

Giving neurons to sensors. An approach to QoS management through artificial intelligence in wireless networks.

Julio Barbancho, Carlos León, Javier Molina and Antonio Barbancho

Department of Electronic Technology, University of Seville.
C/ Virgen de Africa, 7. Seville 41011 (Spain)
Tlfn.: (+034) 954 55 71 92, Fax: (+034) 954 55 28 33.
{jbarbancho, cleon, fjmolina, ayboc}@us.es

Abstract. For the latest ten years, many authors have focused their investigations in wireless sensor networks. Different researching issues have been extensively developed: power consumption, MAC protocols, self-organizing network algorithms, data-aggregation schemes, routing protocols, QoS management, etc. Due to the constraints on data processing and power consumption, the use of artificial intelligence has been historically discarded. However, in some special scenarios the features of neural networks are appropriate to develop complex tasks such as path discovery. In this paper, we explore the performance of two very well known routing paradigms, *directed diffusion* and *Energy-Aware Routing*, and our routing algorithm, named **SIR**, which has the novelty of being based on the introduction of neural networks in every sensor node. Extensive simulations over our wireless sensor network simulator, OLIMPO, have been carried out to study the efficiency of the introduction of neural networks. A comparison of the results obtained with every routing protocol is analyzed. This paper attempts to encourage the use of artificial intelligence techniques in wireless sensor nodes.

Keywords: Wireless sensor networks (WSN); Ad hoc networks, Quality of service (QoS); Artificial neural networks (ANN); Routing; Self-Organizing Map (SOM), ubiquitous computing.

1 Introduction

In recent years technological advances have made the manufacturing of small and low-cost sensors economically and technically possible. These sensors can be used to measure ambient conditions in the environment surrounding them. Typically, wireless sensor networks (WSNs) contain hundreds or thousands of those sensors nodes. Due to the sensor features (low-power consumption, low radio range, low memory, low processing capacity, and low cost), self-organizing network is the best suitable network architecture to support applications in such a scenario. Goals like efficient energy management [1], high reliability and

availability, communication security, and robustness have become very important issues to be considered. This is one of the many reasons why we can not neglect the study of the collision effects and the noise influence.

Many research centers worldwide (specially in Europe and USA) have focused their investigations on this kind of networks. Ian Akyldiz et al. [2] and Holger Karl et al. [3] have made great effort to describe the state-of-the-art of this subject.

Our research group, Computer Science for Industrial Applications, from the University of Seville, is working on the development of protocols and system architectures on Wireless Sensor Networks to support Supervisory Control and Data Acquisition (SCADA) applications. We present in this paper a new routing algorithm which introduces artificial intelligence (AI) techniques to measure the QoS supported by the network.

This paper is organized as follows. In section 2, we relate the main routing features we should consider in a network communication system. A description of the defined network topology is given. Section 3 introduces the use of neural networks in sensors for determining the quality of neighborhood links, giving a QoS model for routing protocols. The performance of the use of this technique in existing routing protocols for sensor networks is evaluated by simulation in section 4. Concluding remarks and future works are given on section 5.

2 Designing the network topology

The WSN architecture as a whole has to take into account different aspects, such as the protocol architecture; Quality-of-Service, dependability, redundancy and imprecision in sensor readings; addressing structures, scalability and energy requirements; geographic and data-centric addressing structures; aggregating data techniques; integration of WSNs into larger networks, bridging different communication protocols; etc.

The protocol stack proposed by our research group is based on the OSI model. In the lower layers we can use the well known IEEE wireless sensor network standard 802.15.4 or our own protocol *Arachne* [4]. In the upper layers there are other protocols, such as transmission clock to base station, ping, data aggregation, and our SIR protocol.

If an application is able to perform at an acceptable level using data from a number of different sensors set, like a typical SCADA application [5], we would schedule the sets so as to maximize the sum of the time that all sensor sets are used. Acknowledging the impact that route selection will have on network lifetime, we would like to determine route selection in conjunction with the sensor schedule. In general, the routes should be chosen so that nodes that are more critical for use as sensors are routed around as often as possible. Many authors have studied this problem [6],[7]. In this section, we model this scenario in which sensors are working, and in section 3 we formalize the routing algorithm, SIR, proposed to solve this problem.

Due to the desire to cover a large area, a communication strategy is needed. there are many studies that approach the problem of high connectivity in wireless ad hoc networks [8], [9], [10], [11]. In our research we consider a random distribution of sensors, as depicted in figure 1.

