International Journal of Computer Networks & Communications (IJCNC) Vol.2, No.5, September 2010

AGENT STRATEGY DATA GATHERING FOR LONG LIFE WSN

Ahmad Sardouk[1], Rana Rahim-Amoud[2], Leïla Merghem-Boulahia[1] and
Dominique Gaïti[1]
[1]ICD/ERA, UMR 6279, Troyes University of Technoloy, Troyes, France
{ahmad.sardouk, leila.boulahia, dominique.gaiti}@utt.fr
[2]LaSTRe, Université Libanaise, Kobeh, Tripoli, Lebanon
rana.rahim@ul.edu.lb

ABSTRACT

The main goal of wireless sensor networks (WSN) is to gather information from the regions of interest through a large number of micro sensor nodes. This gathering is traditionally done by using a client/server communication approach. However, this communication architecture consumes a lot of power and does not take into consideration the information properties. In this paper, we propose a data gathering scheme for WSNs, based on agents cooperation to deal with the importance of the information. This agent cooperation aims to reduce an important amount of the information communicated over the network by eliminating the unimportant information and the inter-sensor-nodes redundancy. This cooperation is empowered by an agent strategy taking into account several parameters related to the node and to the instance of communication for an optimized power management. Successive simulations proved, in large scale WSNs and different densities, the ability of the proposed gathering scheme to reduce the average power consumption of sensor nodes and hence, to extend the network life time.

KEYWORDS

Wireless Sensor Network, Multi-agent Systems, Energy-efficiency, Communication Architecture, Autonomic sensor networks

1. INTRODUCTION

The WSNs are generally composed of a large number of dense, randomly deployed and energy limited nodes. To the best of our knowledge, processing the information locally in the sensor nodes is very cost effective compared to its communication. This is especially due to the fact that a lot of the sensed information could be redundant or not important. For that, an optimized data gathering could be a good technique to save the power of the sensor nodes and hence, to extend the network life time.

In this paper, we propose a scheme for data gathering in WSNs. This scheme reduces heavily the power consumption and tackles, simultaneously, the density of the network and the scalability problems. This scheme aims to extend the network life time by defining a strategy for data gathering. This strategy takes into consideration several parameters to ensure an enhanced power management within a WSN. In addition to the resident power in the sensor node, this strategy gives a big interest to the position of the sensor node within the network. The importance of this point arises for nodes in critical positions, which relay two parts of the network and hence their failure may lead to a network partitioning. The importance of the information is also a key parameter in this strategy. By reducing the amount of the unimportant or non useful communicated information, important power consumption could be saved. In addition to the energy, the position, and the sensed information, this strategy deals with the density of the network around each node. In the dense zones of the network, the redundancy of information could be easily mentioned. These strategy parameters will be discussed in section 4.The proposed scheme is based on multi-agent systems (MASs) [1]. An MAS can be defined as a group of agents able to interact and to cooperate in order to reach a specific objective. In the

DOI : 10.5121/ijcnc.2010.2505

current work, we propose to implement an agent in each sensor node, in order, to process locally the data of its sensor node and to cooperate with its neighbouring agents to gather their processed data and to eliminate the inter-sensor-nodes redundancy. Each agent executes also the proposed strategy in order to make the appropriate decisions (to cooperate or not).

The paper is organized as follows. In section 2, we review the proposed data gathering solutions for reducing the power consumption in WSNs. Our solution and the details of the agent strategy are described in section 3 and 4, respectively. Next, in section 5, we present our simulation setup parameters. Then, in section 6, we evaluate and analyze the performance of our proposed solution. Finally, the conclusion and future works are given in section 7.

2. DATA GATHERING BASED SOLUTIONS FOR ENERGY EFFICIENT WSN

The basic role of sensor nodes in a WSN is to gather information from the environment. This gathering should respect the finite battery of the sensor node to maintain the longevity of the network.

The traditional model of data gathering is the client/server (CS) communication architecture [2]. In this architecture, when the sensing unit of a sensor node perceives information from the environment, it sends it directly as it is (raw) to the sink to be processed through intermediate nodes causing supplementary power consumption. Thus, several works [2][3][4][5][6][7][8][9] have been done to optimize this traditional client/server communication architecture. In the following, we will discuss some of those schemes focusing on their goals and limitations.

Under the scheme of data compression/decompression, the authors of [3][4] proposed a data correlation algorithm that compresses, in a distributed manner, the data in a WSN. In this proposal, only one node is elected to send raw data to the sink and the others only send coded data. After receiving the sensing data, sink node decodes it through the correlations between the compressed and uncompressed data. The key step of these proposals is to find a good coding algorithm supporting multi-rate compression and an energy-aware and low complexity correlation-tracking algorithm. This work is important, however, it is quite difficult to find a non complex coding algorithm suitable for sensor nodes in WSN where we should not consume a lot of power in processing.

In [5], the authors propose to merge the information of a maximum number of nodes. Thus, they proposed a serial incremental fusion (or data fusion (DF)) which can be described as follows. When a node sends its data to the sink, intermediate nodes merge their data with the first node data. As the information of multiple sensor nodes are merged into one message (one overhead instead of many ones), this solution saves some energy. However, intermediate nodes do not always have important information to send and they do not eliminate the unimportant or redundant information.

The authors of [6] present an ant colony based data aggregation for wireless sensor networks. They try to tackle the problem of constructing an aggregation tree for a group of source nodes within the WSN to send sensory data to a single sink node. The proposed mechanism assigned artificial ants to source nodes to establish low-latency paths between the source nodes and the sink. They suppose that every ant will explore all possible paths from the source node to the sink. The obtained paths will create, by accumulating the pheromones, a tree were the sink node is the root. This tree will firstly replace the necessity of routing protocol and then it will be used as a data aggregation tree. By exploring all possible paths to the sink, each ant consumes extra power that could be eliminated. Furthermore, the construction of an appropriate tree depends heavily on the nodes' deployment, which is generally random. Such tree construction consumes an important amount of power.

In [7], the authors propose an adaptive data aggregation (ADA) scheme for clustered WSN. Their goal is to minimize the processing of the data aggregation required at sensor nodes and shift the burden to the resource-rich sink node. ADA is based on three main points. The first one is the reporting frequency at sensor nodes, which is the sensing interval time. The second point is the aggregation ratio at cluster heads (CHs). In fact, when a CH receives the data from the source nodes in its cluster, it sends a part of them to the sink node. By sending a part of the information to the sink node, this latter estimates the relevance of the WSN's information at the same time (t). The last point is that the reliability of a data aggregation will be defined according to the relevance of this information. The authors propose to adjust the reporting frequency at sensor nodes and the aggregation ratio at cluster heads following the reliability at time t. The main importance of this work is related to the incorporation of an adaptive behavior into protocols in such dynamic networks. However, ADA is based on the cluster heading paradigm, which requires an expensive construction in term of energy. Furthermore, in the implementation of this scheme, the authors did not address the complexity issue and the amount of power consumption required to build such clustered WSNs. In addition, authors neglected to study the importance of scalability of such kind of networks.

