Information-Importance Based Communication for Large-Scale WSN Data Processing

Ahmad Sardouk, Rana Rahim-Amoud, Leïla Merghem-Boulahia, and Dominique Gaïti

ICD/ERA, FRE CNRS 2848, Université de Technologies de Troyes, 12 rue Marie Curie, 10010 Troyes Cedex, France
{ahmad.sardouk,rana.amoud,leila.boulahia,dominique.gaiti}@utt.fr

Abstract. Gathering information in an energy-efficient and scalable manner from a wireless sensor network is always a basic need. In this work, we use the multi-agent approach in order to build an Information-Importance Based Communication for large scale wireless sensor network data processing. The principal goal of our proposition is to tackle the problem of network density and scalability in an energy efficient manner. Simulation results are provided to illustrate the efficiency of our proposition.

1 Introduction

Due to advances in wireless communications and electronics over the last few years, the development of networks of low-cost, low-power, and multifunctional sensors has received increasing attention. These sensors are small in size and integrating the capabilities of sensing, computing and communication. Large amounts of these small-size and low cost sensor nodes can be rapidly deployed in a region of interest and form a loosely-coupled distributed networking system called Wireless Sensor Network (WSN). The main goal of this network is to collect information from the target environment. The WSN is 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 comparing to its communication. This is especially due to the fact that a lot of the sensed information could be redundant or not important.

In this paper, we propose a novel scheme for data gathering in wireless sensor networks. This scheme does not only reduce heavily the power consumption but it also tackles the density of the network and the scalability problems. This new scheme is based on the multi-agent approach in order to build a Communication structure Based on the Importance of the Information (IIBC). In addition, a multi-agent cooperation is used to gather the processed information from multiple sensor nodes with an inter-sensor-nodes redundancy elimination.

The paper is organized as follows. In section 2, we give a brief description of the multi-agent systems. In section 3, we review the related work. Our solution is described in section 4. Next, in section 5, we present our simulation setup parameters and the performance criteria. Then, in section 6, we evaluate and analyze the performance of our proposition. Finally, the conclusion is given in section 7.

2 Multi-Agent Systems: Some definitions

According to [1], an agent is a physical (robot) or virtual (real time embedded software) entity having trends and resources, able to perceive its environment, to act on it and to acquire a partial representation of it (called the local view of the agent). An agent is also able to communicate with other peers and devices and has a behavior that fits its objectives according to its knowledge and capabilities. Furthermore, an agent can learn, plan future tasks and is able to react and to change its behavior according to the changes in its environment. A multi-agent system (MAS) is a group of agents able to interact and to cooperate in order to reach a specific objective. Agents are characterized by their properties that determine their capabilities. Different properties are defined like autonomy, proactiveness, flexibility, adaptability, ability to collaborate and to coordinate tasks and mobility. According to its role within its environment, the agent acquires some of these properties. Multi-agent approach is well suited to control distributed systems. WSN are good examples of such distributed systems. This explains partly the considerable contribution of agent technology when introduced in this area.

3 Related works

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 lifetime longevity of such network, the density of the network, and its scalability. The traditional and base model of data gathering is the client/server (C/S) communication architecture. In this architecture, when the sensors of a sensor node perceive information from their environment, they send it directly as it is (raw not processed) to the sink to be processed. Moreover, to send information to the sink, the communication passes through a multi hop communication. That means, intermediate nodes relay farther nodes information. Consequently, this relaying causes supplementary power consumption for this node. Several works have been done to optimize this traditional architecture.

In [2], the authors proposed to merge the information of a maximum number of nodes. Thus, they proposed a serial incremental fusion 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 have always important information to send and they do not eliminate the unimportant or redundant information. Furthermore, this solution is only suitable to some scenarios. A scenario, showing that this solution is not scalable, is when a node, which is at

100 hops from the sink, sends it a message. In this case, the size of the message will be incremented, in each hop, by the size of the current message hop. For example, in the middle of the network, the size will be approximately equal to the original message size multiplied by 50. Such situation (message with huge size) may consume a lot of power and causes a lot of loss in term of packets.

The authors of [3] present an ant colony based data aggregation for wireless sensor networks. They are trying 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 data aggregation tree is constructed by the accumulated pheromones. By exploring all possible paths to sink, each ant consumes extra power that could be eliminated. Furthermore, the construction of the appropriate tree depends heavily on the nodes' deployment, which is generally random. Such tree construction consumes an important amount of power.