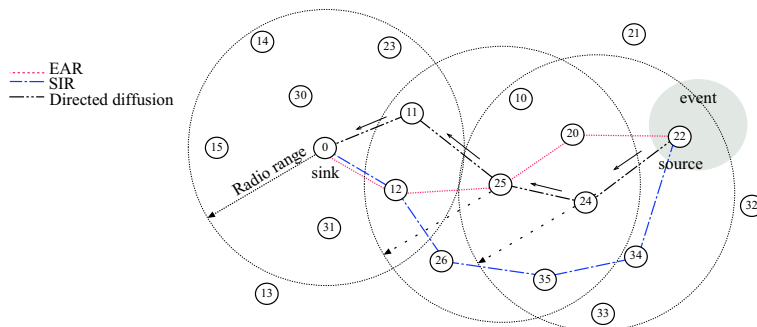


Fig. 1. Event transmission from a source to a sink.

In this scenario, every node has a radio transmitter power and a radio receiver sensibility, which defines an average radio range. There are several network routing protocols proposed for wireless networks that can be examined in the context of wireless sensor networks. Two basic paradigms are minimum-transmission-energy multi-hop routing protocol and direct communication. Using a direct communication protocol, each sensor sends its data directly to the base station. If the base station is far away from the nodes, direct communication will require a large amount of transmitting power from each node. This quickly drains the battery of the nodes and reduces the network lifetime. Using minimum-transmission-energy routing protocol, nodes route data to the base station through intermediate nodes. Thus nodes act as routers for others nodes' data. The problem is how to elect intermediate nodes, in which the final objective es to minimize the global energy consumption.

In general, routing in WSNs can be divided into *flat-based routing*, *hierarchical-base routing*, and *location-based routing*. In this paper we study networks where all nodes are supposed to be assigned equal roles or functionalities. In this sense, flat-based routing is best suited for this kind of networks.

Among all the existing flat routing protocols, we have chosen *directed diffusion* and *Energy-Aware Routing (EAR)* to evaluate the influence of the use of AI techniques.

In directed diffusion [12], sensors measure events and create gradients of information in their respective neighborhoods. The base station request data by broadcasting interests. Each sensor that receives the interest sets up a gradient toward the sensor nodes from which it has received the interest. This process continues until gradients are set up from the sources back to the base station.

EAR [13] is similar to directed diffusion. Nevertheless it differs in the sense that it maintains a set of paths instead of maintaining or enforcing one optimal path at higher rates. These paths are maintained and chosen by means of a certain probability. The value of this probability depends on how low the energy consumption that each path can achieve is. By having paths chosen at different times, the energy of any single path will not deplete quickly.

3 Introducing neurons in sensor nodes

The necessity of connectivity among nodes introduces the routing problem. In a WSN we need a multi-hop scheme to travel from a source to a destiny. The paths the packets have to follow can be established based on a specific criterion. Possible criteria can be minimum number of hops, minimum latency, maximum data rate, minimum error rate, etc. For example, imagine that all the nodes desire to have a path to route data to the *base station*¹. In this situation, the problem is solved by a technique called *network backbone formation*.

Our approach to enhance this solution is based on the introduction of artificial intelligence techniques in the WSNs: expert systems, artificial neural networks, fuzzy logic and genetic algorithms. Due to the processing constraints we have to consider in a sensor node, the best suited, among all these techniques, is the *self-organizing-map (SOM)*. This is kind of artificial neural network based on the self organization concept. A description of our SOM approach is detailed in section 3.3.

3.1 Network backbone formation

This problem has been studied in mathematics as a particular discipline called *Graph Theory*, which studies the properties of graphs.

A *directed graph* G is an ordered pair $G := (V, A)$ with V , a set of vertices or nodes, v_i , and A , a set of ordered pairs of vertices, called *directed edges*, *arcs*, or *arrows*.

An edge $v_{xy} = (x, y)$ is considered to be directed from x to y ; where y is called the head and x is called the tail of the edge.

In 1959, E. Dijkstra proposed an algorithm that solves the single-source shortest path problem for a directed graph with nonnegative edge weights.

In our wireless sensor network we assume that all the links are symmetrical, in the sense that if a node A can reach a node B , then the node B can reach the node A . With these kinds of links, we can model our network as an *undirected graph* $G := (V, E)$.