In addition to [6] and [7], authors in [10][11][12][13] have also proposed a structured (tree and cluster) based data aggregation for WSNs. However, according to [14] and [15], firstly, the overhead of construction and maintenance of the structure may outweigh the benefits of data aggregation. Secondly, structured approaches that centrally compute the aggregation tree are not practical for dynamic scenarios, due to excessive communication overhead and the centralization of the WSN's structure management. Thirdly, the performance of structured approaches is sensitive to the waiting period (data coming from all upstream nodes) at the intermediate nodes. A small period of time can lead to a loss in the accuracy of the aggregated data and a long period can lead to higher latency. Moreover, computing the optimal period requires knowledge of the relative position of the node with respect to its entire subtree, which may not be known accurately. Fourth, in cluster-based sensor networks, sensors transmit data to the cluster head where data aggregation is performed. However if the cluster head is far away from the sensors, they might expend excessive energy in communication.

The authors of [2][8][9] propose the agent technology to gather the information. The main work of these authors is based on the use of Mobile Agents (MA) in WSNs for an energy-efficient data gathering. In these proposals, the MA is defined as a message which contains an application code, a list of source nodes (predefined by the network administrator or the sink) and an empty field to put the gathered data. The message contains also a big header including the required fields (such as the next destination and next hop fields) to route it in the network following the list of source nodes and the header. The MA gathers the information from the source nodes. At each source node, the MA processes the collected data locally and concatenates it with previous source nodes' data. Indeed, these solutions reduce the power consumption in low bandwidth and power constrained networks such as WSN. However, the size of the mobile agent message is large enough to waste an important part of this reduced power, when sending it on the WSN. Another drawback of this type of solutions is the difficulty to create the source nodes' list and to define the starting time of data gathering. Indeed, the sink is not able to decide by itself when the nodes have important data to send. Furthermore, this solution appears to be so far from the scalability issue imposed in a lot of WSN applications. This deficiency appears when measuring the required time for a mobile agent to gather the information from far regions and the number of source nodes' lists and mobile agents needed to gather information from the entire network. Another limitation is the definition of the regions which will be treated by the MA.

After analyzing the previously presented solutions, we can see clearly that there is a serious problem in terms of scalability for all of them. We can also deduce that there is still a lot of work in term of energy-efficiency with paying attention to the packet delivery ratio and latency. Moreover, the needed solution should be independent from the network deployment. It should be efficient for a randomly deployed network as for organized deployment. Therefore, in this paper, we propose an agent based data gathering approach for large scale WSNs. The main idea is based on a distributed computation using the facilities offered by the agent system. Furthermore, this approach takes into account several parameters: the importance of the sensed information, the resident power in each node, the density of the network and the position of the sensor node within the network. These parameters are presented in section 4.

3. STRATEGY-BASED COMMUNICATIONS

In this section, we describe our Strategy-Based Communications (SBC) for an energy efficient data gathering. This gathering scheme is based on the multi-agent approach to treat the sensed data locally within the nodes, eliminate the inter-sensor-nodes redundancy and to gather the important information of the WSN. Firstly, we introduce the key points of SBC, then we present its system design. Next, we describe the knowledge base of our agent and a data gathering scenario to illustrate the SBC. Finally,we give an example of the cooperation message structure.

According to [17], a sensor node expends a maximum energy in data communication, while the energy expenditure in data processing is much less compared to it. The energy cost for the transmission of 1KB over a 100m distance is approximately the same as that for the execution of 3 million instructions by a 100 million instructions per second processor. Hence, we have defined three main points to save the power of each node and to extend the lifetime of the network:

1. The first one is the Information importance Based Communication (IBC). Its main role is to reduce the number of communications. Thus, by estimating the importance of the information locally (in the sensor node), it is possible to prohibit an important number of communications corresponding to non important or redundant information. A good example of this kind of information could be seen in the case of monitoring a stable environment, where, the sensor node may not detect new values or events during a long time. Consequently, the difference between the gathered values will not be important. Therefore, by considering the IBC, a communication is only started by the detection of relevant information;

2. The second point concerns the Elimination of unimportant inter-sensor-nodes Information. Generally, sensor nodes are deployed randomly (for example, a plane throws them in hazard zones). Thus, two or more sensor nodes can cover almost the same area and they will give always the same information (inter-sensor-nodes redundancy);

3. The last point appears in the Data Concatenation. Due to protocol overheads, the communication cost, (in terms of energy), to send a long message is usually less than that of sending the same amount of data using many short messages [2].

These three points will be discussed within the system design of our proposal.

3.1. System Design

A sensor node has generally a physical autonomy presented by (1) its battery that represents its life source, (2) its processing unit to treat information, (3) its memory to save gathered data and (4) its radio entity to communicate this data. To convert a sensor node to a fully autonomous entity, we should empower it by some capacities. A sensor node should be able to make decisions to estimate, for example, the importance of the sensed information. It should be also able to cooperate with other sensor nodes in order to eliminate the inter-sensor-nodes redundancy and/or to concatenate data. Therefore, we have chosen to use a multi-agent system to bring up the full autonomy to the wireless sensor networks. To illustrate SBC, we use the topology presented in Figure 1.

In SBC, we associate an agent to each sensor node. This agent processes the sensed data locally, and estimates its importance. It also makes decision to communicate this information and/or to cooperate with other neighbouring agents in order to eliminate the inter-sensor-nodes redundancy and concatenate the processed information of other sensor nodes. The concatenation and redundancy elimination mechanisms that we used are described in our previous work [17]. In another previous work [18], we have also defined some mechanisms to process the information and to estimate its importance. These mechanisms will be used and enhanced in our present work.

Figure 2 presents the position of the agent in the protocol stack of the sensor node. The agent is implemented in the application layer; however it could be seen as a cross-layer entity as it uses the information of the routing level to build its dedicated view. In general, an agent is interested in the events occurring in its neighbourhood. That is why, in SBC, the local view of the agent will be restricted to its one hop neighbours and the first node on its path to the sink. An agent is also able to know if it is the first hop on the path to the sink for other nodes.

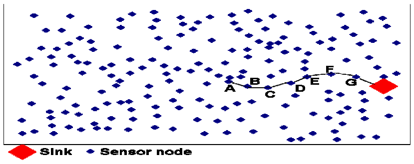


Figure 1 Network topology

In this work, we only address the application and network layers. However, in our future works, we intend to address the actions of our agents on each layer of the sensor node protocol stack, in order to realize a better power-efficient data gathering.