In [4], the authors propose an adaptive data aggregation (ADA) scheme for clustered WSN. The authors define temporal and spatial aggregation degrees, which are controlled by the reporting frequency at sensor nodes and the aggregation ratio at cluster heads respectively. The temporal and spatial aggregation degrees are determined by the current scheme state according to the observed reliability. Their work focuses principally on the incorporation of an adaptive behavior into protocols in such dynamic network. ADA is based on the cluster heading paradigm, which needs an expensive construction in term of energy. Furthermore, in the implementation of this scheme, the authors did not address the needed power consumption which we suppose important. In addition, authors neglected the importance of scalability of such kind of networks.

The authors of [5] have been oriented towards the agent technologies to gather the information. The central work of these authors was based on the use of Mobile Agents (MAs) in WSN for an energy-efficiency data gathering. In these propositions, 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 an important 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. This solution reduces the power consumption in cases of 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 node 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 lake 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 region which will be treated by the MA.

After analyzing the previously presented solutions, we can see clearly that there is a serious problem in term of scalability for all of them. In addition, we can 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 topology. It should be efficient for a randomly deployed network.

4 A Multi-Agent Cooperation for Energy-Efficiency Data Gathering

According to [6], a sensor node expends a maximum energy in data communication where the energy expenditure in data processing is much less compared to it. The energy cost for the transmission of 1KB on 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 so to extend the lifetime of the network:

1. The Information Importance Based Communication: 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 where we monitor a stable environment. Hence, 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.
2. The Elimination of Non Important inter-sensor-nodes Information: generally, sensor nodes are deployed randomly (e.g., a plane throws them in hazards zones). Thus, two or more sensor nodes can cover the same point and trivially they will give always the same information (inter-sensor-nodes redundancy).
3. The Data Concatenation: Due to protocol overheads, the communication cost, in term of energy, of sending a long message is usually less than the one of sending the same amount of data using many short messages.

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

4.1 System Design

A sensor node has generally a battery, a processing unit, a memory and a radio entity. 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 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 an MAS to bring up the full autonomy to WSNs. As we presented in section 2, an agent is by definition capable of making decisions and of cooperating with other agents to execute tasks. To illustrate our proposal, we use the architecture presented in fig. 1, where 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 neighbor agents in order to eliminate the inter-sensor-nodes redundancy and concatenate the processed information of other sensor nodes. In our previous work [7] [8], we have defined some mechanisms to process the information and to estimate its importance. These mechanisms will be enhanced and used in our present work. Our 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 neighborhood. That is why, in our proposition, the local view of the agent will be restricted to its one hop neighbors 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. We would like to underline here that an agent will never know the whole WSN architecture as this is cost expensive in term of communication and non important to achieve the agent goal.

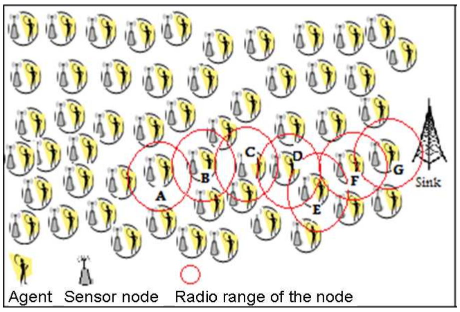

Fig. 1. Network topology

4.2 Agent Knowledge Base

One of the basic attributes of an agent is to be situated (situadness [9]). That is, an agent is a part of its environment. Its decisions are based on what it perceives of this environment and on its current state [10]. The situated view of an agent is then composed of the information obtained from its local observation and the information exchanged with its neighbors. For that, we should define carefully the required information for the agent to achieve its mission. This information will be stored in a data base, which we call a knowledge base. The knowledge base contains the list of the one hop neighbor 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.

This information could be found through the routing level. At this level, our proposal is based on the Dynamic Source Routing protocol [11]. This protocol uses flooding technique to discover the route from a source to a destination. During the route discovery, network nodes build their routing tables from the information existing in the route request messages. For a computer network or a non limited resource network, there is no problem to save all of this information. However, in the case of WSN, we should pay attention to the memory size. In our proposal, as we said previously, the agent needs only the knowledge based on its situated view. Therefore, the agent filters the information received from the route request messages, saves only what interests it and keeps out the others.