We propose a modification on Dijkstra's algorithm to form the network backbone, with the minimum cost paths from the base station or *root*, r , to every node in the network. We have named this algorithm Sensor Intelligence Routing,

¹ In WSN, we often consider two kind of nodes, base stations and sensor nodes. There is usually only one base station.

SIR [14]. In Dijkstra's algorithm the graph has arrows and in our modification the graph has edges. Every edge between nodes v_i and v_j has a weight, w_{ij} , and it is easy to prove that $w_{ij} = w_{ji}$. The distance from the base station to a node v_i is named $d(v_i)$. The set of nodes which are successors or predecessors of a node v_i is denoted by $\Gamma(v_i)$, and can be defined in this way: $\Gamma(v_i) = \{v_j \in V \mid (v_i, v_j) \in E\}$. If we denote a path from the root node to a node v_k by p , we can define $\Gamma_p(v_j)$, if $v_j \in p$, as the subset of nodes which are predecessors or successors of node v_j .

We also assume that $V = \{r, v_i\}_i$ and that there is a subset of V , T , defined as $T := V - \{r\}$. Furthermore, we can denote \bar{T} as the complementary set of T , $\bar{T} = \{r\}$.

With this terminology, our algorithm can be described as follows:

Step 1: Set up phase:
 $d(r) = 0$
 $d(v_i) = \begin{cases} w_{ri} & \text{if } v_i \in \Gamma(r) \\ \infty & \text{if } v_i \notin \Gamma(r) \end{cases}$
 $\Gamma_p(v_i) = \begin{cases} r & \text{if } v_i \in \Gamma(r) \\ \emptyset & \text{if } v_i \notin \Gamma(r) \end{cases}$

Step 2: Find a $v_j \in T$ such as $d(v_j) = \min\{d(v_i) \mid v_i \in T\}$
Do $T = T - \{v_j\}$

Step 3: $\forall v_i \in T \cap \Gamma(v_j)$ calculate $t_i := d(v_j) + w_{ji}$
If $t_i < d(v_i)$ do $d(v_i) = t_i$

Step 4: If $|T| > 0$ go to step 2
If $|T| = 0$ stop

3.2 Quality of Service in Wireless Sensor Networks

Once the backbone formation algorithm is designed, a way of measuring the edge weight parameter, w_{ij} , must be defined. On a first approach we can assume that w_{ij} can be modelled with the number of hops. According to this assumption, $w_{ij} = 1 \forall i, j \in \mathcal{R}, i \neq j$. However, imagine that we have another scenario in which the node v_j is located in a noisy environment. The collisions over v_j can introduce link failures increasing power consumption and decreasing reliability in this area. In this case, the optimal path from node v_k to the root node can be p' , instead of p . It is necessary to modify w_{ij} to solve this problem. The evaluation of the QoS in a specific area can be used to modify this parameter.

The traditional view of QoS in communication networks is concerned with end-to-end delay, packet loss, delay variation and throughput. Numerous authors have proposed architectures and integrated frameworks to achieve guaranteed levels of network performance [15], [16]. However, other performance-related features, such as network reliability, availability, communication security and robustness are often neglected in QoS research. The definition of QoS requires some extensions if we want to use it as a criterion to support the goal of controlling the network. This way, sensors participate equally in the network, conserving energy and maintaining the required application performance.

What is sensor network QoS? Ranjit Iyer and Leonard Kleinrock proposed in [17] a definition of sensor network QoS based on sensor network resolution. They define resolution as the optimum number of sensors sending information toward information-collecting sinks, typically base stations. James Kay and Jeff

Frolik defined sensor network QoS in terms of how many of the deployed sensors are active [18]. The same idea is discussed in [19] by Mark Perillo et al., and in [20] by Veselin Rakocevic et al.

We use a QoS definition based on three types of QoS parameters: timeliness, precision and accuracy. Due to the distributed feature of sensor networks, our approach measures the QoS level in a spread way, instead of an end-to-end paradigm. Each node tests every neighbor link quality with the transmissions of a specific packet named *ping*. With these transmissions every node obtains mean values of latency, error rate, duty cycle and throughput. These are the four metrics we have defined to measure the related QoS parameters.

Once a node has tested a neighbor link QoS, it calculates the distance to the root using the obtained QoS value. The expression 1 represents the way a node v_i calculates the distance to the root through node v_j , where qos is a variable whose value is obtained as an output of a neural network. This tool is described in section 3.3.

$$d(v_i) = d(v_j) \cdot qos \quad (1)$$

3.3 SOM: Self Organizing Map

One of the most powerful mechanism developed in AI is the Self-Organizing Map (SOM) model [21], created by Teuvo Kohonen in 1982, at the University of Helsinki, Finland.

