

Context Aware Wireless Sensor Network Suitable for Precision Agriculture

Nour Brinis, Leila Azouz Saidane

National School of Computer Science, University of Manouba, Manouba, Tunisia
Email: nour.brinis@ensi-uma.tn

Received 21 November 2015; accepted 8 January 2016; published 11 January 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Using the Wireless Sensor Networks WSNs in a wide variety of applications is currently considered one of the most challenging solutions. For instance, this technology has evolved the agriculture field, with the precision agriculture challenge. In fact, the cost of sensors and communication infrastructure continuously trend down as long as the technological advances. So, more growers dare to implement WSN for their crops. This technology has drawn substantial interests by improving agriculture productivity. The idea consists of deploying a number of sensors in a given agricultural parcel in order to monitor the land and crop conditions. These readings help the farmer to make the right inputs at the right moment. In this paper, we propose a complete solution for gathering different type of data from variable fields of a large agricultural parcel. In fact, with the in-field variability, adopting a unique data gathering solution for all kinds of fields reveals an inconvenient approach. Besides, as a fault-tolerant application, precision agriculture does not require a high precision value of sensed data. So, our approach deals with a context aware data gathering strategy. In other words, depending on a defined context for the monitored field, the data collector will decide the data gathering strategy to follow. We prove that this approach improves considerably the lifetime of the application.

Keywords

Wireless Sensor Network, Precision Agriculture, Data Collector, Context Aware

1. Introduction

Water consumption in agriculture is the most part of water resources consumption in domestic, industrial and agricultural purposes. So, valuation of agricultural water productivity is important for improving the agricultural water resources utilization, especially in the arid areas [1]. For this reason, the science of precision agriculture

has been involved, having the purpose to enhance efficiency in crop farming and water resources. This science consists of precise understanding, estimating and evaluating crops condition with the aim of different parameter readings. WSN (Wireless Sensor Networks) is one of the promising technologies for monitoring crops and doing the suitable readings. A number of studies have been published in the last few years on this theme [2]-[5]. We distinguish between two kinds of applications: reactive WSN applications and data gathering applications. Using a reactive WSN, even the person in the field is replaced by an automatic actuator to do some critical tasks like irrigation and fertilization when sensed parameters achieve a given value; this is called the Wireless Sensor Actuator Network WSAN. In the other case, the actuation task is not considered. So, the WSN has the limited purpose of monitoring the field. In case of reactive WSN, the precision farming system has the following parts:

- 1) Identification of the sensing agricultural parameters.
- 2) Identification of the sensing location and deployment of the sensor network.
- 3) Transferring data from crop field to control station for decision making.
- 4) Actuation and Control decision based on sensed data.

In the second kind of applications, the actuation part is not considered. These parts are detailed in the following paragraphs.

2. A WSN-Based Precision Farming System

2.1. Identification of the Sensing Agricultural Parameters

This task is performed by the agricultural engineer. It consists of choosing the agricultural parameters to sense in order to achieve the target. Some of the common and most critical measured factors in agricultural WSN are air temperature (T) and relative humidity (RH), soil moisture, salinity, etc. [6] [7]. Of course the decision of what factors to monitor depends on the crop type. In addition to these, there are other factors to measure such as leaf thickness and atmospheric pressure, which depends on the crop being monitored. The option of the sensor types such as Sensirion SHT75 or SHT71 for RH and T depends on factors related to sensor accuracy, resolution, range, power consumption, precision, cost, etc. **Figure 1** present different sensors used for agriculture field. We notice that some platforms like Mica2 support many kind of sensor boards in order to do different readings at the same time [8].

2.2. Sensor Network Deployment

Every deployment has its own needs imposed by the type of the monitored crop or plant or by other special application. The coverage of the sensor nodes in agricultural WSN must be dense, *i.e.* one sensor node every one square meter (1 m^2) [9] [10]. Dense deployment serves to capture all the necessary measurements in order to have complete and reliable knowledge of the monitored area. For instance, crops have always different climate, known as microclimate. The micrometeorological parameters outside the field are monitored with the installation of weather stations. Besides the aforementioned issues, precision agriculture has maintenance-safety issues. In fact, maintenance of the WSN system must be considered, because of the long-term nature of the deployment. The sensor nodes must be put into protective cases preventing them from moisture, mud, etc. These cases have ratings in form of IP00, which means no protection. The first digit means protection against solid objects, while the second one means protection against liquids and every level has its definition. For example, in IP67, the 6 digit means total protection against dust and the 7 digit means protection against the effects of temporal immersion till 1 m underwater [11].



Figure 1. Agricultural sensors.

2.3. Transferring Data from Crop Field to Control Station

Many agricultural deployment strategies assume the sensor nodes sensing and sending data to be time-driven, such as in [12]-[14], in order to acquire a complete picture of the crop circumstances and act accordingly. There are other sensing strategies like event-driven, which are mostly used when monitoring phenomena such as volcanoes, earthquakes, etc. In these cases the monitored subject is the event itself. Finally, the sensing task can be done on-demand based on the user/application requirements. The sensing intervals vary from one minute to one hour.

2.4. Actuation and Control Decision

Actuators like sprinklers, foggers, and valve controlled irrigation system are available and are controlled by the base station.

As a delay tolerant application, precision agriculture may rely on a data gathering strategy better than a tree-routing technique. In particular, we are interested in designing data gathering protocols in WSNs to increase the lifetime of the system, making it more energy efficient. Many data gathering schemes have been proposed for wireless sensor networks. Our data gathering strategy inspires from the strategies presented in the following paragraphs.

