A Coordination Protocol for Target Tracking in Wireless Sensor Networks

Underléa Corrêa[1], Carlos Montez[2], Diego Ferreira[2], Tiago Semprebom[2]
[1] Department of Electrical Engineering, [2]Department of Automation System
University of Santa Catarina(UFSC), Centro Tecnológico, CEP 88040-900, Caixa Postal 476, Florianópolis/SC, Brazil
{underlea, montez, tisemp}@das.ufsc.br

Abstract. Usually, traditional multi-agent systems (MAS) run on powerful nodes such as desktops and the like. Over the past few years, agent concepts and technologies have been introduced as an option to increase the network lifetime and to provide application reprogramming functionalities in wireless sensor networks (WSN). One important problem is however how agents impact those computing systems, which are typically highly resource-constrained. In this paper we propose a tiny multi-agent model for WSN using target tracking as the motivational application. The goal is to use this model and application to assess potential benefits of using agent concepts in wireless sensor networks.

Keywords: Wireless Sensor Networks, Solving Distributed Problems, Coordination Protocol, Multi-Agent Systems.

1 Introduction

Wireless sensor networks are also expected to be both autonomous and long-lived, surviving environmental conditions, while being as much as possible energy-efficient. All of those requirements need for expertise approaches, to be meeting. Much of these are already possible to find in different MAC and network layer protocols, which perform smart behaviour such as, putting the nodes to sleep during idle periods, setting short duty-cycle and performing in-network processing to avoid data redundancy transmission. Nevertheless, the application layer usually act as a simple data producer, only storing and forwarding its environment readings, without performs any specialized decision about what it is observing, before transmit it.

Since a WSN is a large-scale distributed system with limited resources, and in some applications – such as target tracking – the overall task of monitoring different targets cannot be sensed from one single position, the nodes should coordinate their activities among themselves, acting as proactive as possible to achieve their own interests and satisfy the global application goals. In this paper we adopt agent-like concepts in a coordination target tracking protocol that attempt analyses its observing phenomena before decide send or not a sink notification. The main hypothesis is that, using this coordination approach, resource requirements are reduced, while network lifetime is extended.

The rest of the paper is organized as follows. In Section 2 we introduce our system model using a target tracking application over wireless sensor networks as a motivational application. Section 3 we introduce some analyses discussion made over an experimental test bed environment. Related Works are presented in Section 4. Finally, in Section 5 some conclusions are drawn and ongoing works described.

2 System Model

We assume a scenario with one static sink node, and a large number of sensor nodes randomly disposed in a grid region |R|. Each sensor node has an *omnidirectional* radio and acoustic sensing capacity performed also by an *omnidirectional* microphone. The nodes have knowledge about its geographic coordinate reference position, and each one will host an reactive agent, which should help them to make decisions about what to do with the observed phenomena.

2.1 Agent Model

In this system, agents are independent reactive software entities which are permanently perceiving, deliberating and executing (Figure 1). They have incompatible goals, insufficient skills and local resources, to deal alone with all of sensing task that they need perform [1]. Thus, we suppose, the agents over WSN nodes, forming a multi-agent system which perform a *collective conflict over resource* interaction, to prevent three main situation: (1) *transmission conflict* over the shared wireless channel. The agents are hosted over nodes that share a correlated time and space. (2) to *increase the neighborhood confidence* level. Since radio anomalies and sensor readings errors make the activity detection a difficult task, a single node sampling reading is not enough to represent, to the sink, the right network situation. However, not all readings coming from sensor nodes, needs to be routed to sink direction. Thus, the third agent interaction is justified (3) to *avoid spent energy* with *transmission/retransmission* of low confidence sensor readings.

To find those referred objective, we shall assume that the agents goal are incompatible and that the agents are benevolent and they try to help each other, or to reach a compromise, if interests relating to consistent network sink view are invoked (see Fig. 2). For this research work, incompatible goals means that agent *A*, is incompatible with agent B, if agent A and B have, as their respective goals to be achieved, the states describe by *p* and *q* respectively, and if p <=> ~q, that is: satisfies (goal(A, p)) <=> ~satisfies(goal(B,q)).

Usually, target tracking implies the existence of three phases: **Detection; Decision; and Tracking process**. In this research, we have introduced a target tracking application based on target's sound pressure level. We introduce in our agents an ODT-Model – *Observation, Decision and Tracking Model* – which should represent the agent's actions as a *computing processes.* For *detect* target presence, an acoustic sensor is periodic started; The **Decision** is performed after a certain **Observation** time window; at this point an agent will read the sensor readings buffer, classifies the target and decides whether it will notify other agents placed on different node in its same region (usually one-hop node); and **Tracking process** where messages are effectiveness forwarded/reported to sink direction.