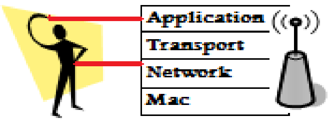


Figure 2 Agent position in the sensor node protocol stack

3.2. Agent Knowledge Base

One of the basic attributes of an agent is to be situated (situatedness [19]). That is, an agent is a part of its environment. Its decisions are based on what it perceives from this environment and on its current state [20]. The situated view of an agent is then composed of the information obtained from its local observation and the information exchanged with its neighbours. For that, we should define carefully the required information for the agent to achieve its goals. This information will be stored in a knowledge base (KB), which contains the list of the one hop neighbour agents, the first agent in its path to sink and also, if the agent is a first agent on the path to the sink for other agents. These information could be obtained through the routing layer. The KB contains also other kinds of information presented in details in section 4.

In order to respect the memory constraints of the sensor nodes, the amount of the KB information is limited. Thus, SBC is designed to not require a memory size for routing information as each sensor node needs only to know its first hop in the path to sink. Even, the route maintenance is lighten to respect the constraint of the sensor node. Thus, when the battery of node *i* reaches a predefined threshold, node *i* sends an alert message to its one hop neighbours. Upon the reception of an alert message, node *j* verifies if node *i* is its first hop in the path to the sink. If it is the case, node *j* sends a help message to its one hop neighbours. Each neighbour responds the node *j* by sending a help message containing its residual power. Node *j* defines the neighbour with the highest residual power value as its new first hop in the path to the sink.

3.3. Gathering Session Scenario

Figure 3 illustrates the main role of the agents implemented in the sensor nodes during a gathering session, which starts when an agent (sensor node) detects important information. This agent invites its one hop neighbours[1] to cooperate in order to gather the maximum possible processed information and to create a cooperation message (presented in section 3.4) summarizing these collected information[2]. However, the neighbouring agent, who is at the same time the first hop on the path to the sink for the agent in question (source node), will not respond to the cooperation request. Indeed, once the cooperation message is ready, this neighbour agent (called intermediate agent) will receive the message[3] and will invite its one hop neighbouring agents to cooperate[4]. The intermediate agent will gather the information of its one-hop neighbours[5] and extend the initial cooperating message. This message will be then sent to the next intermediate agent[6]. The new intermediate agent will, on its turn, repeat the same scenario. This scenario will be repeated until is reached the sink node.

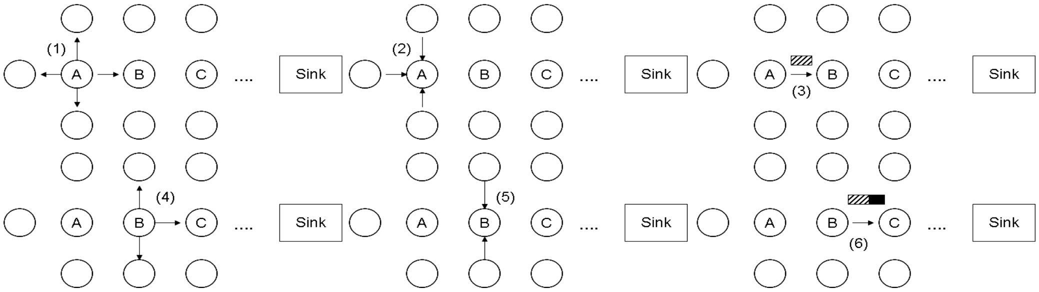


Figure 3 Gathering session scenario

In the following, we present an example to illustrate the main role of the agents implemented into the sensor nodes during a gathering session. In this example, we consider the network topology as shown in Figure 1.

---

[1] The values from (1) to (6) indicate the steps on Figure 3

First of all, we suppose that the sensors of node *A* detect information. The information is sent directly to the corresponding agent (agent *A*) to be processed. After processing, we consider that agent *A* estimates this information as important.

Secondly, due to the importance of the detected information, agent *A* sends a cooperation request to its one hop neighbours. The cooperation request is a short message with a key field containing a predefined string as: *cooperation request* and includes in the data field the information detected in *A*. The communication of this request passes by a one hop broadcast. Indeed, the one hop neighbours will be programmed to not re-broadcast it.

By sending the cooperation request, agent *A* invites its one hop neighbour agents to join cooperation for a data gathering session. A neighbour agent makes the decision to cooperate or not based on a well defined strategy. This strategy takes into consideration several parameters: the importance of the locally sensed information, the resident battery in the sensor node (the energy), the position of the sensor node within the WSN and the density of the network. These parameters will be discussed in details in section 4. According to their local information, the cooperating agents decide cooperatively if the information is as important as estimated by *A* or not. For example, if *A* was in a sleep mode, thus it detected an event lately when it has been waked up, while this information has been previously reported by its neighbours. Hence, the gathering session could be cancelled, if the information is finally judged unimportant by the majority of cooperating nodes.

After taking the appropriate decision, each cooperating agent responds (within a predefined fixed cooperation delay) by sending its processed data. These data will be concatenated within a cooperation message after an inter-sensor-nodes redundancy elimination [18]. This message contains two main parts. The first one is for the sensor nodes' addresses and the second one is reserved for the corresponding processed data.

In a one hop communication, the probability of loosing a cooperation request message is very low, and through several simulations, we have discovered that it is close to zero percent. Thus, *A*, which is the sender of the cooperation request, will wait to receive the messages of the cooperating agents during a fixed delay, then it sends the cooperation message to its first hop on the path to the sink which is agent *B* (Figure 1). Node *A* considers the agents of the non received messages as non cooperating agents.

As agent *B* is a one hop neighbor for agent *A*, it has previously received the cooperation request sent by agent *A*. Agent *B* did not respond this request as it knows, from its knowledge base, that it is an intermediate agent (on the path to the sink for agent *A*). When receiving the cooperation message, agent *B* sends its cooperation request to its one hop neighbours to gather their processed data.

Agent *A* and agent *B* may have some common one hop neighbour agents. These common neighbours receive two cooperation requests but answer only the first request and neglect the second one. In addition, the initiator agent (*A* in this example) identify the gathering session by a unique sequence number. This, number is generated randomly and will be stored for a short predefined delay in each cooperating agent to avoid a double participation in the same data gathering session.

Agent *B* concatenates its cooperation message with the initial message received from agent *A*. Then, it sends it to its first hop on the path to the sink, which is agent *C*. Finally, agent *C* and all the intermediate agents repeat the same procedure as agent *B* until reaching the sink node.

In the next section, we present the structure of the cooperation message used by the implemented agents, in order to gather the information and to send it to the sink.

3.4. Cooperation Message

Figure 4 presents the cooperation message structure that will be sent to the sink node. This message is divided into two parts separated by a flag. The first part contains source nodes' addresses. Since all the nodes have the same address length, there is no need to put a field separator between these fields.