4.3 Gathering Session Scenario

The information gathering session starts when an agent (sensor node) detects important information. This agent invites its one hop neighbors to cooperate in order to gather the maximum possible of information and to create a cooperation message resuming these collected information. However, the neighbor 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 response the cooperation request. Indeed, once the cooperation message is ready, this neighbor agent (called intermediate agent) will receive the message and will invite its one hop neighbors' agents to cooperate. The intermediate agent will gather the information of its one-hop neighbors and extend the initial cooperating message. This message will be then sent to the next intermediate agent. The new intermediate agent will, in its turn, repeat the same scenario. This scenario will be repeated until reaching the sink node.

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

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 estimated this information as important. Consequently, agent A sends a cooperation request to its one hop neighbors. The communication of this request passes by a one hop broadcast. Indeed, the one hop neighbors will be programmed to not re-broadcast it.

By sending the cooperation request, agent A invites its one hop neighbor agents to join cooperation for a data gathering session. A neighbor agent decides to cooperate if it has also important information

After taking the appropriate decision, each cooperating agent responses by sending its processed data. This data will be concatenated within a cooperation message after an inter-sensor-nodes redundancy elimination. A model of this cooperation message has been previously presented in one of our previous works [8]. This message contains two main parts. The first one is for the sensor nodes' addresses and the second one is reserved for the correspondent data.

Agent A sends the cooperation message to its first hop on the path to the sink which is agent B (fig.1 1). As agent B is a one hop neighbor for agent A, it has previously received the cooperation request sent by A. Agent B did not response this request as it knows 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 neighbors to gather their data. If agent A and agent B have common one hop neighbor(s) agent(s), the common agents receive two cooperation requests but answer only the first request and neglect the second one.

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

5 Simulation Setup

In order to evaluate the relevance of our proposal (IIBC), we have carried out a set of simulation tests. In this section, we define the simulation parameters in order to demonstrate the performance of IIBC by comparing it to the traditional client/server (C/S) communication architecture. We have implemented these two approaches on GlomoSim [12] which is a scalable simulation environment for wireless and wired network systems.

We have run our simulation 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 [13]. The size of sensed data is 24 byte per node and the sensed data interval is 10 seconds. The transmission and reception powers parameters, which influence directly the radio range, have been chosen carefully from the ranges defined in the sun spot system technical document [14].

In order to test the scalability and the relevance of IIBC 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 inspired from the work realized in [5], where the processing code is put in a message (mobile agent) sent by the sink. Indeed, transferring this code from the message to the node and placing it in the appropriate place of the memory will take some time. We have estimated this time to 10 ms. The authors of [5] have fixed the processing time to 50 ms which means that 40 ms will be sufficient in IIBC.

5.1 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 IIBC. In our simulations, we compute the average value of power consumed by each node which is composed of three main parts:

- The communication entity of the sensor node, which is the most energy-intensive function in the node. In order to compute the amount of energy consumed in communication, we use the equation (1) defined in [13]. E_{TX} is the power consumed during transmission and E_{RX} is the power consumed during the reception. Both of them are computed following the data length and distance of transmission (radio range of the node) (l,d);

$$\begin{array}{l} E_{TX}(l,d) = lE_c + led^s \\ E_{RX}(l,d) = lE_c \end{array} \quad \text{where} \quad \text{e=}\{ \begin{array}{lll} e_1 & s = 2, & d < d_{cr} \\ e_2 & s = 4, & d > d_{cr} \end{array} \tag{1}$$

  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 (or e_2) is the unit energy required for the transmitter amplifier when d $< d_{cr}$ (or d $> d_{cr}$). 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 [13] have defined a rule based on the number of instructions and the frequency of the processor. In IIBC, we use the processor defined in the sun spot technical document [14], which defines the processor frequency of their sensor nodes to 180 Mhz. According to [13], a processor with such frequency consumes approximately 0.8 nJ per instruction. For this parameter, we would like to underline that for the C/S approach the power consumption of the CPU is neglected as the information processing occurred in the sink and not in the sensor node;
- The consumption of the sensing action: we neglected this consumption as it is supposed to be the same for IIBC and C/S approach.