SOM is an unsupervised neural network. The neurons are organized in an unidirectional two layers architecture (figure 2). The first one is the input or

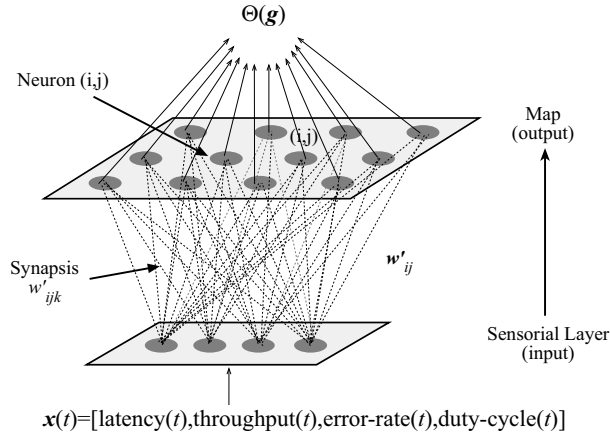


Fig. 2. SOM architecture.

sensorial layer, formed by m neurons, one per each input variable. These neurons

work as buffers distributing the information sensed in the input space. The input is formed by stochastic samples $\mathbf{x}(t) \in \mathcal{R}^m$ from the sensorial space. The second layer is usually formed by a rectangular grid with $n \times n' \times y$ neurons². Each neuron (i, j) is represented by an m -dimensional weight or reference vector called *synapsis*, $\mathbf{w}'_{ij} = [w'_{ij1}, w'_{ij2}, \dots, w'_{ijm}]$, where m is the dimension of the input vector $\mathbf{x}(t)$. The neurons in the output layer -also known as the competitive Kohonen layer- are fully connected to the neurons in the input layer, meaning that every neuron in the input layer is linked to every neuron in the Kohonen layer. In SOM we can distinguish two phases:

Learning phase: In this phase, neurons from the second layer compete for the privilege of learning among each other, while the correct answer(s) is (are) not known. This implies that for a certain input vector, there is only one neuron that gets activated. To determine which neuron is going to be activated, the input vector is compared with the vector that is stored in each of the neurons, the so-called synaptic-weight-vectors. Only the neuron whose vector most closely resembles the current input vector dominates, $d(\mathbf{w}'_{\mathbf{g}}, \mathbf{x}) = \min_{ij} \{d(\mathbf{w}'_{ij}, \mathbf{x})\}$. Consequently, only the winning neuron is allowed to learn; and its synaptic-weight-vector is updated.

This phase is executed in a central data processing unit (e.g. *offline processing*).

Execution phase: The weights are declared fixed.

First, every neuron (i, j) calculates the similarity between the input vector $\mathbf{x}(t)$, $\{x_k \mid 1 \leq k \leq m\}$ and its own synaptic-weight-vector \mathbf{w}'_{ij} . This function of similarity is based on a predefined similarity criterion.

Next, it is declared a winning neuron, $\mathbf{g} = (g_1, g_2)$, with a synaptic-weight-vector, $\mathbf{w}'_{\mathbf{g}}$, similar to the input \mathbf{x} . This phase is implemented in every node as a *C++* function (table 1).

SOM gives an output denoted by *qos*. This value is returned by a function Θ defined by the SOM user, according to its aims. Θ depends on the winning neuron: $qos = \Theta(\mathbf{g})$. In section 4.3 we define this function.

This phase is executed in every node (e.g. *online processing*).

4 Performance evaluation by simulation

Due to the desire to evaluate the SIR performance, we have created two simulation experiments running on our wireless sensor network simulator OLIMPO [4]. Every node in OLIMPO implements a neural network (SOM) running the execution phase detailed in table 1 (online processing).

² Although this architecture is the most customary in SOM, sometimes it is used layers with only one dimension (linear neuron chain) or with three dimensions (parallelepiped) [21].

