterations. The larger the  $\lambda_{\text{inc}} / \lambda_{\text{dec}}$  relationship, the faster is the convergence to the minimum error; however, the cost tends to increase in the following iterations. As a conclusion, a  $\lambda_{\text{inc}} / \lambda_{\text{dec}}$  relationship of 2 is adequate for continuous fields with slow spatio-temporal variation as the simulated one, however, for fields with high variation, a larger value is more convenient.

From simulations, we conclude that, for the simulated environment, a good error-cost trade-off can be obtained using  $\lambda_{\text{dec}} = 0.1$ ,  $\lambda_{\text{inc}} = 0.2$  or  $0.4$  (according to the temporal field dynamic),  $\lambda_{\min} = 0.1$  and  $\lambda_{\max} = 1$ .

#### *4.2 Performance*

The performance of PhINP mechanism is compared with broadcast-based flooding and gossip query dissemination schemes for a network size of 400 nodes, which is shown in Figure 7. This comparison considers only query disseminating cost (i.e., query cost) as the cost of replies is negligible with respect to the query one. Flooding disseminates the query to all nodes, while gossip to only half of nodes (setting a gossip level = 0.5), which are randomly selected as the query is propagated through the network. PhINP is configured using  $\lambda_{\text{dec}} = 0.1$ ,  $\lambda_{\text{inc}} = 0.2$ ,  $\lambda_{\min} = 0.1$  and  $\lambda_{\max} = 1$ . Our proposal performs like flooding in terms of



errors but shows a cost reduction of 60%. As expected, gossip is able to limit the query cost but with a high error. These results validate that our proposal is able to search and learn where relevant data are located and reduce the cost of disseminating the query to these areas.

Finally, network lifetime, analyzing the percentage of active sensor nodes, and search error are analyzed for gossip, flooding and PhINP mechanisms. We suppose that each sensor node is powered by two AA alkaline batteries, with 9360 Joules each one. A standard IEEE 802.15.4 radio transceiver at 0 dBm power output approximately requires 0.12 mJ for the transmission or reception of a byte [22]. A packet size of 20 kB is defined, which enables the transmission or reception of 7800 packets per node, considering negligible the energy for sensing and computing. After that, a sensor node fails generating holes in the network. Additionally, we consider that the sink node has unlimited energy, because it could be connected to the power line. Figure 8 shows the query cost and the percentage of active nodes in the network as a function of the query iteration.

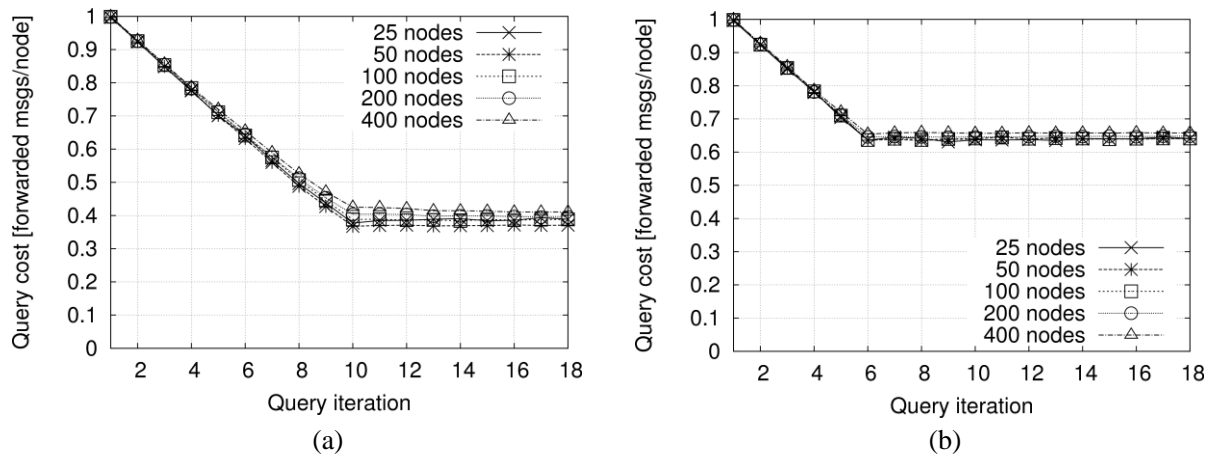


Figure 5. Convergence and sensitivity: query cost for (a)  $\lambda_{\min} = 0.1$  and (b)  $\lambda_{\min} = 0.5$ .