The second part of this message contains the assembled data that are separated by a data length field. This latter is used as an indicator to show the end of each data field. If the data length is zero, this means that the corresponding node has the same data as the previous one. Therefore, the next field will be a data length field and not a data field. This represents the case of full data redundancy between the nodes. It is important to note that each address field located in the first part corresponds to a data field in the second part.

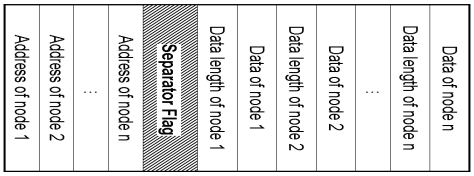


Figure 4 Cooperation Message

The next section details the strategy used by each agent to make its appropriate decision to cooperate or not in a data gathering session.

4. AGENT STRATEGY

The agent that we use in our proposal has two possible behaviours. The first one is the selfish behaviour, i.e., the agent cooperates, with other agents, the less possible in order to save the maximum of its battery. Indeed, the reduction of cooperation may lead to the loss of some appropriate information. For example, an agent asks its neighbour agent to cooperate in an object localization task; if the neighbour agent is applying a selfish behaviour, it will not respond the cooperation request and hence, the object localization task may fail. However, this behaviour could be useful, in some cases, to the survivability of the whole network as we will explain in section 4.3. The second behaviour that our agent could apply is the cooperative one. i.e., the agent accepts to cooperate with a maximum possible of the received cooperation requests. Indeed, the agent consumes some power when cooperating, however it maximizes the gathered information and hence, reduces the probability of loosing an important information.

In this section, we define the strategy that allows the agent to select the better behaviour. The strategy selection is completely distributed, where each agent decides its behaviour according to several parameters. These parameters are as follow:

- Energy (E);
- Network density (D);
- Position of a sensor node within the network (P);
- Information importance degree (I).

We express the defined parameters by the equation (1) to compute the relevance (R) of a cooperation. Each parameter is multiplied by a coefficient (that is called impact factor) according to its importance. More important is the impact factor more is its influence on the computed R. Then, based on a predefined threshold of R, the agent decides to apply a selfish (R< threshold) or a cooperative behaviour (R ≥ threshold).

Finally, to overcome the unit differences between the parameters, we change their values to percentages. The manner of passing is described in the next parts with the explaining of the parameters.

$$R = E \times w_e + \frac{1}{D} \times w_d + P \times w_p + I \times w_i \tag{1}$$

Where w_e, w_d, w_p, and w_i are the influence factors for the energy, the density, the position, and the information importance degree respectively.

4.1. Energy

The energy is an important parameter in a resource limited network such as the WSN. Indeed, the remaining battery level appears to be the most important value in this parameter but it is not the only one. In order to better use the energy of a node, we define an administrator power strategy parameter (APS), which allows the network administrator to influence the selfish behaviour of the agent. i.e., to extend the WSN life time, he could multiply the available battery level of the node by a percentage (APS) following its preferences. Consequently, the agent will reject some cooperation requests because it will consider that there is not enough energy. The same cooperation request would be accepted if the administrator strategy was disabled. This rejection allows sensor nodes to save more energy, and hence it extends the whole network life time.

Otherwise, the importance of the administrator power strategy could be emphasized also in the case of multi-application sensor network [21], where the administrator will be able to define the importance of each application. Therefore, the value of (E) is given by the equation (2):

$$E = \frac{(A \times APS)}{fillbattery} \tag{2}$$

Where E is the energy parameter in the equation (1), A is the available or remaining power in the battery and APS is the administrator power strategy parameter. As presented in equation (2), the energy E could just be the remaining battery level if no administrator power strategy has been defined (APS=1).

To express E in percentage, we divide the obtained value by the full battery.

4.2. Network density

The network density is the number of nodes per square meter. It varies from one deployment to another and from one node to another within the same deployment depending on the node distribution.

According to [16], this parameter does not have a fixed value to be used as a reference. The ideal value is application and environment dependent. In addition, this parameter has a network management importance as it helps to identify the dense zones of the network and the non well covered zones. Hence, it may lead to redeployment of more nodes in some zones for a better coverage.

In the current work, we suppose that the more an agent has neighbours, the less is the importance of its participation in a cooperation; that is why in the equation (1), we use the inverse of the density (D). To illustrate the importance of this parameter, let us take the example of a tracking application where the position of the desired object can be defined by at least three agents. Finding these three agents in a dense network is an easy task. However, if ten agents participate in this task, instead of three, we will have an undesired loss of power and time.

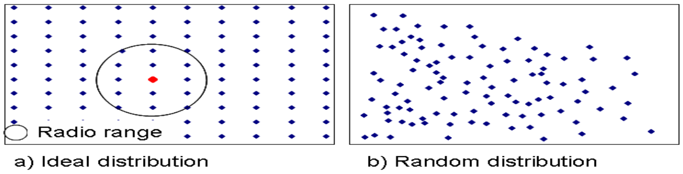
The density is computed by each agent. There are two main reasons behind that: the first one is that each agent has a situated view and there is no agent with a global network view. The second and most important reason is the fact that for a specific task, we need cooperation between the agents of the same zone (geographical part of the network) and not farther agents. For simplicity's sake, we propose the equation (3) to compute this density (D). In this equation, we compute the percentage of the real density compared to the theoretical density (both of them are explained later on), i.e., we may have density bigger than 100% in case of very dense zone (in this case, the tendency of the agent will be toward the selfishness). Otherwise, if the density computed by a node is equal to 0% the node is disconnected from the network.

$$D = \frac{realdensity(RD)}{theoreticaldensity(TD)} \quad \text{where,} \quad \begin{cases} RD = \dfrac{N_{real}}{\left(\pi \times r^2\right)}, \text{ and} \\ TD = \dfrac{N_{theoretical}}{\left(\pi \times r^2\right)} \end{cases} \tag{3}$$

Hence,

$$D = \frac{N_{real}}{N_{theoretical}}$$

Where r is the radio range of the node, $N_{theoretical}$ is the theoretical number of nodes and it is given from the ideal distribution of the nodes or the grid distribution (Figure 5a). $N_{theoretical}$ corresponds to the number of nodes within the radio range of a reference node (RN). A RN is a node in the centre of the area to eliminate the special cases of border nodes.



a) Ideal distribution b) Random distribution

Figure 5 Network topology

N_{real} is the number of the one hop neighbour nodes, appearing on the KB of the node in question. N_{real} should be equal to $N_{theoretical}$ in the ideal case. In Figure 5b, we show an example of randomly distributed nodes to give an idea about real network densities.

4.3. Position within the network

Before discussing the position parameter, it is important to mention that in the current work, we suppose that during the deployment phase, several nodes with global positioning system will be deployed. These nodes will be deployed for a short period of time. Their only role will be to identify the (x, y) position of the other sensor nodes.