Table 1. Implementation of the winning neuron election in C++.

```

int WinnerNeuron(float *x)
{
    float d2 = 0; % distance ^ 2
    float d[12]; % distance between input and every neuron weight
    for (int m = 0; m < 12; m++)
        d[m] = 0;
    for (int i=0; i < 12; i++)
    {
        d2 = 0;
        for (int j=0; j < 4; j++)
        {
            float aux = IW[i][j]-x[j]; % IW[i][j] is the input weights matrix,
            %obtained in the learning phase
            d2 += aux*aux;
        }
        d[i] = sqrt(d2);
    }
    float aux = d[0];
    int neuron = 1;
    for (int n=0; n<12; n++)
    {
        if (aux>d[n])
        {
            aux = d[n];
            neuron = n+1;
        }
    }
    return neuron;
}

```

4.1 Radio channel analytical performance evaluation

In order to accurately model the sensor networks, the wireless channel is equipped with certain propagation models which allows sensors to determine the strength of the incoming signal. These models are integrated in the channel object of the simulation tool.

For the purpose of this research, the values shown in table 2 have been considered.

Table 2. Values of radio communication parameters.

Resonating frequency [†] :	869.85 MHz	Communication bandwidth [†] B :	0.5 %
Number of radio channels [†] :	1	Antenna gain [‡] :	$G_r = 1, G_t = 1$
Radio transmitter power:	$P_t = 5mW$	Radio receiver sensibility:	$P_s = -101dB$
System loss	$L = 1$	Path loss exponent:	$n = 2$
Modulation:	FSK	Transmission rate, R :	4800 b/s
Input noise power density N_{in} :	-174 dBm/Hz	Noise Figure $(NF)_{dB}$:	10 dB

[†]Based on licensed free standard ETSI EN 301 291

[‡]Antennas are assumed to be omnidirectional.

In this scenario, two sensor nodes attempting to establish a radio communication link can be 218 meters separated³. In our simulations we have assumed that the distance between every pair of sensor nodes is set up randomly, as shown in figure 1. We have focused our simulation on a wireless sensor network composed by 250 nodes.

³ According to free space propagation model [22]

4.2 Noise influence

Noise influence over a node has been modelled as an Additive Gaussian White Noise, (AWGN), originating at the source resistance feeding the receiver. According to the radio communication parameters detailed in table 2 we can determine the signal-to-noise ratio at the detector input with the equation 2 [22], $S/N_d = 26.7 \text{ dB}$. This signal-to-noise ratio can be expressed as an associated BER (Bit Error Rate)⁴. If S/N_d is less than 26.7 dB the receiver can't detect any data on air. An increase of the noise can degrade the BER. In another way, due to the relation between E_b/N_o and the transmission rate (R), $E_b/N_o = (S/R)/N_o$, an increase of R can also degrade the BER.

$$(P_s)_{dBm} = (N_{in})_{dB} + (NF)_{dB} + (10 \log B)_{dB} + (S/N)_d \quad (2)$$

To evaluate the effect of noise we have defined a node state declared as *failure*. When the BER goes down below a required value (typically 10^{-3}) we assume this node has gone to a failure state. We measure this metric as a percentage of the total lifetime of a node. In section 4 we describe two experiments according to different percentages of node failures.

4.3 SOM creation

Our SOM has a first layer formed by four input neurons, corresponding with every metric defined in section 3.2 (latency, throughput, error rate and duty cycle); and a second layer formed by twelve output neurons forming a 3x4 matrix.

Next, we detail our SOM implementation process.

Learning phase: In order to organize the neurons in a two dimensional map, we need a set of input samples $\mathbf{x}(t)=[\text{latency}(t), \text{throughput}(t), \text{error-rate}(t), \text{duty-cycle}(t)]$. This samples should consider all the QoS environments in which a communication link between a pair of sensor nodes can work. In this sense, we have to simulate special ubiquitous computing environments. These scenarios can be implemented by different noise and data traffic simulations. In our research we create several WSNs over OLIMPO with 250 nodes and different levels of data traffic. The procedure to measure every QoS link between two neighbors is detailed as follows: every pair of nodes (eg. v_i and v_j) is exposed to a level of noise. This noise is introduced increasing the noise power density N_o in the radio channel in the proximity of a determined node. Hence, the signal-to-noise ratio at the detector input of this selected node decreases and consequently the BER related with its links with every neighbor gets worse.