3. Data Gathering Schemes

3.1. EDGM: A Single Hop Gathering Strategy Using Data Mules

The goal of this strategy [14] is to determine the optimal number of data mules required to optimize the covered network lifetime while meeting a given data Delivery Delay Deadline DDD . The goal of each sensor node is restricted to capture data and then transmit them to a one-hop node: directly to the sink, if this node is located in its transmission range, or to a Data Mule DM , when it is one hop away. In other words, sensor nodes do not relay the traffic generated by other nodes. At the proximity of each static node, a DM stops at a Break Point, denoted $BPoint$, for a defined Break Period, denoted $BPeriod$, and sends a $HELLO$ message. When it arrives at the proximity of the sink and the energy terminal, a DM sends the collected data to the sink. Each sensor node SN is associated to a unique $BPoint$. The break point positions are defined as the tops of equilateral triangles of edge length $\sqrt{3}rc$ with rc is the communication range of sensor nodes [15]. Each SN remains awake only during its break period to send its captured data. Then, it turns off its communication unit. During the break period, since several sensor nodes are authorized to send data to the DM , collisions may occur. A deterministic medium access scheduling is set by the DM , in order to avoid the collision problem. More precisely, each node starts sending its sensed data when it receives an invitation delivered by a DM through an $INVIT$ message which contains the medium access scheduling. During the first tour, each SN receives the information needed to synchronize itself with the DM s present in the considered area. Once it is synchronized, it keeps awake only during a DM break period in its associated $BPoint$. During this time, it is authorized to send its collected data through an $INVIT$ message. **Figure 2** presents the data collector trajectory and break points.

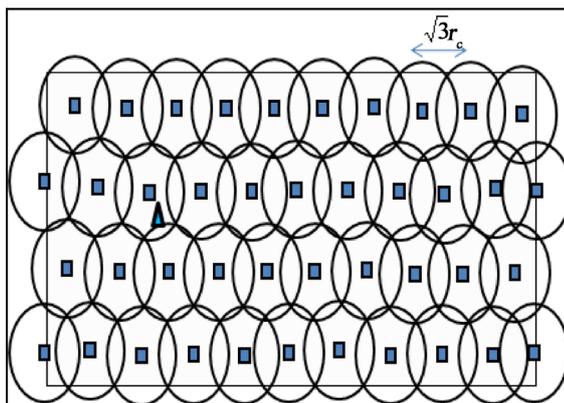


Figure 2. EDGM strategy.

3.2. ADA-DG: Adaptative Delay Aware Data Gathering Strategy

The key idea of ADA-DG [16] consists of finding the suitable trajectory of the Data Collector DC for which we guarantee sending data to the sink before a required data delivery delay deadline. For this purpose, each static sensor node sends their collected data to the DC with incorporating the tolerated data delivery delay deadline DDD. Knowing the DDD, the DC selects the suitable trajectory that allows carrying data to the sink at the appropriate time while minimizing as much as possible the static nodes energy consumption. The suitable trajectory is selected according to a suitable selection of a relay hops bound, denoted H_{max} . Therefore, a subset of the whole break points is selected and it is colored Red, denoted RBP s. Only the RBP s must be visited by the DC. The remaining nodes attached to the remaining break points, colored green, will be affiliated to one of the RBP s. So, ADA-DG strategy aims at coloring the sensor network area into red-green disks. This problem is resolved by two steps. The first step consists of determining the red breakpoints, at which the DC stops to gather data. This sub-problem is converted into a logical satisfiability problem. The propositional variables are the set of breakpoints $\{BP_1, BP_2, \dots, BP_n\}$ where BP_i is set *true* means that the corresponding breakpoint is red. Knowing, the network graph, we set n logical clauses corresponding to the possibilities of each breakpoint to be colored red. For instance, let C_i be the clause corresponding to the breakpoint BP_i :

C_i : BP_j or (BP_k and BP_l)

this constraint means that:

“ BP_i is Green” \rightarrow “ BP_j is red” or “ BP_k and BP_l are red”.

Our goal consists of satisfying all the clauses C_i . By resolving this problem, the different truth statement defines the different possibilities to construct red-green network. The goal of ADA-DG strategy is to find the trajectory visiting the minimum number of red break points; this means that the required solution is that having the minimum number of truth values in their truth statement. We may have many solutions. So, the second criterion for selecting the RBP consists of minimizing the cost of the selected trajectory, in terms of number of relay hops. Therefore, [16] computes this cost for each truth solution. Then, the word offering the minimal cost is chosen to select the RBP s.

At the second step, the DC computes the shortest possible route that visits each RBP exactly once and turns back to the RBP containing the sink. This route is deduced using an heuristic for the Traveling Salesman Problem TSP.

Figure 3 presents an example of sensor network coloring using the ADA-DG strategy.