In the current work, we define three types of node positions: (1) normal, (2) edge and (3) critical. The normal position is the position inside the network where the node has multiple neighbours. For this kind of nodes, the agent may tend to exhibit a cooperative behaviour to maximize the amount of the important information. The edge node is a node at the border of the network and having a restricted view of the network limited to only one neighbour.

A node is considered in a critical position if it connects two parts of the network. That means, if the node runs out of battery, the network will be partitioned or in the best case, several nodes will need to build a new route to the sink. This new route will be generally longer, and hence it is expensive in term of energy as the number of hops is increased. Figure 6 illustrates a case of a node in a critical position (n5). As we can see if n5 runs out of battery, the network will be divided in two parts.

The strategy should allow an agent in a critical position to decrease its power consumption to maintain the connection between the two parts of the network the longest possible time. Thus, the value of the importance factor of the node position should help the agent to apply a selfish behaviour and, hence, it should be greater than or equal to the energy or the information importance degree factors.

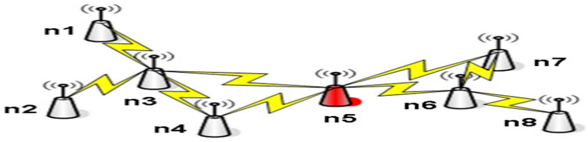


Figure 6 Node in critical position

To facilitate the computation of P, we have defined a fixed value for each type of node position. The values are 10%, 50% and 100% for the normal, edge and critical, position respectively.

4.4. Information Importance Degree

The last parameter is the information importance degree (I) that depends heavily on the desired application. This parameter could be computed by a local processing in the node. This processing allows the agent to estimate the importance of the gathered information.

Information is considered by an agent as important if it is the first information containing the desired object or a new event. For example, in a tracking application, if the detected object is the desired one or in the case of a visual application [22], if the captured picture contains an animal face (supposing that we are searching for new species in a forest), the agent will judge this information as important.

In other domains as in environmental monitoring (humidity, temperature, etc.), the agent saves the last gathered information to compare it with the newly gathered one. If the difference between both is greater than a predefined threshold, this information will be considered as important. However, the agent drops the old information and saves the recent one and marks the information as unimportant. The same technique could be used also in tracking, when the object stays in the sensor zone during two or more gathering cycles.

5. SIMULATION SETUP

In this section, we define the basic simulation and the agent strategy parameters. In order to demonstrate the performance of SBC, we chose to compare it to the client server (CS) approach as it is still the base for the majority of the proposals and actually, the most deployed one. In addition, we compare SBC to the data fusion (DF) proposal presented in section 2 that appears to be an interesting proposal.

We have implemented these approaches on GlomoSim [23] which is a scalable simulation environment for wireless and wired network systems.

In our simulation setup, as presented in Table 1, we summarize the different simulation parameters that we have used for the evaluation of our proposal. We have run our simulations over a 1000mx1000m square with a random distribution of nodes during 1000 seconds. We have limited the radio range and the data rate of each node to 87 meters and 1Mbps, respectively, as suggested in [24]. The transmission and reception power parameters, which influence directly the radio range, have been chosen carefully from the ranges defined in the sun spot system [25].

Table 1 Basic simulation parameters

Simulation Parameters	**values**
Network size	1000mx1000m
Node distribution	Random
Radio range	87m
Throughput	1Mbps
Size of sensed data	24 byte per node
Sensed Data Interval	10 seconds
Simulation time	1000 seconds
Local processing time	40ms

In order to test the scalability of our proposal and its relevance across different network densities, the simulations are done for a number of nodes varying from 100 to 900 nodes with an interval of 200.

The local processing time is fixed to 40 ms. This value is inspired from the work realized in [2].

The agent strategy parameters are presented separately in section 4. As summarized in cides consequently to cooperate

Table 2, the importance factors of these parameters w_e, w_d, w_p, and w_i are fixed to 0.30, 0.10, 0.30, and 0.30, respectively. These values reflect the importance of their corresponding parameters. By giving the same value to w_e, w_p, and w_i, we give the same importance to the energy, the position and the information importance degree in the calculation of the relevance value.

The density has a lower priority compared to the other parameters as we suppose that it does not influence directly the performance of the whole WSN. Based on these factors and through several simulations, we have found that the majority of the computed cooperation relevance (R) values were between 0.6 and 0.8. Thus, we have chosen to fix a threshold of R to 0.7. By comparing the computed R to this threshold, an agent will decide to cooperate or not. Hence, if the computed R during a data gathering session is less than 0.7, the correspondent agent considers this cooperation as unimportant. However, if R is greater than or equal to 0.7, the agent considers the cooperation important for the current data gathering session and decides consequently to cooperate

Table 2 Agent strategy equation parameters

Agent Strategy Parameters	**Values**
Threshold of R	0.7
w_e,w_p, w_i	0.3
w_d	0.1

6. PROPOSAL EVALUATION

In order to evaluate the performance of SBC, we define, in this section, different performance criteria that we will use later in order to compare SBC with other approaches.

6.1. Performance criteria

In this section, we present the main performance criteria and the base of their evaluation through our simulations:

1. Energy consumption is the parameter that defines the life duration of a sensor node and consequently of the concerned wireless sensor network. Therefore, we consider this parameter as the most important criterion to evaluate the performance of our proposal. In our simulations, we compute the average value of power consumed by each node. This value will be composed of three main parts:

    - The communication entity of the sensor node: It is the most energy-intensive function in the node. In order to compute the amount of energy consumed in communication, we use the equation (4) defined in [24]. E_{TX} is the power consumed during the transmission and E_{RX} is the power consumed during the reception. Both of them are computed following the data length and transmission distance (radio range of the node) (l, d);

    $$\begin{aligned} E_{TX}(l,d) &= lEc + led^{s} \\ E_{RX}(l,d) &= lE_{c} \end{aligned} \quad \text{where e=} \begin{cases} e_1 & s=2,\ d<d_{cr} \\ e_2 & s=4,\ d>d_{cr} \end{cases} \qquad (4)$$

    Where E_c is the base energy required to run the transmitter or receiver circuitry. A typical value of E_c is 50nJ/bit for a 1-Mbps transceiver; d_{cr} is the crossover distance, and its typical value is 87m; e_1 (e_2 respectively) is the unit energy required for the transmitter amplifier when $d < d_{cr}$ (or $d > d_{cr}$ respectively). Typical values of e_1 and e_2 are 10pJ/bit.m^2 and 0.0013pJ/bit.m^4, respectively.