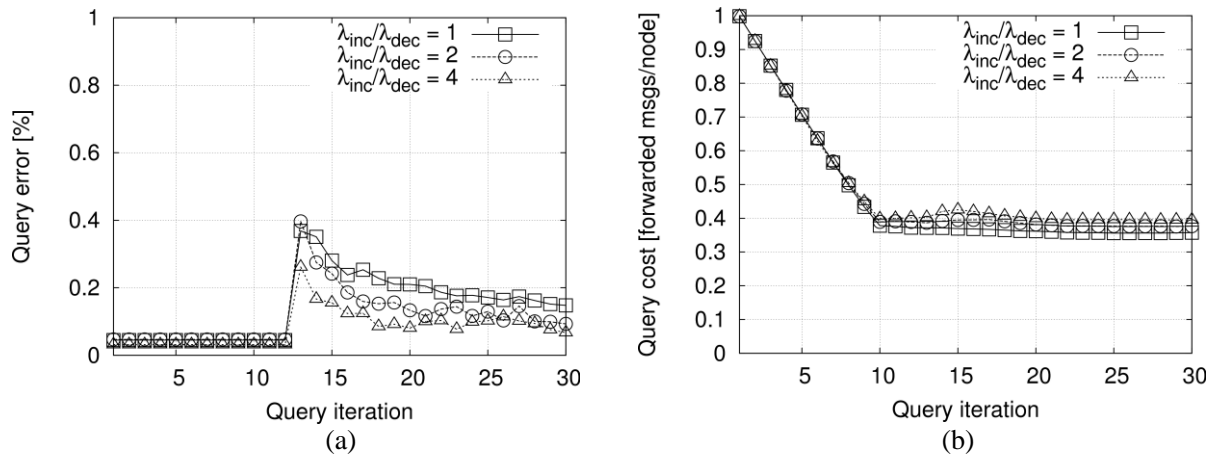


Figure 6. Convergence and sensitivity: query error (a) and cost (b) for different  $\lambda_{\text{inc}} / \lambda_{\text{dec}}$  relationships, applying amplitude changing of data sources after algorithm adaptation.