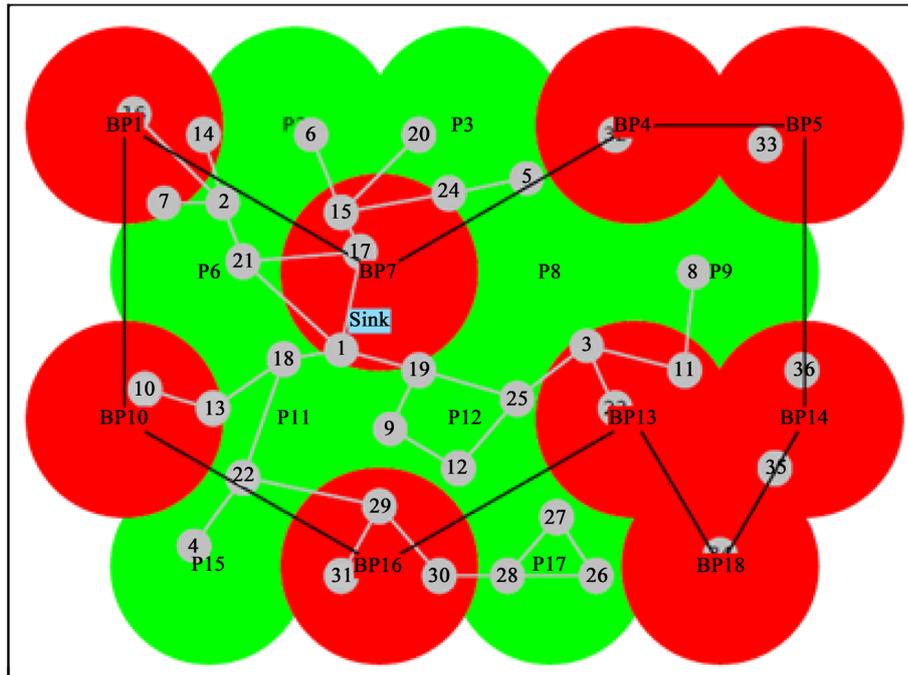


Figure 3. ADA-DG strategy.

Our contribution in this paper consists of including context awareness in data gathering schemes for considering the in-field variability agricultural parcel. In fact, context awareness would incorporate a degree of intelligence in the sensor nodes so that they can decide the action to do, depending on a local context. Since we are dealing with agriculture, two types of contexts may be considered: the first one consists of the sensed data value such as temperature or humidity (current context) and the second one is an implicit input defined for different land conditions such as crop types, location or time of the day.

4. Context-Aware Data Gathering Strategy

In this paper, we deal about the topology illustrated in **Figure 4**. An agricultural parcel is divided into a number of fields, each of them are planted with a type of crop. We suppose that the data collector makes a smart decision about its trajectory depending on each field context. This idea tends to get over supporting a fixed kind of sensor topology for variable fields. In the two following section, we propose two data gathering approaches. Each of them uses a context definition in order to decide the data collector strategy.

4.1. Data Gathering Scheme: Based on ADA-DG Strategy

We now explain how we incorporate context awareness in choosing the data gathering strategy. Our approach is based on the *ADA-DG* strategy. Besides, we suppose that a Field Master *FM* is selected in each field. His role consists of saving the last context of the field and sending it to the Parcel Master *PM* using *WIFI* technology. The field master saves the worst value of captured data, adds the date of capture and sends them to the *PM*. We also note that the data gathering may be done by a data collector robot or any other instrument that visits the field to make insecticide, to harvest crops or any other reason. The decision of the strategy is made only by the *PM* after collecting the n last contexts. The *PM* will visit k fields ($k \leq n$) depending on the time of the last data gathering of each field, the last context and the required delay. Concerning the context model, we inspire from the pyramid model proposed in [16]. The agriculture information space is represented as a concept pyramid with the vertices representing the various contexts of information retrieval (temperature, humidity, land type, crop type). In our methodology, we consider a space level query for the whole agricultural parcel (crop, land type, temperature) and an edge level query for each field (land type, temperature). The *FM* saves the worst context which is defined as the worst values of sensed data (temperature). We denote DDD_j the required Data Delivery Delay of field j from the k fields to visit. The fields are organized in croissant order of the DDD . These fields will be visited by the *DC* under control of the *PM*. The control is done by the *PM* depending on the last received context of each field and their DDD . We apply the *ADA-DG* strategy in each field. In other words, the *DC* goal consists of applying the pre-explained *ADA-DG* strategy for each part of the field. The DDD of each part is decided by the *PM* based on the last context of each part. **Algorithm 1** resumes the behavior of the *DC* when receiving the decision from the base station. **Figure 5** illustrates an example of this data gathering scheme application.

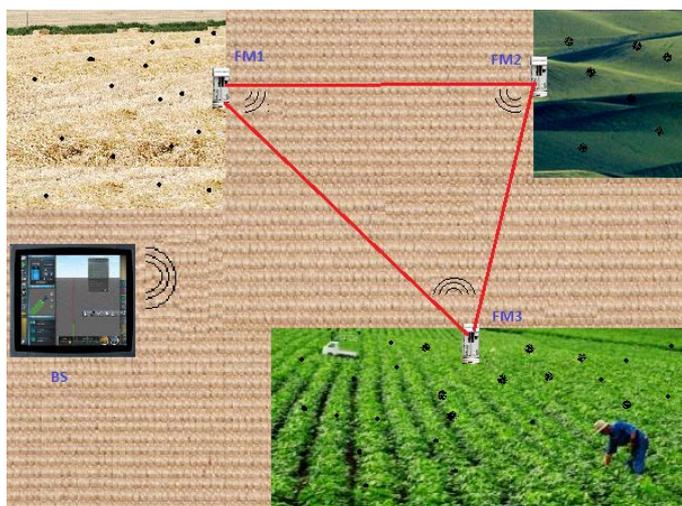


Figure 4. Parcel topology.

<p>Algorithm 1: DC moving path INPUT: the network graph OUTPUT: the DC moving path</p>
<p>1. j: the counter of fields 2. When arriving to a <i>FM</i>, update <i>D</i>. D is the time elapsed from the <i>PM</i> decision make until reaching the concerning <i>FM</i> 3. $DDD_j = DDD_j - D$ 4. ADA-DG(DDD_j) 5. k is the counter of RBP 6. for each BP_k in RBP do 7. collect data from the associated nodes 8. end for 9. move to the next <i>FM</i></p>

Algorithm 1. DC moving path.