    - The energy consumed by the CPU: To compute this energy, authors in [24] have defined a power consumption estimation method based on the number of instructions and the frequency of the processor. In SBC, we use the processor defined in the sun spot technical document [25], where the processor frequency of the sensor nodes is equal to 180 Mhz. According to [24], a processor with such frequency consumes approximately 0.8 nJ per instruction. For this parameter, we would like to underline that for the CS approach, the power consumption of the CPU is neglected as the information processing is realized in the sink and not in the sensor node;

- The power consumption of the sensing action: It is the lowest amount compared to the above described sources of expenditure [24]. This power utilization is supposed to be the same for SBC, DF and CS approaches. Therefore, we have decided to neglect it in our simulations.

2. The average end-to-end delay is an important criterion. However, it is heavily dependant on the application itself. In the present work, this parameter represents the average latency needed to carry a message from a source sensor node till the sink. This delay computation is applicable to the CS approach as all the processing is done in the sink node. However, in DF and in SBC, the average end-to-end delay includes the local processing time. In DF, the local processing time is used to process data to merge the information of the nodes, while in SBC it is needed for estimating the importance of data and cooperating with neighbouring agents;

3. The packet delivery ratio is the ratio of data packets received by the sink node to the number of packets generated by the source nodes (a node that has information to send). Hence, in SBC, the source nodes are the nodes that start a data gathering cooperation, which means that the number of messages sent by source nodes is equal to the number of data gathering cooperation sessions;

4. The saved overhead: This parameter emphasizes the importance of the data concatenation presented in section 3. It defines the average number of messages' headers needed to carry out the information of n-source nodes. In the traditional CS architecture, we need one message header to send the information of one node to the sink, while in SBC and in DF approach, we need one message header to carry out the information of a certain number of nodes;

5. The average power consumption of nodes in critical positions has a particular importance, as presented previously in section 4.3. This criterion is presented to underline the risk that the network may face if the power consumption of these nodes is not well managed. To evaluate this criterion, we extract the nodes in critical positions from the simulated topology, and then we compute the average of their consumed power.

6.2. Results analysis

In this section, we present the simulation results to highlight the relevance of our proposal (SBC). We show the advantages of SBC by comparing it to the CS communication architecture and the DF approaches. In addition, to understand the performance of the agent strategy, we compare SBC to our previous work [17] Information Importance Based Communication (IIBC) where we did not use the agent strategy.

We focus mainly on the efficiency of SBC in terms of power consumption and scalability in different network densities. As presented in the simulation setup section, we have varied the number of nodes from 100 to 900.

In Figure 7a, we can observe that compared to CS and DF, SBC highly decreases the power consumption. In addition, it is clear that the saved power is more important for the higher number of nodes. These results prove that SBC is significantly better designed for large scale or dense networks than the CS and the DF approaches. Indeed, for a number of nodes varying from 100 till 900, the power consumption obtained by using SBC is in average reduced by a factor of 9 and 6 compared to CS and DF respectively, which means a huge amount of saved power.

In addition, we compare, in Figure 7b, SBC to our previous proposal (IIBC), which follows a similar data gathering scheme where the agent makes its cooperation decision based only on the importance of the information in place of the agent strategy presented in section 4. We can point out that by applying the SBC strategy, we divide the consumption by 1.5, approximately. In addition, by observing the curves, we can see that SBC is more scalable as the values between 700 and 900 are approximately stable, while the slope is bigger in IIBC.

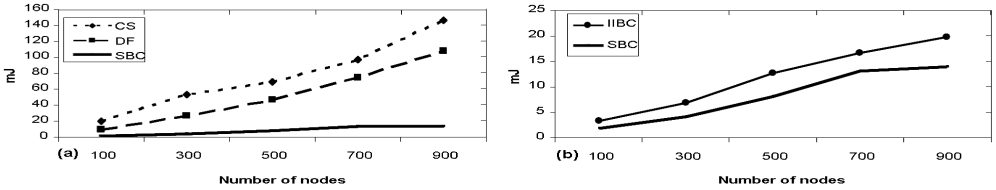


Figure 7 Average power consumption per node

In the next parts, we will not compare explicitly IIBC to SBC, in terms of end-to-end delay (Figure 8) and packet delivery ratio (Figure 9), as they have similar performance for these criteria. This similarity is due to the resemblance of both approaches in terms of data gathering. We recall here that IIBC is one of our previous works [17].

In Figure 8, we show the average end-to-end delay in the network. The presented curves point out an extra latency in SBC compared to DF and CS. This latency is related to the local processing time and the agent cooperation needed in SBC. We could see also that the difference of latency between SBC and DF is less compared to the difference of latency between SBC and CS as DF spends extra latency to merge the information, cooperate with other agents and gather other sensor nodes' data, which is not the case for CS that sends directly the perceived data to the sink.

This latency allows SBC to save more energy (Figure 7) and more overheads (Figure 10). In addition, by observing the results, and by comparing SBC to CS, which presents the greater difference, we can see that the different values, obtained by using SBC and the CS approach, are lower than 0.3 second. This means only the applications, which are very sensitive to latency and require less than 0.3 second precision, could be influenced.

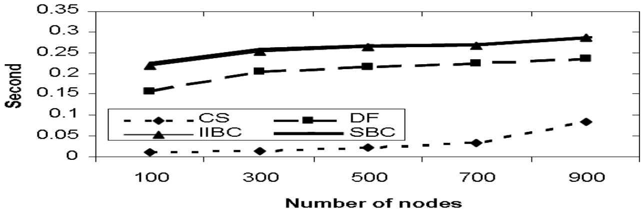


Figure 8 Average end-to-end delay

On the other hand, it is clear that the average end-to-end delay in SBC is approximately stable and follows a low raising curve for a number of nodes varying from 300 to 900. That means the network density does not really influence the end-to-end delay and the scalability could be supported easily in terms of latency. Moreover, these differences could be easily explained by looking to the saved overheads results (Figure 10), which will be discussed later.

In Figure 9, we plot the packet delivery ratio. We can see that the packet delivery ratio decreases when the density of the network increases. This is due to the fact that the main source of loosing packets is the collisions. Furthermore, the number of messages sent in the network and the probability of collision are higher for denser networks. Nevertheless, the packet delivery ratio in SBC is, as shown in Figure 9, always close to 100%. This high level of packet delivery ratio could be explained by the fact that in SBC there is just one hop communications. Indeed, the initial (or intermediate) node sends a cooperation request to only its one-hop neighbours. Next, these neighbours respond to the initial (or intermediate) node by one hop communication.

Finally, the initial (or intermediate) node sends the cooperation message to its first node on the path to the sink. Thus, we eliminate the problems of congestion and collision in a multi-hop communication. In DF, the packet delivery ratio is also close to 100% as this approach in terms of communication is similar to SBC. Indeed, each node sends its information to the next hop towards the sink that merges its data with the previous sensor node data before sending the message to its next hop. Hence, in DF, there are several one hop communications before reaching the sink, which eliminates the packet loss caused by collisions.