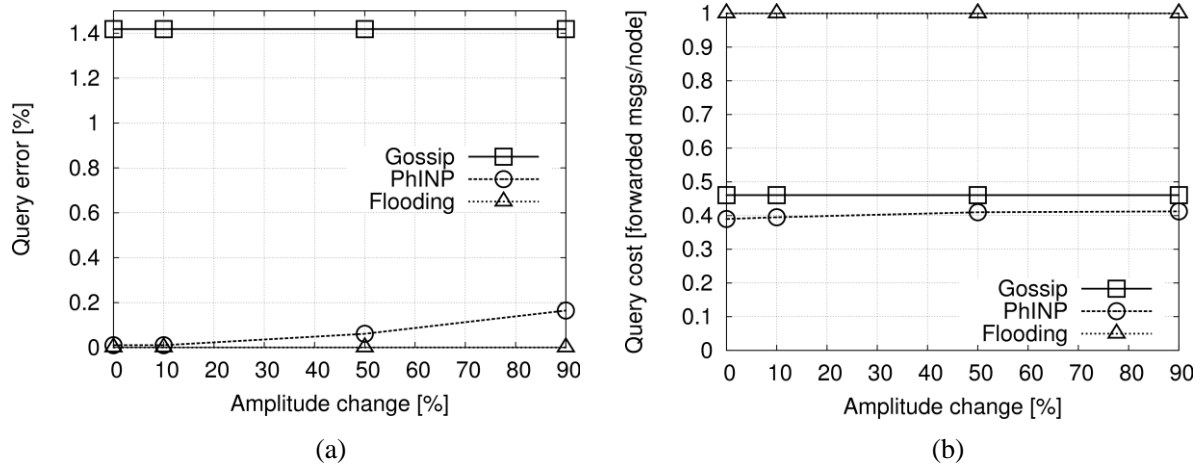


Figure 7. Performance comparison respect to amplitude change

We can note that PhINP mechanism has a similar performance that flooding respect to query error and cost. From 300 iterations, flooding suffers from a fast decrement of the percentage of active nodes, with a convergence to 60% of active nodes. Gossip and PhINP schemes, instead, outperform the percentage of active nodes with respect to flooding from 400 iterations approximately, increasing the network lifetime. Analyzing the temporal energy levels in all sensor nodes, we note that nodes at 1-hop of the sink have high energy consumption, due to the high message exchange. In order to overcome this limitation, a strategy of sink node selection given a period of time can be applied (i.e. role changing).

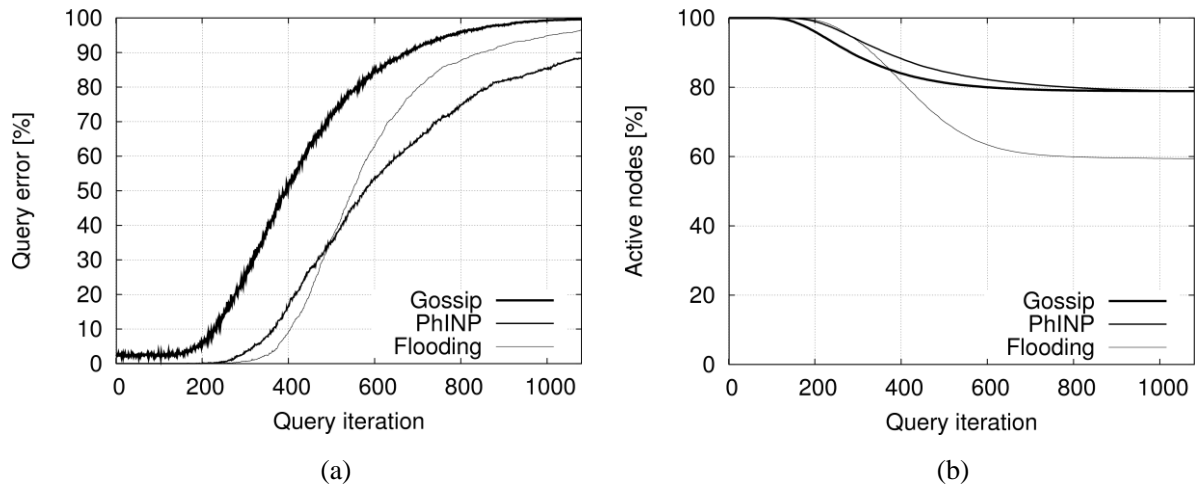


Figure 8. Performance comparison of routing mechanisms based on query iteration.

A view of the query dissemination space in a deployment of 400 nodes is shown in Figure 9. In this case, all the wireless interconnections between nodes are indicated by gray lines. The sink node is located at the center, and the black lines indicate the links used in the query dissemination. For the case of gossip mechanism (Figure 9 a), a gossip level = 0.5 is defined. Figure 9 b shows the query cost of PhINP after 12 iterations, for  $\lambda_{dec} = 0.1$ ,  $\lambda_{inc} = 0.2$ ,  $\lambda_{min} = 0.1$  and  $\lambda_{max} = 1$ . The big and green nodes represent nodes configured to answer due to they have relevant data.