Figure 1 shows the proposed agent action model as a state machine diagram: E={O, D, LN, T, SN} are the state set. The initial state *O (Observation)* represents the nodes starting periodic its acoustic sensors to scanning the physical environment with the goal of detecting a target presence. The alphabet A={a, b, c, d, e, f, g, h, i} is composed by the set of transition events that may occur in the system. After an observation time window, the agents will take from local buffer, those referred readings and compute its belief level about the environment sample (*a*). Each agent will change from *O* (*observation)* state to *D* (*decision)* state, whenever they have reading from its sensors any relevant target presence information (*b*).

In the D (*decision)* state, once an agent believes that it has detected a target, it should changes to *LN* (*LocalNotification*) state and broadcast a *local notification* (*e*) message, since the wireless channel is free to be used, and go back to O (*observation)* state (*f*). Otherwise, it just should back to *O* state and wait for a neighbor notification (*c*). Whenever the agents receive a *local notification* they should re-compute its belief to take a decision about starting a tracking process (*d*). On *T* (*tracking)* state, an agent can decide to go to *SN* (*SinkNotification*) state and diffuse a *sink notification* (*g*)*,* and back to O state (*h*), or send nothing and just go back to observing the local buffer with physical measurements (*i*) or neighbor sink notification.

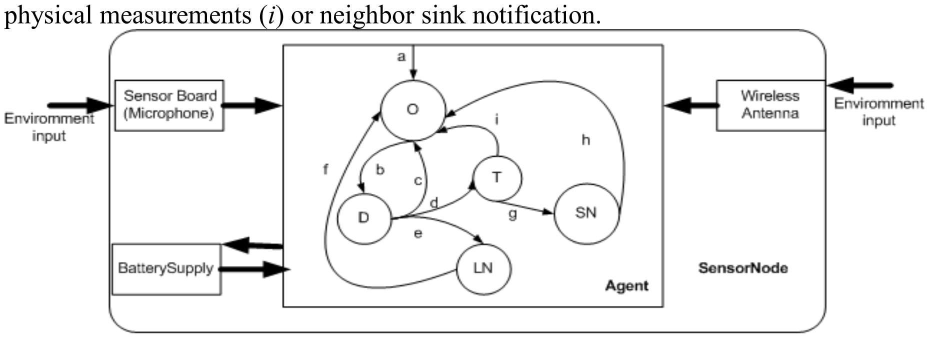

Fig. 1 – State diagram of agents performing a target tracking task. Circle denotes each possible agent action state while the arcs represent state action transitions.

Note that we have introduced two kinds of notification message, termed *local notification* and *sink notification* to be used on different stages. The first message is to the nodes over same region coordinate itself increasing its own confidence about the target and the probability of sent a sink notification. The second one is to notify the sink about the network condition (target presence, in this case).

The main difference between them is that the former is less energy consuming for the network because its message notification is propagated only on the neighborhood. The agents will be able to forward a *local notification* only when they are in *decision* state and they have computed a high belief level (see Table 1 on subsection 2.1.1) about an observed phenomenon. When one agent broadcast its *local notification* all of those neighbors must receive the sent message and suppress its sent local notification.

2.1.1 Agent Observation State. On this state, related to the Detection phase, each agent should be able to autonomously detect the target presence by reading the acoustic sensor buffer readings and computing the incoming environment sound pressure level. Sound pressure normally is represented on a logarithmic amplitude scale, which reflects a closer relationship to the human perception of hearing. Since we are concerned about sound source objects that can be sensed by humans, we introduce in our agents a simple Sound Pressure Level (SPL) model based on human perception hearing that is computed as follows:

$$L_p = 20 \log_{10}(P/P_{ref}) \qquad \textbf{(2.1)}$$

where P is the actual sound pressure and P_{ref} is the power reference given by 20µPascal, which roughly corresponds to the lower threshold of human hearing (0 dB)[2].

2.1.2 Agent Decision Process. In this stage, it is assumed that after computing its sensed pressure level (L_p) the agent should determine if what it has observed is or not relevant to forward a local or sink notification. For this first experiment we have introduced a decision model based on intensity sound measure and on the residual energy level of each node.