This result means that SBC and DF are scalable and tolerate the high density network. However, CS is sensitive to the density of the network as the value of packet delivery ratio decreases when the number of nodes increases.

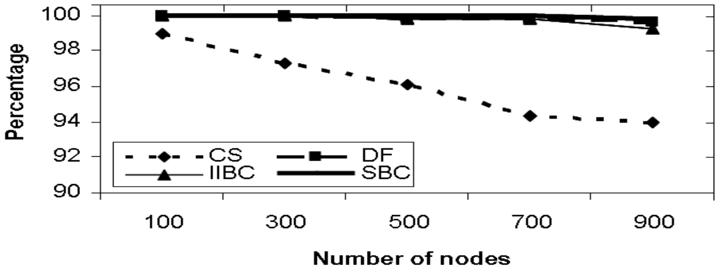


Figure 9 Packets delivery ratio

The saved overhead criterion helps to better explain the results obtained in terms of power consumption and end-to-end delay.

As we can observe in Figure 10, for a number of nodes equal to 100 (non-dense network), we have around 14 sensor-nodes information sent in only one cooperation message. In terms of power consumption, this result means that in SBC we send one message with one header instead of sending 14 messages with 14 headers in the CS approach or 4 messages with 4 headers in the DF approach. By referring to Figure 7a, we can notice that the power consumption in CS, DF and SBC was around 20 mJ, 9 mJ and 2 mJ, respectively, which gives a ratio of 10 between CS and SBC and 4.5 between DF and SBC. These ratios mean that for a non-dense network, SBC allows us to send the same amount of significant data with a power consumption divided by 10 or 4.5 comparing to the CS and the DF approaches, respectively. For a denser network (900 nodes), the values obtained depict also a real gain in term of energy.

The saved overhead explains also the end-to-end delay occurred in SBC. By looking at the values obtained in Figure 8, we can consider that SBC sends the information of 14 sensor nodes processed in around 220 ms, while the CS approach requires 9 ms to send one message of non processed information and DF requires 157 ms to send only 4 messages of processed information. Firstly, by comparing the time that the sink node takes to compute 14 messages and their communication time to the 220 ms required in SBC, we could estimate that there is not a significant latency difference between SBC and the CS approaches. Secondly, we could point out that 157 ms is required to carry out the information of 4 sensor nodes in DF while 220ms is sufficient for 17 sensor nodes information.

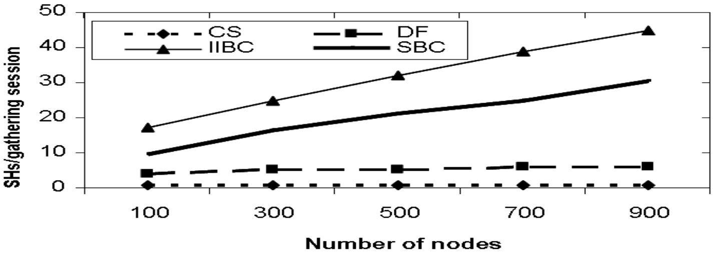


Figure 10 Number of saved headers per gathering session

In Figure 10, we can mention that IIBC saves more headers than SBC. That is due to the average number of agents cooperating in each data gathering session. In IIBC, an agent makes its cooperation decision according to the importance of its detected information only. However, in SBC, in addition to this latter, the decision will be based on several other parameters, as explained in section 4. Therefore, in SBC, an agent may refuse to cooperate, even if it has important information, in order to optimize other criteria, e.g. to maintain the connectivity of the network as presented in Figure 11. Hence, in IIBC, we have more cooperative agents and so more saved headers, while SBC optimizes other criteria, e.g. power management of nodes in critical positions, by reducing the cooperativity of some agents.

Figure 11 compares the average power consumption per node for the nodes in critical positions in the four approaches. Hence, this criterion underlines the network partitioning possibility. More the power consumption of these nodes is higher, less is the mean time to first network partition.

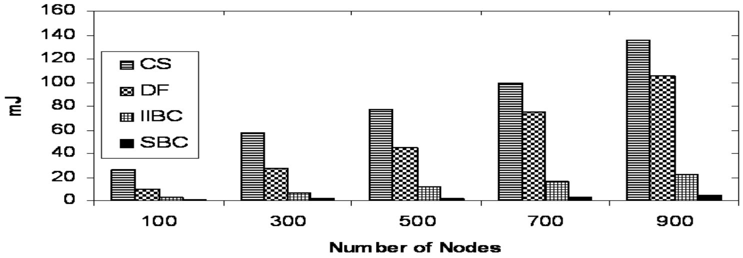


Figure 11 Average power consumption per node in critical position

As we can mention, in Figure 11, SBC decreases the average power consumption of these nodes in an important manner. It also shows that more the network is dense more the amount of decreased power is important. We can also see that for 700 and 900 nodes, SBC divided by more than 20 the consumption of these nodes, compared to CS and DF. In addition, SBC reduces approximately 4 more times the consumption of this kind of nodes, compared to IIBC. Hence, we can deduce from these curves that the agent strategy offers better power management for nodes in critical positions, independent of the network scale and density. Moreover, these results prove the importance of applying a selfish behaviour to this kind of nodes, which allows to maximize the time to the first partitioning in the network.

7. CONCLUSION AND FUTURE WORKS

In this paper, we have presented a data gathering scheme based on multi-agent for wireless sensor networks. The presented scheme is built by the implementation of an agent per node. Each agent treats locally the sensed information of its correspondent sensor node and estimates its importance. Then, it cooperates with its neighbouring agents to gather their information and to eliminate the inter-sensor-nodes redundancy. A message containing the concatenation of all the processed and non-redundant information, results from the agents' cooperation. Each intermediate node, which relays the message to the sink, repeats the same scenario and adds its information to the same message.

This scheme limits the communications to only the important information, decreasing as a result the amount of traffic and the power consumed.

The presented scheme is also based on a strategy that takes into consideration several parameters judged as important for long life WSN. From these parameters, we distinguish the position of the sensor node within the network, which could be critical when, for example, the sensor node relays two parts of the network. Hence, if this node runs out of battery, it divides the network.

Successive simulations in large scale and different WSN densities proved that our scheme called SBC (Strategy Based Communication) compared to the CS and the DF approaches had reduced, heavily, the average power consumption of the sensor nodes wile maintaining a high packet delivery ratio. A justified drawback has appeared in terms of delay. This extra latency is related to the local processing of the information and to the cooperation between agents, which allowed the elimination of the inter-sensor-nodes redundancy. In addition, An important overhead has been saved by carrying the information of n-nodes with only one message header. The results show also that SBC well manages the nodes in critical positions by minimizing their power consumption to the less possible, in order to avoid the network partitioning.