### 4.3 Robustness

Finally, the robustness of PhINP to network changes is analyzed. To this end, we evaluate its performance under packet loss conditions. Figure 10 shows the query error and cost as a function of both the network size and packet loss probability using  $\lambda_{\text{dec}} = 0.1$ ,  $\lambda_{\text{inc}} = 0.2$ ,  $\lambda_{\text{min}} = 0.1$  and  $\lambda_{\text{max}} = 1$ . Probabilities up to 10% of packet loss keep the error below 5% for networks of up to 400 nodes. Under different packet loss conditions, the cost remains almost the same. The robustness of PhINP respects to failed nodes and query iteration is shown in Figure 8.

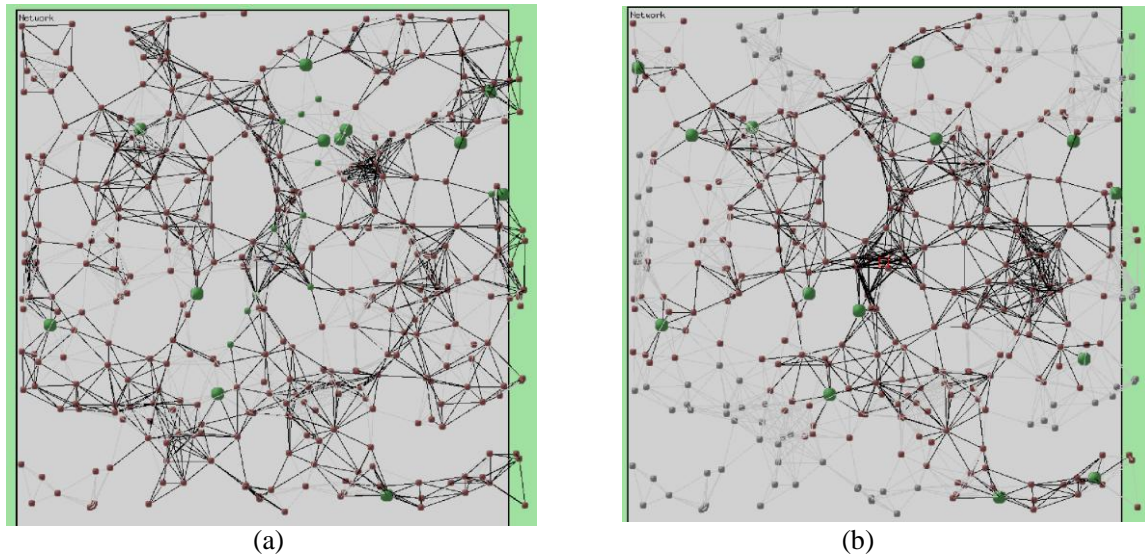


Figure 9. Performance comparison: (a) Gossip algorithm, and (b) PhINP algorithm after adaptation