Table 1 – Sensing reading states and agent actions

Belief	Condition	Action
0	Illegible	Go back and listen the environment
1	Legible	Start a decision process

The range over which a signal can be detected (*active space*) is not solely a function of its source level, but also the signal attenuation (due to distance) and the level of background noise. Following those environment conditions the input signal for a sensor node at position *p*, observing a sound event source at position *q* has an intensity measured as follows:

$$I = P_{med} / 4\pi r^2 \qquad \textbf{(2.2)}$$

where r is the distance from source and P_{med} is the source power output, which is propagated overall a spherical surface area $4\pi r^2$ [2].

In our model it is supposed that all sensors (of the same modality) have a common sensing range *r*, and the events entering in this sensing range are detected with two levels (legible or illegible) of confidence, while the events outside this range are not detected at all. Since the spherical wave intensity will vary with $1/r^2$, we are supposing the nodes placed far from source path will compute a sound pressure level shorter than those nearest from source target.

The source sound intensity, I_s, measured for agents will be compared with a minimum threshold (fixed in advance). For instance, assume that I_d is the sound pressure level detected for a sensor node at a distance *d*, I_p is the intensity propagated from the sound source , and β is the agent belief that is associated with I_d and the belief can be either in *legible* or *illegible* state (Table 1). Thus the agent belief is computed as follows:

$$\beta = \begin{cases} 0: \text{If } I_d < I_p \\ 1: \text{otherwise} \end{cases} \qquad \textbf{(2.3)}$$

In accordance with its computed belief, the agent will take a decision about starting a local notification (to perform a collaborative decision approach) or a sink notification. It will depend of the next tracking stage.

2.1.3 Agent in Tracking Target State. The **Tracking** state is related with the packet forward mechanism from sensor nodes to sink node. Differently from other approaches [2,3], we assume a local tracking process without a static node coordinator. Thus, instead of each agent forwarding its own sensed data, to increase a static coordinator confidence we use the inherent spatial node correlation. The main idea is to assume that the node which computes a high belief should encourage its neighbors to broadcast a local notification informing its time and sensed SPL.

```
Invoke when
    The sensor reading is legible;
    or
    The node receives a local notification;
Action
  const EnergyThreshold = 50%;
RULE 1:
  // If belief == 1 and energy level is not
  // enough but nobody has transmitted
  // a localNotification then
  // sendLocalNotification (Social behavior)
  if ( stateBelief == 1 and
      energy < EnergyThreshold and
      listenNotification() == false)
  then sendLocalNotification();
RULE 2:
  // If belief == 1 and energy level is enough
  // but someone alredy has transmitted a
  // localNotification then Don't
  //sendLocalNotification(Selfish behavior)
  if ( stateBelief == 1 and
      energy >= EnergyThreshold and
      listenNotification() == true)
   then suppressLocalNotification();
    Go to Rule 3
  else Go to Rule 1

RULE 3:
  // If received a localNotification and
  // the number of localNotification is α
  // then send sinkNotification
  if (listenNotification() == true and
      receivedLocalNotification == α)
  then sendSinkNotification();
```

Fig. 2 – Agents basic conventions rules.

Note that there are two important characteristics associated to our agent behavior: (i) *Selfish Decision (Rule 2)* and (ii) *Social Dependence (Rule 1)* (Fig.1) [1]. Selfish decision results from the fact that sensor readings with low energy and belief level are not transmitted, and the agent should suppress the sending of its *localNotification* where it has received a *localNotification* with same period from its local reporting. Social Dependence exists since nodes, more confident about the target observation, will notify its neighbors encouraging them to forward a sink notification. This social dependence is based on the spatial correlation between one node and its one single-hop neighbors. Since one node has detected any target in some position, at certain time, there is a probability P_d that the neighbors on the same coverage range (on active status) will detect the object moving from position p_i to position p_{i+1}.

Although the goal of the application is to notify the sink node about the sensed targets, the agent aims to maximize its node resources or network utilization (taking selfish decisions) to avoid network congestion, collision, and data retransmission. Thus, in a coverage area, only the node with the high belief in its detection will transmit its local readings, and the node which listens *n* local notifications should create and forward a sink notification. Decreasing the amount of data transmission could not be a good solution because it also decreases the sink data delivery reliability. Thus, to attempt finding some good trade-off between delivery reliability and network congestion we introduce a variable *n* that will determine the frequency between the agent taking a selfish or a social decision (see Figure 2). All nodes detecting a legible reading should not forward a sink notification while its local notification is not equal to value *n*. The optimal trade-off between sink delivery reliability and network congestion is now being object of ongoing research considering specific WSN infrastructures.