As a future work, we think that a mathematical model could be interesting in order to study the possibility of using a variable cooperation relevance threshold. Then, the agent strategy will be studied in the case of multiple applications over the same physical WSN. We also aim to manage entire sensor node protocol stack by agent approach, in order to realize a better energy-efficient data gathering.

REFERENCES

[1] Ferber, J.: Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Harlow: Addison Wesley Longman (1999)

[2] Chen, M., Kwon, T., Yuan, Y., Leung, V.C.M.: Mobile Agent Based Wireless Sensor n-Networks. Journal of Computers, 14–21 (2006)

[3] Chou, J., Petrovic, D., Ramchandran, K.: Tracking and exploiting correlations in dense sensor networks. Conference Record of the Thirty-Sixth A silomar Conference on Signals, n-Systems and Computers. 39–43 (2002)

[4] Chou, J., Petrovic, D.,Ramchandran, K.: A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. In:Twenty-Second Annual, Joint Conference of the IEEE Computer and Communications Societies. 1054–1062 (2003)

[5] Das, S.R., N.A.P.S.: Serial Data Fusion Using Space-filling Curves in Wireless Sensor Networks. In IEEE, ed.: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks.182–190 (2004)

[6] Liao, W.H., Kao, Y., Fan, C.M.: Data aggregation in wireless sensor networks using ant colony algorithm. J. Netw. Comput. Appl. 31(4) 387–401 (2008)

[7] Chen, H., Mineno, H., Mizuno, T.: Adaptive data aggregation scheme in clustered wireless sensor networks. Comput. Commun. 31(15) 3579–3585 (2008)

[8] Lang, T., Qing, Z., Srihari, A.: Sensor networks with mobile agents. In: The IEEE Military Communications Conference, place City, Boston, IEEE 688–693 (2003)

[9] Qi, H., Xu, Y., Wang, W.: Mobile-agent-based collaborative signal and information processing in sensor networks. Proceedings of the IEEE 91(8) 1172–1183 (2003)

[10] Al-Karaki, J.N., Ul-Mustafa, R., Kamal, A.E.: Data aggregation and routing in wireless sensor networks: Optimal and heuristical gorithms. Comput. Netw. 53(7) 945–960 (2009)

[11] Upadhyayula, S., Gupta, S.K.S.: Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced converge cast (dac) in wireless sensor networks. AdHoc. Netw. 5(5) 626–648 (2007)

[12] Jie, S., Ning, Y.: A spanning tree algorithm for data aggregation in wireless sensor networks. In:WCICA2008, IEEE explorer 5014–5018 (2008)

[13] BoYu, J., Li, Y.: Distributed data aggregation scheduling in wireless sensor networks. In: IEEE INFOCOM, IEEE explorer (2008)

[14] Fan, K.W., Liu, S., Sinha, P.: Structure-free data aggregation in sensor networks. IEEE Transactionson Mobile Computing 6(8) 929–942 (2007)

[15] Rajagopalan, R., Varshney, P.K.: Data aggregation techniques in sensor networks: A survey. Comm. Surveys&Tutorials, IEEE 8 48–63 (2006)

[16] Akyildiz, I.F., Su, W., Sankarasubramaniam,Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38 393–422 (2002)

[17] Sardouk, A., Rahim-Amoud, R., Merghem-Boulahia, L., Gaïti, D.: Information-Importance Based Communication for Large-Scale WSN Data Processing. In: WMNC, Gdansk, Poland, IFIP (2009)

[18] Sardouk, A., Merghem-Boulahia, L., Gaïti, D.: Agent-Cooperation Based Communication Architecture for Wireless Sensor Network. In: IFIP/IEEE Wireless Days/Adhoc and Wireless Sensor Networks, UAE, IFIP (2008)

[19] Brooks, R.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation 2(1) 14–23 (1985)

[20] Rahim-Amoud, R., Merghem-Boulahia, L.,Gaïti, D.: An Autonomic MPLS DiffServ-TE Domain.Whitestein Series in Software Agent Technologies and Autonomic Computing 9 149–16821. (2008)

[21] Sardouk, A., Rahim-Amoud, R., Merghem-Boulahia, L., Gaïti, D.: Multi-Agent system for Multi-Application WSN. In: ManWeek/MMNS, Venice, Italy, IEEE 183–188 (2009)

[22] Chen, M., González-Valenzuela, S., Leung, V.C.M.: Applications and design issues of mobile agents in wireless sensor networks.IEEE Wireless Communications Magazine 14(6) 20–26 (2007)

[23] U.P.C., W.A.M.: Laboratory, Glomosim: A scalable simulation environment for wireless and wired network systems. In: The 3rd International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks.(2005)

[24] Sohraby, K., Minoli, D., Znati, T.: Wireless Sensor Networks, Technology, Protocols, and Applications. WILLEY (2007)

[25] Sun: SunTM Small Programmable Object Technology (SunSPOT) Theory of Operation. Technical report, Sun Microsystem, SunLabs (2008

Authors

Ahmad SARDOUK received the Master of Networking, Knowledge and organisation from the University of Technology of Troyes, France, in 2007. Currently, he is a PhD student at the University of Technology of Troyes, France. His research interests include agent cooperation, energy efficiency, data management, and routing in wireless sensor networks.

Rana RAHIM-AMOUD received with distinction her Computer science and Telecommunication Engineer degree from the Lebanese University - First Branch, Tripoli, Lebanon, in 2002. She then obtained her Master degree (DEA) in Networking and Telecommunication in 2003 from the USJ University and the Lebanese University, Beirut, Lebanon, and her PhD degree in January 2008 from the University of Technology of Troyes (UTT) - France. She was a postdoctoral researcher at the UTT from October 2008 to October 2009. She is currently a researcher and teaching assistant at the Lebanese University. Her research interests include Networking, System management, Quality of Service, MPLS Networks, WSN Networks, IoT Networks, Adaptable and Autonomous systems.

Leïla Merghem-Boulahia is an assistant professor at the Université de Technologie de Troyes (UTT) in France still 2005. She received her M.Sc. and her Ph.D. from the University of Paris 6, France, in 2000 and 2003, respectively. Her main research topics include behavioural multi-agent simulation, quality of service management, and the intelligence in networks.

Dominique Gaiti received the Ph.D. and the "Habilitation à diriger des recherches" degrees in Computer Science from the University of Paris VI and Paris IX on 1991 and 1995 respectively. She is currently a professor at the University of Technology in Troyes (France), a member of the Institute Charles Delaunay (ICD). She is the leader of the team “autonomic networking” in this institute. Before she was a research scientist at the University of Columbia (New York-USA), 1992-1994 and a researcher at the University of Paris 6, member of the LIP6 laboratory (Paris - France), 1996-1997. She is the chairperson of the IFIP WG 6.7 on "smart networks". Her research interests include the smart networks, the intelligence in networks, and the control and management (through intelligent agents) in all types of networks. She is the author of one book and has edited several proceedings.