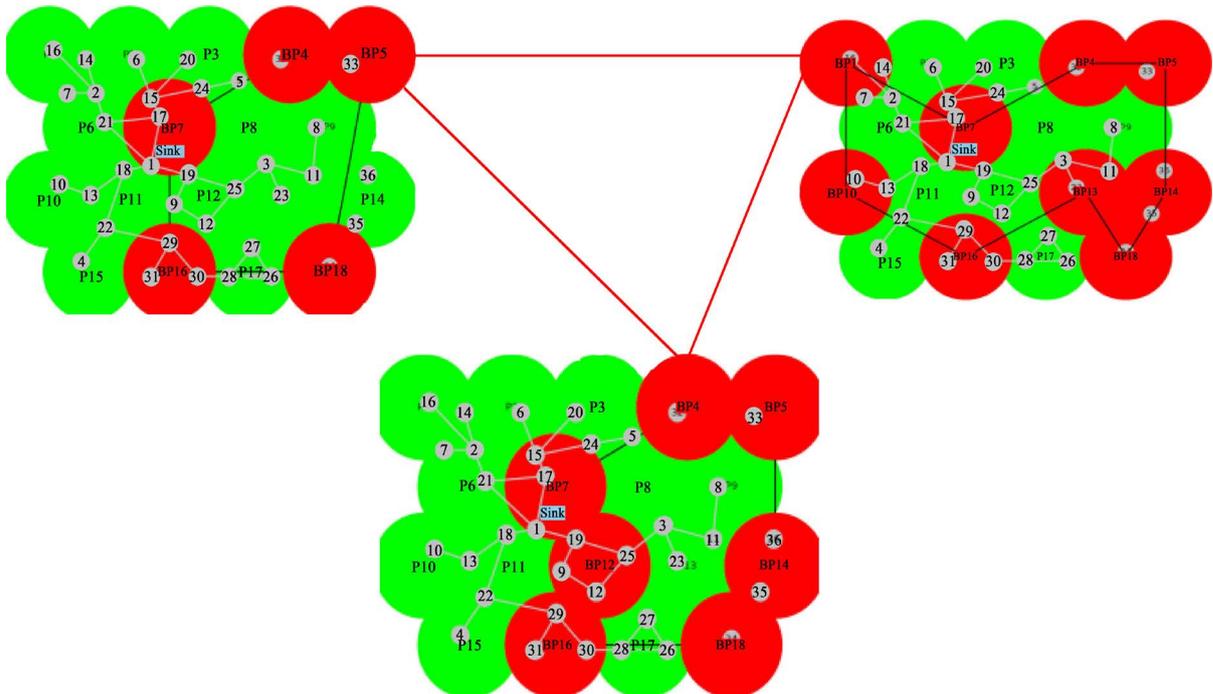


Figure 5. Example: data gathering based on AD-DG strategy.

4.2. Data Gathering Scheme: Based on Islands Creation

In the first approach, we suppose that all the sensors deployed in a field send their data separately to the data collector. However, as a fault-tolerant application, precision agriculture does not require a high precision value of sensed data. So, in order to improve the lifetime of the application, we propose to group the sensor nodes into islands with near sensed values. For instance, in a given land, we may have some permanent variabilities, for example, a hill or plain space, a shadow or light area for a given time-of the day... So, having the same permanent condition of a given part of the field may imply finding very near values of sensed data. Then, sensing data from only one node of a part of the field, having near conditions, may satisfy the precision agriculture application. For this purpose, we propose in this section to group the nodes into islands having near sensed values. In the following paragraph, we define the islands definition. Once the islands are defined, alternatively each node is visited in the same time-of the day. The other nodes sleep in order to preserve their energy. We notice that a trigger of new island creation is defined. In our case, a high difference between the actual and the last sensed data from the same island triggers new islands creation. Other triggers may be defined depending on the type of crops. In the following paragraphs, we explain the island creation algorithm and the corresponding data gathering approach.

4.2.1. Island Creation Algorithm

The goal of this algorithm consists of grouping the sensor nodes having usually near values of sensed data into the same island. For this purpose, deployed sensors exchange the sensed data and compare them with the received values of their neighbors. Neighbor nodes having sensed data, that values are in a given precision interval $[Vmin, Vmax]$, are considered in the same island. Otherwise, a new island is defined when a neighbor node senses data that values are out of the precision interval. We suppose that network is connected, which means that each node can be reached. The neighboring nodes exchange the following messages:

- *CMsg* (*Check Message*): containing the dynamic sensed data (such as temperature T), the neighbor nodes check the similarity of T with their sensed data (belonging to a given $[Vmin, Vmax]$ interval).
- *NIMsg* (*New Island Message*): Announce a new island definition.
- *EMsg* (*End Message*): allow the Island Master *IM* to know the node's identifiers of its island.

We denote:

- D_i : the data sensed by node i .
- T_i : the instant when node i receives a *CMsg*.
- LT_i : the last time when node i received a *CMsg*.
- Delay: the maximum delay to construct an island.
- Dbt : the minimum delay between two triggers to reconstruct islands.
- i is an end node if it has not descendant neighbors.
- *CMsg* (n, E, D): a *CMsg* sent by node n , passing by nodes in the set E . E is initially an empty set. D is the data captured by node n .