2 Experimental Environment

3.1 Scenarios Definitions

Our approach was exercised on a topology of irregular communication (Figure 3 b), where each node has the ability to receive and/or send notifications to/from any of its neighbours (one-hop distance). Always that a node is elected the leader of round[1], it should send a notification to sink through multi-hops. The idea is that each new round, only one agent wins the local notification transmission turn (following the rules set out in subsection 2.1.3). This behaviour will conduct to dynamically selecting a round coordinator, to receive the local notifications, create and send a Sink notification. Our experiments was contrasted with a traditional communication star model (typically used in applications for tracking target that suggest local processing). Unlike what we propose in our model, each node has a static role (producer or coordinator), defined in advance and running throughout the life of the network nodes.

The producer node only receives and forwards information to its static coordinator, while the coordinator receives and forwards the received notifications, toward the Sink (Figure 3 (a)). This idea of producer/coordinator is the same for both models (to send sink notification). The difference is the dynamic coordinator election, proposed in our model and how this election is made (Section 2.1). The proposed model (Figure 3 (b)), the goal of each producer node is, not forward, but suppress their local notification, giving the transmission preference for the node (one node) which in a specific round, had calculated greater confidence level about the physical environmental measurements. Since the nodes are spatially and temporally correlated. There is no need to all of them send its measurements to coordinator. One local message should be enough to increase the confidence level from each other node on the region.

[1] A round is defined as the period that a node starts each new reading of the environment held by the sensors and finishes with an agent decision about wait to send a notification to Sink Node.

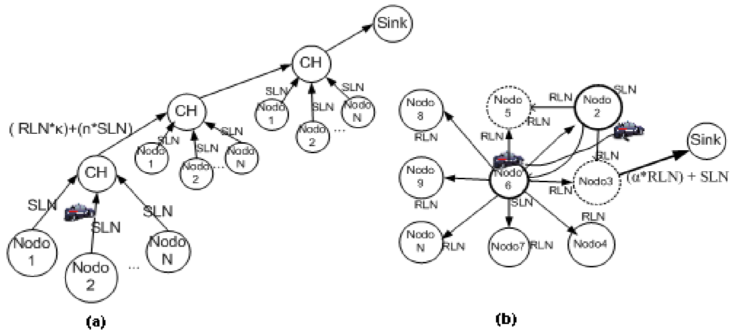

Fig. 3. Communication Models of static coordinator(a), and dynamic coordinator (b). In (a), each row represent a broadcast message send for each N node over a region. While in Figure (b) each *n* row spread from node 6 means only one broadcast message.

The hypothesis is that by reducing the transmission of messages over a region, it is also possible decreased the energy consumption on it. Note that in Figure 3 (b), only the node 6 broadcast a notification on your local area. All other nodes will receive the sent message and suppress its local notification, satisfying the Rule 1 (the node to send a notification to local) and Rule 2 (nodes that suppress its local notification) from the set rules for each agent (Figure 2).

To evaluate the energy consumption of each model, we define two kinds of tasks, from the energy model presented in [4], involving the whole execution process of the nodes. These tasks are: (1) SLN (SendLocalNotification) for the task of sending a notification local (or Sink Notification) and (2) RLN (ReceiveLocalNotification) to receive a local notification. The spent energy to perform each of these tasks is computed as follow

$$T_{SLN} = E_d + E_{listen} + E_{tx}$$
$$T_{RLN} = E_d + E_{listen} + E_{rx} \quad (3.1)$$

In traditional model (Figure 3(a)) at each round all producer node will send a local notification to its static coordinator, that should run *k* times the same RLN task spent a total energy giving for $E = ((T_{RLN}*\kappa)+(n*T_{SLN}))$, where κ is the number of producer neighbour from each static coordinator, and *n* is the number of packets retransmitted from each static coordinator. In our model, the coordinator spend no more than twice with the task of receiving a local notification and once with the task to send a sink notification. Thus, for this model the total coordinator energy consumption, in each round will be E = (TRLN * α) + TSLN, where an object is accurately detected by a sensor node on the region of the coordinator node elected. Note that the number of local reports, is determined by the value of α, defined on the agents Rule 3 (Figure 2). In this model, we set α to get a value equal to 2. This definition is given for the attempt in establishing a trade-off between reliability in the delivery of messages to the base station, and energy consumption in the region. As, the greater the need for reliability in the delivery of messages, the lower the value of α.