2. The average end to end delay is an important criterion. This parameter represents the average latency needed to carry a message from a source sensor node till the sink. This delay computation is applicable as it to the C/S approach as all the processing is done in the sink node. However, in IIBC, the average end to end delay includes the local processing time (needed for estimating the importance of data and cooperating with neighbor 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. In IIBC, 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. It defines the average number of saved messages' headers needed to carry out the information of n-source nodes. In the traditional C/S architecture, we need one message header to send the information of one node to the sink, while in IIBC; we need one message header to carry out the information of n-cooperating nodes.

6 Results and Analysis

In this section, we present the simulation results to highlight the relevance of IIBC. We show the advantages of IIBC by comparing it to the traditional C/S approach. We have chose to compare IIBC to the C/S approach as it is the main base for all the propositions and actually it is the most deployed one. We focus mainly on the efficiency of IIBC in terms of power consumption and scalability in different network densities. As presented in the simulation setup section.

In fig. 2 a, we can observe that IIBC 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 IIBC is significantly better designed for scalable or dense networks than the C/S approach. Indeed, for a number of nodes varying from 100 till 900, the power consumption obtained by using IIBC is in average reduced by a factor of 7.11. In fig. 2 b, we show the average end-to-end delay in the network. We see that the values when IIBC is employed are higher than those given by the C/S approach.

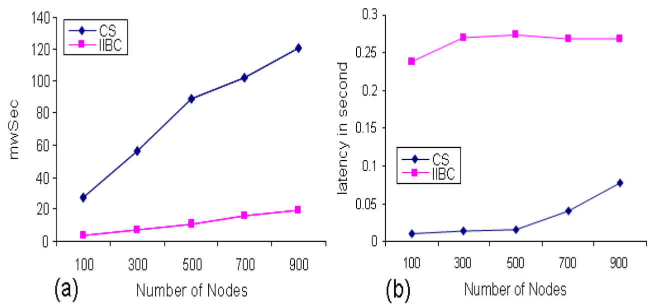

Fig. 2. a) Average Power Consumption per Node, b) Average End-to-End Delay

This gives the false impression that the performance of IIBC in term of end-to-end delay is bad. However, this is not correct, as the end-to-end delay in the C/S approach does not include the processing time. If we add this processing time to the end-to-end delay, we think the difference with IIBC will be really reduced. In fact, by observing the results, we can see that the different values, obtained by using IIBC and the C/S 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. In the other hand, it is clear that the average end-to-end delay is approximately stable and follows a straight curve for a number of nodes varying from 300 to 900. That means the network density does not influence the end-to-end delay and the scalability could be supported easily in term of latency. Moreover, this difference could be easily explained when discussing later the saved overhead results (fig. 4).

In Fig. 3, we plot the packet delivery ratio (percentage of received packets). We see that the packet delivery ratio decreases when the density of the network increases. This is attributed to the fact that the main source of loosing packets is the collision. Furthermore, the number of messages sent in the network and the probability of collision are higher for the denser networks. Nevertheless, the packet delivery ratio in IIBC is always close to 100%. This high level of packet delivery ratio in IIBC could be explained by the fact that we have just one hop communications. Indeed, the initial (or intermediate) node sends a cooperation request to only its one hop neighbors. Next, the neighbors respond to the initial (or intermediate) by a 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 eliminated the problems of multi hop communication in term of packets loss.

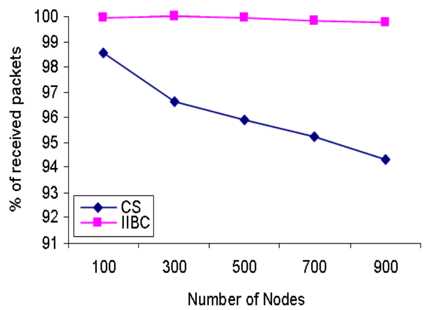

Fig. 3. Percentage of received packets

This result means that IIBC is scalable in term of packet delivery ratio. However, the C/S approach is sensitive to the density of the network as the values of packet delivery ratio decrease when the number of nodes increases. Thus, the C/S approach is not scalable regarding the packet delivery ratio. It should be noted here that the fact of having high values in both solutions is due to the use of a relatively big sensing interval (10s), which is the case of monitoring applications.