The flowchart of the proposed algorithm is shown in **Figure 6**. The first node triggered to launch the algo-

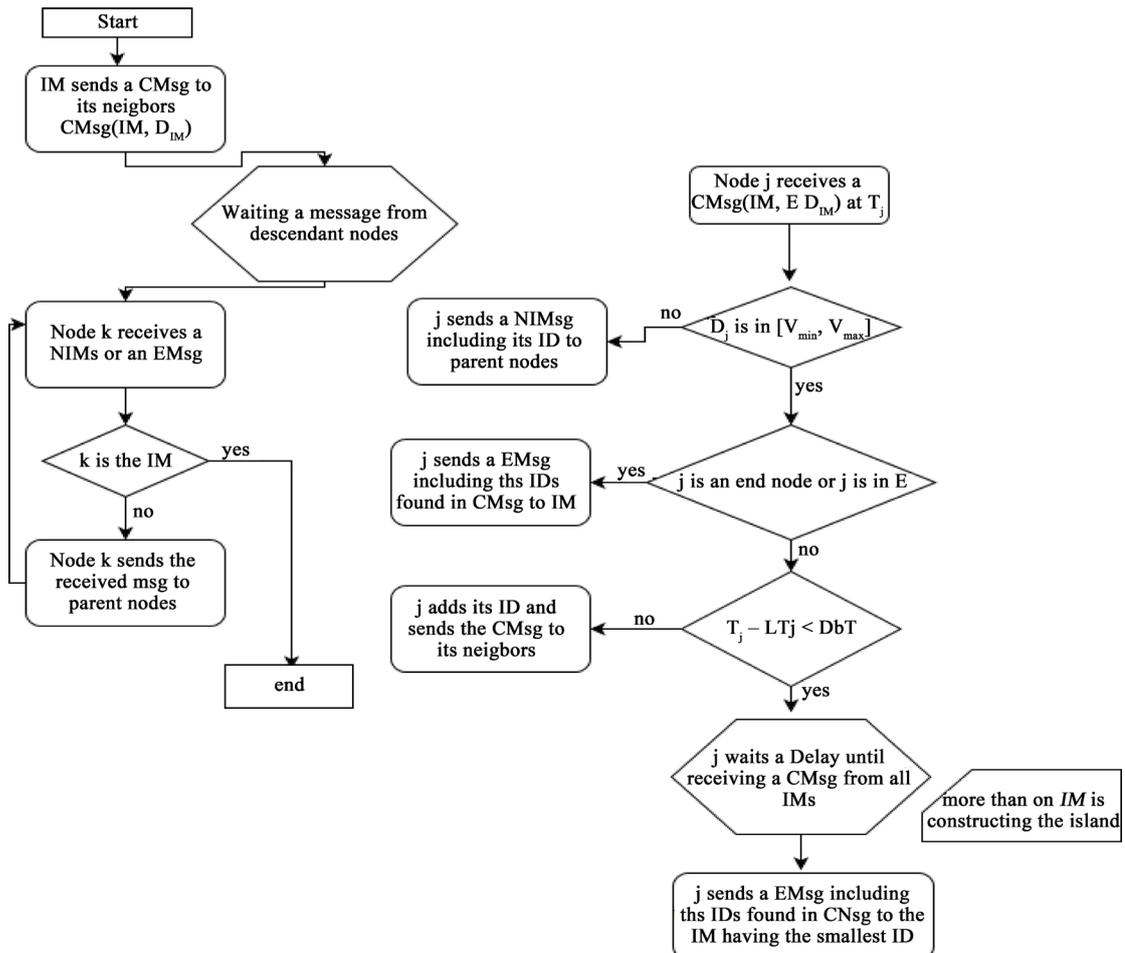


Figure 6. Flowchart of the island creation algorithm.

rithm, is declared *IM*. It sends a *CMsg* (*IM*, *TIM*) containing its *ID* and the sensed data to its neighbors. Each node *j*, receiving a *CMsg*, compares the received data with that sensed. In case of belonging to the precision interval, then if node *j* is neither an end node nor a node in a loop situation, than it sends the data to its neighbors after adding its *ID*. Otherwise, it sends an *EMsg* to the parent node (source node). In the other case, a *NIMsg* is sent to parent node.

An alternative circumstance could occur when more than one *IM* is declared in the process. This situation is marked by receiving more than one *CMsg* from different source nodes within a *Dbt* delay. In that case, the node waits until receiving all *CMsgs*, and then it send an *EMsg* to the *IM*, having the smallest *ID*, after including all nodes known from other received *CMsgs*.

We notice that a node *i* concludes that it has received a *CMsg* (*E1*, *T*) in a loop situation when the last received *CMsg* (*E2*, *T*) is such that $E1 \subset E2$. After performing the algorithm, each *IM* maintains the *IDs* of nodes in its island and the *IMs* of neighbor islands.

4.2.2. Island Creation Example

For the sake of clarity, an example of islands creation with 3 distinct temperature intervals: (27) (29) (30) (32) and (33) (35) is illustrated in **Figure 7**.

Initially, the node 1 sends a *CMsg* including its identifier and the sensed temperature (30) to its neighbors. Each node receiving *CMsg* checks if its sensed data is in the corresponding temperature interval. In that case, it adds its identifier and forwards the *CMsg* to its neighbors. Otherwise, it sends a *NIMsg* to its parent nodes. Nodes receiving a *NIMsg* forward it to the *IM*. In this case, node 9 senses a data that value is out of the interval (30) (33). So, it is declared *IM* and sends a *NIMsg* to the source node (node 3). This one forwards the message until the *IM* (node 1). Node 9 does not have descendant node. So, it forms an island containing only itself (node 9). Node 6 receives a *CMsg* in a loop situation, and then it will send an *EMsg* to *IM* (node 4).

4.2.3. Data Gathering Scheme

In this paper, we are interested to precision agriculture applications. In that case, network is not necessarily connected. However, knowing that the DC maintains the positions of all deployed nodes, we apply the island creation algorithm to each connected part of the network. Then, the DC will visit the different nodes that are declared *IM*. The first tour of the DC is a discovery tour. In the issue, the DC will maintain the real *IM*'s list in an ascendant order of their identifiers and the identifiers of nodes composing each island.