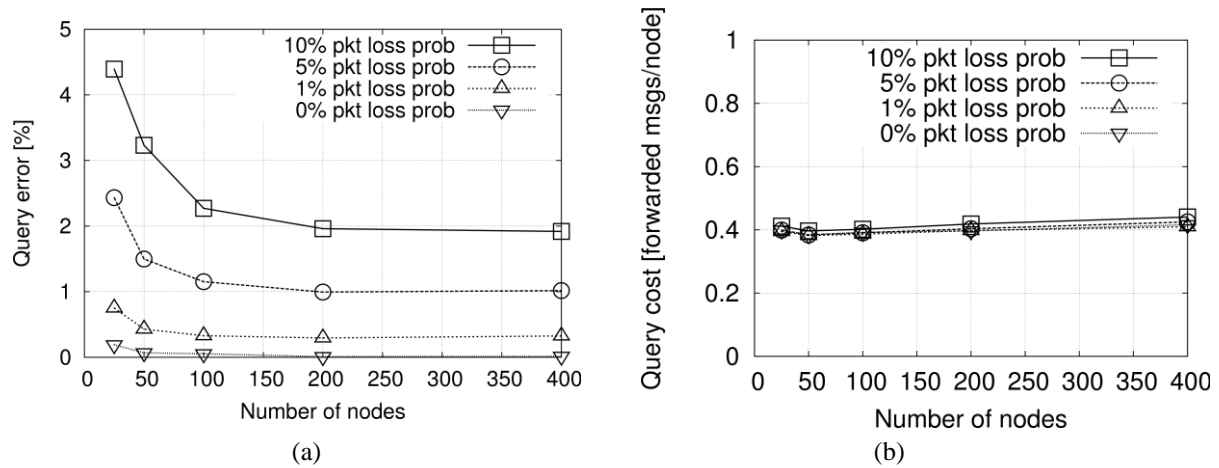


Figure 10. Robustness: Query error (a) and cost (b) for different packet loss probabilities.

## 5. Conclusion

In this work, we propose a simple and efficient mechanism for monitoring spatio-temporal continuous fields using wireless sensor networks. The scheme tackles the query dissemination problem applying in-network filtering, and reinforcing distribution paths using an iterative pheromone-based process. It is able to reduce the messages exchanged in the

network, by allowing some minimal error, in order to maximize the network lifetime. Results show that the query cost can be significantly reduced while keeping errors very low, outperforming flooding and gossip mechanisms. Ongoing works includes the implementation of dynamic increment and decrement pheromone values, based on the data distribution, in order to obtain a better distribution of energy consumption in the network, and the implementation of PhINP mechanism in real wireless sensor networks.

## Acknowledgement

This work is partially funded by the FONCyT - National Technological University IP-PRH 2007 Postgraduate Grant Program and by the SECyT - National University of Cordoba 2010-2011 Research Program.

## References

- [1] Yick J., Mukherjee B., and Ghosal D., "Wireless Sensor Network Survey", Computer Networks, Vol. 52, No. 12, pp. 2292-2330, 2008.  
<http://dx.doi.org/10.1016/j.comnet.2008.04.002>
- [2] Mulligan R., and Ammari H., "Coverage in Wireless Sensor Networks: A Survey", Journal of Network Protocols and Algorithms, Macrothink Institute, ISSN 1943-3581, Vol. 2, No. 2, 2010. <http://dx.doi.org/10.5296/npa.v2i2.276>
- [3] Anastasi G., Conti M., Di Francesco M., and Passarella A., "Energy Conservation in Wireless Sensor Networks: A Survey", Ad Hoc Networks, Vol. 7, No. 3, pp. 537-568, 2009. <http://dx.doi.org/10.1016/j.adhoc.2008.06.003>
- [4] Pal A., "Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges", Journal of Network Protocols and Algorithms, Macrothink Institute, ISSN 1943-3581, Vol. 2, No. 1, 2010. <http://dx.doi.org/10.5296/npa.v2i1.279>
- [5] Park B., Park S., Lee E., Noh S., and Kim S., "Large-scale Phenomena Monitoring Scheme in Wireless Sensor Networks", IEEE Vehicular Technology Conference (VTC 2010-Spring), pp. 1-5, 2010. <http://dx.doi.org/10.1109/VETECS.2010.5494214>
- [6] Li M., and Liu Y., "Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks", In IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No. 5, pp. 699-710, 2010. <http://dx.doi.org/10.1109/ICDCS.2007.115>
- [7] Sun J., "Multi-Threshold Based Data Gathering Algorithms for Wireless Sensor Networks", Journal of Networks, Vol. 4, No. 1, pp. 30-41, 2009
- [8] Kulkarni R. V., Förster A., and Venayagamoorthy G. K., "Computational Intelligence in Wireless Sensor Networks: A Survey", IEEE Comm. Surveys & Tutorials, vol. 13, No 1, 2011. <http://dx.doi.org/10.1109/SURV.2011.040310.00002>
- [9] Saleem M., Di Caro G., and Farooq M., "Swarm Intelligence-based Routing Protocol for Wireless Sensor Networks: Survey and Future Directions", Inform. Sci. (2010), 2010 <http://dx.doi.org/10.1016/j.ins.2010.07.005>
- [10] Sumpter D. J. T., "The Principles of Collective Animal Behavior", Philosophical Transactions of the Royal Society B 361: 5-22, 2005.