The last performance criterion is the saved overhead. This criterion has a special particularity as it helps to better explain the results obtained in term of power consumption and end-to-end delay. As we can observe in fig. 4, for a number of nodes equal to 100 (non dense network), we have around 17 sensor-nodes information sent in only one cooperation message. In term of power consumption, this result means that in IIBC, we send one message with one header instead of sending 17 messages with 17 headers in the C/S approach. By referring to fig. 2 a, we can notice that the power consumption in the C/S approach and in IIBC was around 27 mw and 3 mw respectively, which is a ratio of 9. This ratio means that for a non dense network, IIBC allows us to send the same amount of significant data with a power consumption divided by 9 comparing to the C/S approach. For the denser networks, the values obtained show a real gain in term of energy.

The saved overhead explains also the end-to-end delay occurred in IIBC. By looking to the values obtained in fig. 2 b, we can consider that IIBC sends the information of 17 sensor nodes processed in around 250 ms, while the C/S approach requires 10 ms to send one message of non processed information. By comparing the time that the sink node takes to compute 17 messages and their communication time to the 250 ms required in IIBC, we could estimate that there is not a real latency difference between IIBC and the C/S approach.

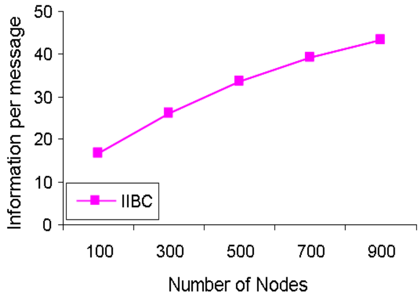

Fig. 4. Saved overhead for each cooperation

7 Conclusion and future works

In this paper, we have presented a new information importance based communication, scalable and energy efficient approach for data gathering in wireless sensor networks (IIBC). This approach is based on a multi-agent cooperation solution, in order to gather the maximum possible of important information in each data gathering session. By cooperating, the agents eliminate the inter-sensor-nodes redundancy and concatenate all the processed information of one gathering session into one message. That means a gain in the amount of information and the overhead needed to send them. We have proved through successive simulations that IIBC is very scalable comparing to the client server approach. Indeed, IIBC consumes little amount of energy, justified latency by taking into consideration the gain in overhead, and has a very high packet delivery ratio even if the network is dense. In the future work, we are intended to define an agent strategy to make a more appropriate decision to cooperate or not taking in consideration several important parameters (the available power in the node, its position within the network, etc.)

References

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

2. 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. (2004) 182–190
3. Liao, W.H., Kao, Y., Fan, C.M.: Data aggregation in wireless sensor networks using ant colony algorithm. J. Netw. Comput. Appl. **31**(4) (2008) 387–401
4. Chen, H., Mineno, H., Mizuno, T.: Adaptive data aggregation scheme in clustered wireless sensor networks. Comput. Commun. **31**(15) (2008) 3579–3585
5. Chen, M., Kwon, T., Yuan, Y., Leung, V.C.M.: Mobile Agent Based Wireless Sensor Networks. Journal of Computers (2006) 14–21
6. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks **38** (2002) 393–422
7. Sardouk, A., Merghem-Boulahia, L., Gaiti, D.: Agents Cooperation for Power-Efficient Information Processing in Wireless Sensor Networks. In: Networking and Electronic Commerce Research Conference, Italy (2008)
8. Sardouk, A., Merghem-Boulahia, L., Gaiti, D.: Agent-Cooperation Based Communication Architecture for Wireless Sensor Network. In: IFIP Wireless Days/Ad-hoc and Wireless Sensor Networks, UAE, IFIP (2008)
9. Brooks, R.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation **2**(1) (1985) 14–23
10. Rahim-Amoud, R., Merghem-Boulahia, L., Gaiti, D.: An Autonomic MPLS DiffServ-TE Domain. Whitestein Series in Software Agent Technologies and Autonomic Computing **9** (2008) 149– 168
11. Johnson, D., Hu, Y., Maltz, D.: The Dynamic Source Routing Protocol (DSR) for placeMobile Ad Hoc Networks for IPv4. Technical report (2007)
12. Laboratory, U., Laboratory, W.: 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)
13. Sohraby, K., Minoli, D., Znati, T.: Wireless Sensor Networks, Technology, Protocols, and Applications. WILLEY (2007)
14. Sun: Sun^{TM} Small Programmable Object Technology (Sun SPOT) Theory of Operation. Technical report, Sun Microsystem, Sun Labs (2008)