The experimental scenario was composed of 8 sensor nodes MICAZ category Motes, arranged on a region |4x4|. Each node is limited to cover an area up to 4*m*, called here node coverage cell. The routing table of each node is composed of at least four addresses of neighbors reachable with a single hop, and the nodes are periodically synchronized by the receipt of a message sent by the sink node.

The target tracking used in the experiments was a remote control car, emitting a sound pressure of 70 dB and moving at a speed of 0.5 m/s.

3.2 Results Analyses

In this Section, we present some experiments results comparing the advantages (from the energy point of view) to use an application layer implementing a software system that runs in accordance with its energy node resources. Different from objects paradigm, agents, when autonomous, have their own rules to solve a problem, for the benefit of themselves or their society.

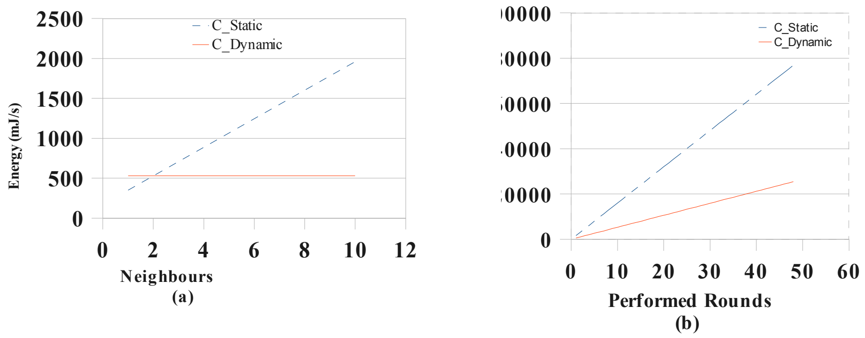

Fig. 4. Energy consumption for each Coordinator of Different Models.

Our hypothesis were that by introducing a small level of processing, and reduce the spread of messages on a region, it is also possible reduce the energy consumption of each nodes in this region. Figure 4 *(a)* shows that in regions with up to 2 neighboring nodes (producer nodes), the energy consumption of the model with static coordinator is less than the model that implements the dynamic coordinator. This is because, when the static coordinator has only one neighbor, the number of messages received is less than the minimum number of messages received for a node that is elected coordinator dynamically(α = 2, as mentioned on above Section). In this case, we noticed that the energy consumption are equivalent in both models. What makes our model more advantageous is the uniform way to spend energy. While the model with static coordinator (Figure 3 (a)) the energy consumption of the coordinator, grows proportionally with the number of producers nodes . In our model the energy consumption will be the same for a region with 2 or N nodes.

In Figure 4 (b), we show the results referring to total energy consumption on each coordinator doing the same number of Sink notifications. Although the task of sending a message, is energetically less costly than the task of receiving a message, again, we see that the employment of a cooperation protocol is more economical, because the number of messages propagated on the model that implements the cooperation protocol, is considerably smaller than the number of messages disseminated by sensors that perform only the production and propagation of messages within their region. In Figure 5, we show the results for the *total energy* consumption on a region for the different models contrasted in this work. Note that when the network region is small, up to 4 nodes, the total energy consumption for the task of send a Sink notification is lower for the model that implements a static coordinator. This is because receive a message is more expensive than sending a message over the wireless channel.

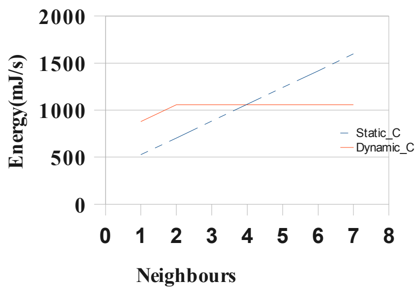

Fig 5. Total region energy consumption.

Thus it is not quite true to say that to decrease the amount of transmitted messages also reduce the amount of energy consumed over a region. This assumption is directly related to the network density. Then, what we can say is that our model shows to be more advantageous in terms of energy consumption for groups of nodes greater than 5.

4 Related Works

In the WSN literature, agents have been introduced to perform activities related to both data and resources management. In many of those works [5-7] mobile agents are used to perform some kind of in-network processing, such as data gathering, aggregation or fusion. In some other works [8-10] agents are used to execute sensor node reprogramming for update, replacement or reconfiguring code in the sensor nodes. There are also works with are directly concerned with target tracking [11, 12] and with some decision maker process related with sensor readings. Nevertheless, those research does not introduce a multi-agent system in wireless sensor networks to cooperate one each other, aiming to solve a distributed problem and safe energy resources. Although could be considered a smart strategy each node performs tasks such as data fusion, data aggregation or any other data filter activity before they send their environment data reading to sink, those behaviour still spent much of energy whether we consider the dynamic system aspects.