- <http://dx.doi.org/10.1098/rstb.2005.1733>
- [11] Dressler F., and Akan O., “A Survey on Bio-Inspired Networking”, Elsevier Computer Networks, Vol. 54 (6), pp. 881-900, April 2010.  
<http://dx.doi.org/10.1016/j.comnet.2009.10.024>
- [12] Camilo T., Carreto C., Silva J., and Boavida F., “An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks”, In Proceedings ANTS, vol. 4150 of Lecture Notes in Computer Science (LNCS), Springer, pp. 49-59, Brussels, Belgium, 2006.  
[http://dx.doi.org/10.1007/11839088\\_5](http://dx.doi.org/10.1007/11839088_5)
- [13] Okdem S., and Karaboga D., “Routing in Wireless Sensor Networks Using an Ant Colony Optimization (ACO) Router Chip”, Sensors 2009, 9, 909-921, 2009.  
<http://dx.doi.org/10.3390/s90200909>
- [14] Ducatelle F., Di Caro G., and Gambardella L., “Using Ant Agents to Combine Reactive and Proactive Strategies for Routing in Mobile Ad Hoc Networks”, International Journal of Computational Intelligence and Applications (IJCIA2005), pp. 169-184, 2005  
<http://dx.doi.org/10.1142/S1469026805001556>
- [15] Brandl M., Posniecek T., Kellner KH., Kos A., Mayerhofer C., and Fabian C., “A Simple Pheromone based On-Demand Routing Protocol for Wireless Sensor Networks”, Proceed. Fourth Int. Conf. Syst. Netw. Commun. (ICSNC2009), pp. 254-260, 2009  
<http://dx.doi.org/10.1109/ICSNC.2009.114>
- [16] Vieira L., Lee U., and Gerla M., “Phero-Trail: Bio-inspired Routing in Underwater Sensor Networks”, IEEE Journal on Selected Areas in Communications - JSAC, Vol. 28, No. 4, pp. 553-563, 2010. <http://dx.doi.org/10.1109/JSAC.2010.100505>
- [17] Hui X., Zhi-gang Z., and Feng N., “A Novel Routing Protocol in Wireless Sensor Networks based on Ant Colony Optimization”, IEEE International Conference on Environmental Science and Information Application Technology (ESIAT2009), Vol. 2, pp. 646-649, 2009. <http://dx.doi.org/10.1109/ESIAT.2009.460>
- [18] Michlmayr E., Pany A., and Kappel G., “Using Taxonomies for Content-based Routing with Ants”, International Journal of Computer and Telecommunications Networking, Vol. 51, Iss. 16, pp. 4514-4528, 2007. <http://dx.doi.org/10.1016/j.comnet.2007.06.015>
- [19] Riva G., and Finochietto J., “Pheromone-based In-Network Processing for Wireless Sensor Network Monitoring Systems”, IEEE International Conference on Communications (ICC2012), Ottawa, Canada, 2012
- [20] Stützle T., and Hoos H., “Max-Min Ant System”, Future Generation Computing Systems, 16:889-914, 2000
- [21] Omnet++ simulation library. Available at: <http://www.omnetpp.org>
- [22] Amiri M., “Evaluation of Lifetime Bounds of Wireless Sensor Networks”, Cornell University Library, 2010. <http://arxiv.org/pdf/1011.2103>

### Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).