The *DC* will visit only one sensor of each defined island *IM*. In order to balance the energy consumption between all nodes, *IM* is alternatively selected from island nodes and with the condition that at the same time of the day different *IM* are selected. Knowing the *IM* in each tour, a node can select its tour to be an *IM*. During the

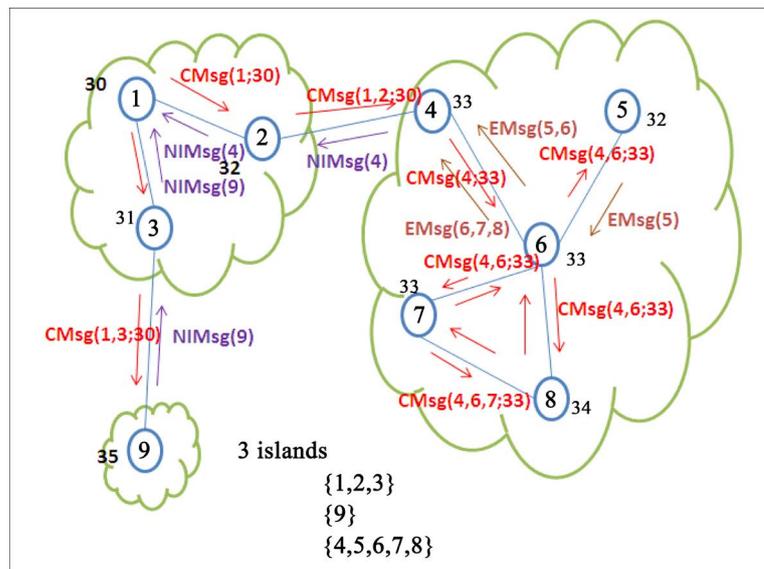


Figure 7. Example of island creation.

other tours, it remains in sleeping mode in order to conserve its energy. The reconstruction of islands is triggered by the DC.

5. Simulations and Results

In this section, we present simulation results to evaluate the efficiency of the proposed data gathering strategies. We carried out several simulations using NS2 simulator.

5.1. Simulation Parameters

Table 1 summarizes the simulation networking parameters.

5.2. Simulation Results

We analyze the strategies efficiency in the two following case:

- case 1: using the first approach for which we suppose that readings are done every 1 hour when the temperature goes on or above the threshold level, here it is 30 degree Celsius. Otherwise, readings are done every 2 hours.
 - case 2: using the first approach, readings are done every 1 hour whatever the previous sensed temperature.
- In Figure 8, we plot the average residual energy. We observe a minimum gain of 25% of energy in the first

Table 1. Simulation parameters.

Network configuration	Number of fields	3
	Fields area	(200 m × 200 m), (300, 300), (400, 400)
	Rc	50 m
	Bandwidth	2 Mbps
	Mac protocol	802.11
	DC speed	1 m/s
Energy model	Type	Battery
	Initial energy	100 Joules
	Transmit power	0.36 watt
	Receive power	0.24 watt
	Sensing power	0.015 watt

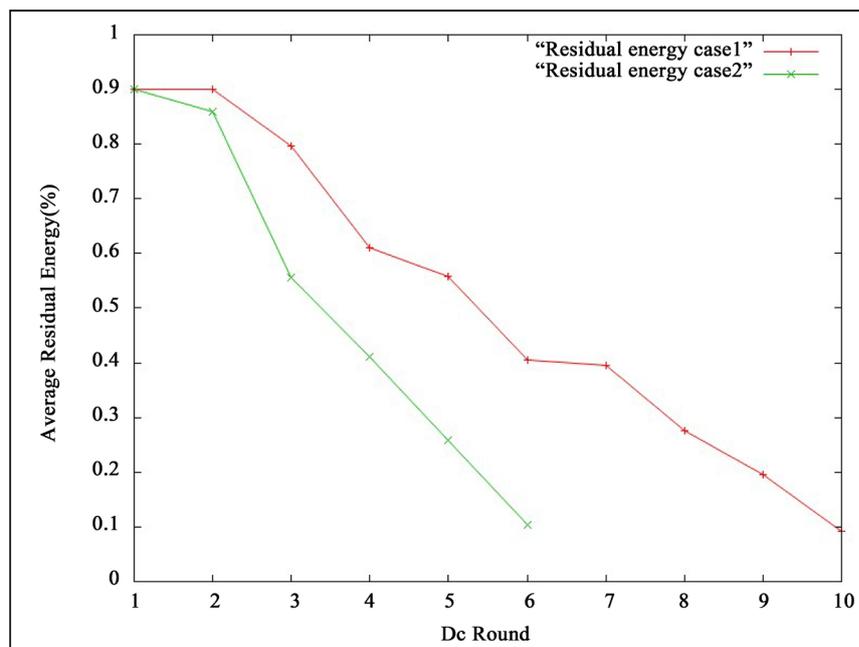


Figure 8. Average residual energy—first approach.

case. So, when introducing context awareness based on the dynamic parameter temperature, we can gain in terms of network lifetime. In the first case, the DC can do more than 10 tours before sensors failure; however in the second case only 6 tours are done.

Similarly, in **Figure 9**, we compare the first approach case 2 and the second approach using readings every 1 hour. We observe that the second approach is more relevant with making sleep some nodes of a given field. In fact, only one sensor of the island, IM, must sense and send the sensed data to the DC, the other island nodes sleep.