In a multi-agent system, the inputs can be processed aiming to provide answers that meet specific goals for overall system considering not only the inputs, but also the conditions of the system. Thus, different from other work, our research aims introduce the development of a multi-agent system on a real platform of wireless sensor networks, that is capable of coordinating the task of sending messages to sink direction by establishing a trade-off between the consistent delivery of information without depreciate quickly the system energy resources.

5 Conclusion and Future Works

We presented a model for coordination message protocol among sensor nodes, employed by a model of reactive agents to verify the advantages (related to energy consumption) to deploy an application layer with some kind of autonomous behaviour about take decision of to send or not a local and sink notification. The hypothesis was that by introducing a small level of processing, and reduce the spread of messages over a region, also the energy consumption will be reduced. The results of experiments showed that this hypothesis is true for networks with more than 5 sensor nodes per cluster. Therefore, use an application layer implementing a coordinator protocol is advantageous for large WSNs.

For future work, we are preparing a comparisons scenario between our proposed model and other model implementing some kind of data fusion tasks over the static coordinator. Moreover, focusing our efforts to measure aspects related to time constraints, we will introduce on each reactive agent, a task model of real time to evaluate delays (minimum and maximum) from end to end communication and delays about target detection that are acceptable for the suggested target tracking application.

References

1. Ferber, Jaques. Mulit-Agent Systems – An Introduction to Distributed Artificial Inteligence, Addison-Wesley, 1999.
2. Brooks, Richard R., Ramanathan, Parameswaran, Sayeed, Akbar M. (2003) “Distributed Target Classification and Tracking in Sensor Networks”, Proceedings of the IEEE, Vol. 91, No. 8.
3. Horling, Bryan, Vincent, Regis, Mailler, Roger, Shen, Jiaying, Becker, Raphen, Rawlins, Kyle and Lesser, Victor. (2001) “SPT: Distributed Sensor Network for Real Time Tracking”, In Fifth International Conference on Autonomous Agents, Montreal – Canada.
4. Polastre, Joseph, Hill, Jason, Culler, David, (2004) “Versatile Low Power Media Access for Wireless Sensor Networks”, SenSys' 04, Baltimore, Maryland, USA.
5. Malik, Haroon, Shakshuki, Elhadi, Dewolf, Travis, Denko, Mieso K.. (2007) “Multi-agent System for Directed Diffusion in Wireless Sensor Networks”, Proc. of 21st Int. Conf. on Advanced Information Networking and Applications Workshops (AINAW´07), IEEE Computer Society.
6. Hussain, Sajid, Shakshuki, Ellhadi, Matin, Abdul Wasey. (2006) “Agent-based System Architecture for Wireless Sensor Networks”, Proc. of the 20th Int. Conf. on Advanced Information Networking and Applications (AINA´06), IEEE Computer Society.
7. Chen, Min, Kwon, Taekyoung, Yung, Yong, Yanghee, and Leung, Victor C. M.. (2007) “Mobile Agent-based Directed Diffusion in Wireless Sensor Networks”, EURASIP Jornal on Advances in Signal Processing.
8. Massaguer, Daniel, Fok, Chien-Liang, Venkatasubramainan, Nalini, Romain, Gruia-Catalin and Lu, Chenyang. (2006) “Exploring Sensor Networks using Mobile Agents”, Proc. of the fifth international joint conference on Autonomous agents and multiagent systems, Hakodate, Hakodate.
9. Hui, J. W., Culler, D. (2004) “The dynamic behavior of a data dissemination protocol for network programming at scale”, In Proc. of the Second Int. Conf. on Embedded Networked Sensor Systems ACM Press.
10. Soe, Khin Thanda, (2008) “Increasing Lifetime of Target Tracking Wireless Sensor Networks”, Proceedings of World Academy of Science, Engineering and Technology, Volume 32, ISSN 2070-3740.
11. Toh, Yue Khing, Xiao, Wendong, Xie, Lihua. (2007) “A Wireless Sensor Network Target Tracking System with Distributed Competition based Sensor Scheduling”, Third International Conference – Intelligent Sensor Network and Information Proceeding, Melbourne, Australia.