In this section, we evaluate the effect of the node islands creation on the efficiency of the solution. **Figure 10**

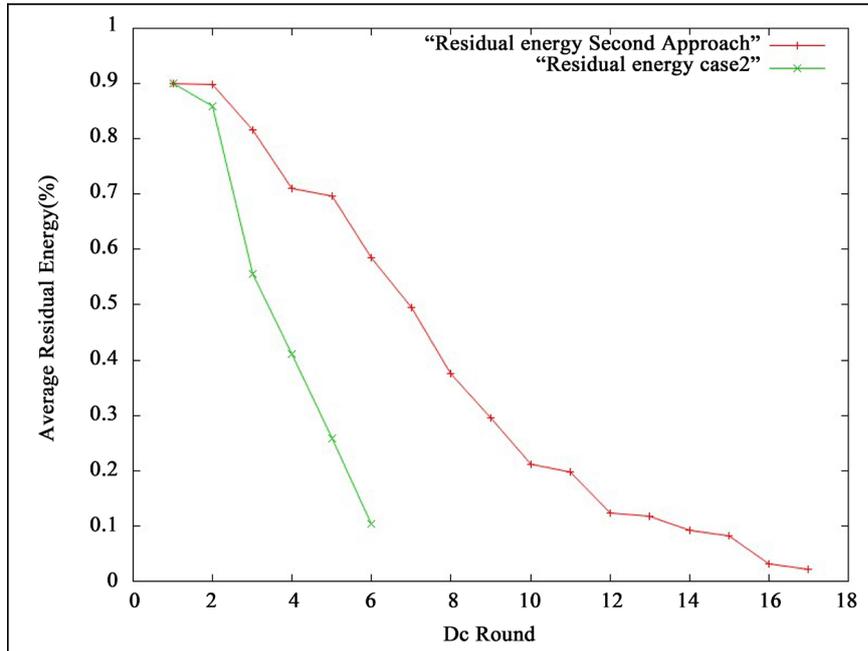


Figure 9. Average residual energy—first and second approaches.

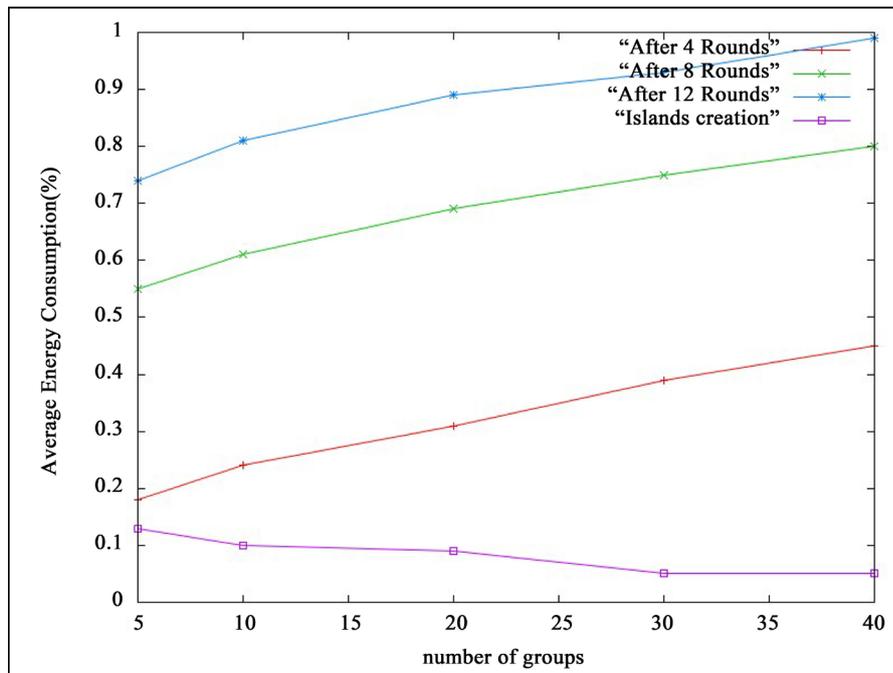


Figure 10. Average energy consumption.

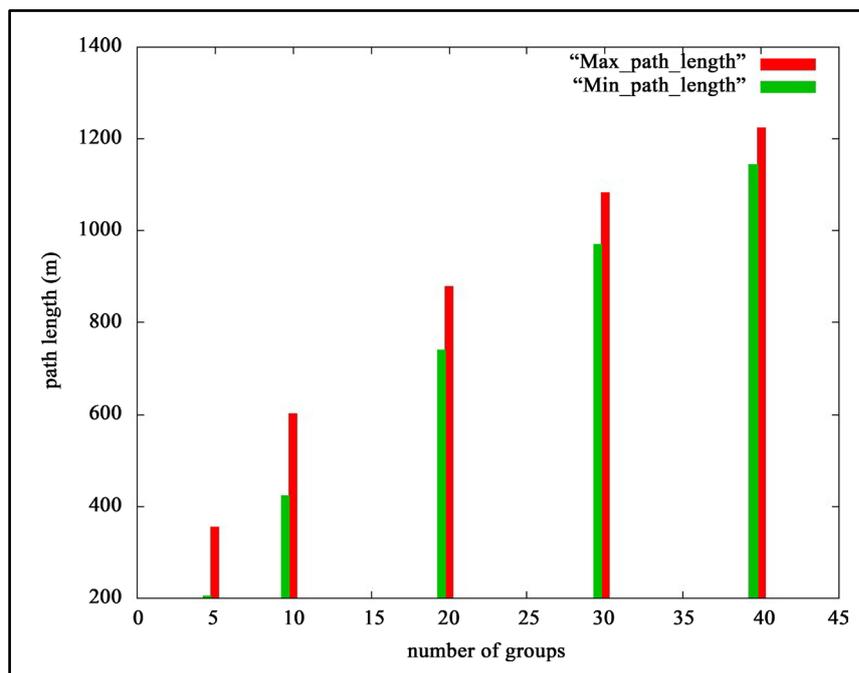


Figure 11. Maximal and minimal path length.

presents the energy consumed rate after 4, 8 and 12 tours. Besides, we present the energy consumed for island creation. Obviously, we found that the energy increase with the number of tours. Moreover, when the number of groups increases, the corresponding number of island masters increase. So, we observe that the energy consumed will increase with the number of active nodes representing the island master nodes.

Figure 11 depicts the maximal and minimal path length according to the number of groups. This result has an effect on the data delivery delay from sensed nodes to the base station. The path length increases with the number of islands. However the gap between minimal and maximal path length decrease when the number of islands increases. In fact, regrouping more nodes in the same island means that the distance between nodes increase, so, the island master selection will affect considerably the path length.

6. Conclusion

The aforementioned approaches have incorporated context awareness for gathering data from an agricultural parcel. Uniform and dynamic parameters are defined. Depending on these parameters, data gathering strategy is defined with optimum energy preservation. Our approaches have been evaluated based on some simulations scenarios but the feasibility of the system and their productivity cannot be proved without real system tests. In order to prove their feasibility, our future work consists of modeling the system with an adequate analytical approach.

References

- [1] Martinelli, M., Ioriatti, L., Federico, V., Benedetti, M. and Massa, A. (2009) A WSN Based Solution for Precision Farm Purposes. *IGARSS*, 469-472.
- [2] Zhang, W., Kantor, G. and Singh, S. (2004) Integrated Wireless Sensor/Actuator Networks in an Agricultural Application. *Proceedings of the 2nd Academic International Conference on Embedded Networked Sensor Systems (Sensys 04)*, Acm Press, 317-318. <http://dx.doi.org/10.1145/1031495.1031560>
- [3] Anurag, D., Roy, S. and Bandopadhyay, S. (2008) Agro-Sense: Precision Agriculture Using Sensor-Based Wireless Mesh Networks. *First ITU-T Kaleidoscope Academic Conference*.
- [4] Burrell, J., Brooke, T. and Beckwith, R. (2004) Vineyard Computing: Sensor Networks in Agricultural Production. *Sensor and Actuator Networks*, **3**, 38-45.
- [5] Lpez Riquelme, J.A., Soto, F., Suardaz, J., Snchez, P., Iborra, A. and Vera, J.A. (2009) Wireless Sensor Networks for

- Precision Horticulture in Southern Spain. *Computers and Electronics in Agriculture*, **68**, 25-35. <http://dx.doi.org/10.1016/j.compag.2009.04.006>
- [6] Garcia-Sanchez, A.-J., Garcia-Sanchez, F. and Garcia-Haro, J. (2011) Wireless Sensor Network Deployment for Integrating Video-Surveillance and Datamonitoring in Precision Agriculture over Distributed Crops. *Computers and Electronics in Agriculture*, **75**, 288-303. <http://dx.doi.org/10.1016/j.compag.2010.12.005>
- [7] Liu, Y., Zhang, J. and Richards, M. (2009) Towards Continuous Surveillance of Fruit Flies Using Sensor Networks and Machine Vision. *Proceedings of 5th International Conference on Wireless Communications, Networking and Mobile Computing*, 1-5.
- [8] Liu, H., Meng, Z. and Wang, M. (2009) A Wireless Sensor Network for Cropland Environmental Monitoring. *Proceedings of the IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC 09)*, Wuhan, 25-26 April 2009. <http://dx.doi.org/10.1109/nswctc.2009.306>
- [9] Goumopoulos, C., Kameas, A. and O'Flynn, B. (2007) Proactive Agriculture: An Integrated Framework for Developing Distributed Hybrid Systems. *Proceedings of the 4th International Conference on UIC*, 214-224. http://dx.doi.org/10.1007/978-3-540-73549-6_22
- [10] Dursun, M. and Ozden, S. (2011) A Wireless Application of Drip Irrigation Automation Supported by Soil Moisture Sensors. *Scientific Research and Essays*, **6**, 1573-1582.
- [11] Ayday, C. and Safak, S. (2009) Application of Wireless Sensor Networks with GIS on the Soil Moisture Distribution Mapping. *Proceedings of the GIS Ostrava Symposium*.
- [12] Cao, X., Chena, J., Zhang, Y. and Sun, Y. (2008) Development of an Integrated Wireless Sensor Network Micro-Environmental Monitoring System. *ISA Transactions*, **47**, 247-255. <http://dx.doi.org/10.1016/j.isatra.2008.02.001>
- [13] Brinis, N., Azouz Saidane, L. and Minet, P. (2012) EDGM: Energy Efficient Data Gathering with Data Mules in Wireless Sensor Networks. *6th International Conference on Sensor Technologies and Applications, SENSORCOMM 12*, Rome, 19-24 August 2012.
- [14] Brinis, N., Saidane, L.A. and Minet, P. Adaptive Delay Aware Data Gathering Strategy for Wireless Sensor Networks. *ICAIT12*.
- [15] Commuri, S. and Watfa, M.K. (2006) Coverage Strategies in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, **2**, 333-353. <http://dx.doi.org/10.1080/15501320600719151>
- [16] Ingole, P., Pawar, J. and Vhatkar, H. (2013) Context Aware, Mobile Phone Based Sink Model of Wireless Sensor Network for Indian Agriculture. *Advances in Computing, Communication and Control Communications in Computer and Information Sciences*, **361**, 301-308. http://dx.doi.org/10.1007/978-3-642-36321-4_28