In order to measure the QoS metrics related with every N_o , we run a ping application between a selected pair of nodes (eg. v_i and v_j). Node v_i sends

⁴ The minimum probability of bit error $P_{e,min}$, in a FSK system with an adaptive filter at the radio receiver, is typically expressed in the literature with the expression: $P_{e,min} = \frac{1}{2} \text{erfc} \left(\sqrt{\frac{E_b}{N_o}} \right)$, where $\frac{E_b}{N_o} = \frac{(S/R)}{N_o} = \frac{S}{N}$.

periodically a ping message to node v_j . Because the ping requires acknowledgment (ACK), the way node v_i receives this ACK determines a specific QoS environment, expressed on the four metrics elected: latency (seconds), throughput (bits/sec), error rate (%) and duty cycle(%). For example, for a noise power density of $N_o = -80 \text{ dBm/Hz}$ and a distance of separation⁵ between node v_i and node v_j of 60 meters the QoS measured in node v_i and expressed in the metrics defined is [0.58, 1440, 10.95, 2.50]. This process is repeated 100 times with different N_o and d . This way, we obtain a set of samples which characterize every QoS scenario.

With this information, we construct a self-organizing map using a high performance neural network tool, such as MATLAB[®], on a Personal Computer. This process is called *training*, and uses the learning algorithm detailed in section 3.3. Because the training is not implemented by the wireless sensor network, we have called this process *offline processing*.

Once we have ordered the neurons on the Kohonen layer, we identify each one of the set of 100 input samples with an output layer neuron. According to this procedure, the set of 100 input samples is distributed over the SOM. The following phase is considered as the most difficult one. The samples allocated in the SOM form groups, in such a way that all the samples in a group have similar characteristics (latency, throughput, error rate and duty cycle). This way, we obtain a map formed by clusters, where every cluster corresponds with a specific QoS and is assigned a neuron of the output layer. Furthermore, a synaptic-weight matrix $\mathbf{w}'_{ij} = [w'_{ij1}, w'_{ij2}, \dots, w'_{ij4}]$ is formed, where every synapsis identifies a connection between input and output layer.

In order to quantify the QoS level, we study the features of every cluster and, according to the QoS obtained in the samples allocated in the cluster, we assign a value between 0 and 10. As a consequence, we define an output function $\Theta(i, j), i \in [1, 3], j \in [1, 4]$ with twelve values corresponding with every neuron $(i, j), i \in [1, 3], j \in [1, 4]$. The highest assignment (10) must correspond to that scenario in which the link measured has the worst QoS predicted. On the other hand, the lowest assignment (0) corresponds to that scenario in which the link measured has the best QoS predicted. The assignment is supervised by an engineer during the offline processing.

Execution phase: As a consequence of the learning phase, we have declared an output function, that has to be run in every sensor node. This procedure is named the *wining neuron election algorithm*.

In the execution phase, we create a WSN with 250 nodes. Every sensor node measures the QoS periodically running a ping application with every neighbor, which determines an input sample. After a node has collected a set of input samples, it runs the wining neuron election algorithm. For example,

⁵ Considering the free space propagation model, the power transmitted from the source decreases according the expression $P_r = P_t \left[\frac{\lambda}{4\pi dL} \right]^2 G_t G_r$, where P_r , is the radio power received at a distance d from the transmitter; P_t is the transmitter signal power, G_t and G_r are the antenna gains of the transmitter and the receiver respectively; L ($L \leq 1$) is the *system loss* and λ is the electromagnetic wavelength.

if a specific input sample is quite similar than the synaptic-weight-vector of neuron (2,2), this neuron will be activated. After the winning neuron is elected, the node uses the output function Θ to assign a QoS estimation, qos . Finally, this value is employed to modify the distance to the root (eq. 1). Because the execution phase is implemented by the wireless sensor network, we have called this process *online processing*.

4.4 Evaluating SIR performance

Our SIR algorithm has been evaluated by the realization of three experiments detailed as follows.

Experiment #1: No node failure. The purpose of this experiment is to evaluate the introduction of AI techniques in a scenario where there is no node failure. This means that no node has gone to a failure state because of noise, collision or battery fail influence.

To simulate this scenario, a wireless sensor network with 250 nodes is created on our simulator OLIMPO. Node # 0 is declared as a sink and node # 22 is declared as a source. At a specific time, an event (eg. an alarm) is provoked in the source. Consequently, the problem now is how to route the event from the specified source to the declared sink.

As detailed in section 2 we solve this problem with three different routing paradigms: SIR, directed diffusion and EAR. We choose two metrics to analyze the performance of SIR and to compare it to others schemes. These metrics are:

- Average dissipated energy. This metric computes the average work done by a node in delivering useful tracking information to the sinks. This metric also indicates the overall lifetime of sensor nodes.

According to the first energy consumption order model proposed by Wendi Rabiner Heinzelman in the *LEACH* protocol [23], we can assume the radio dissipates $E_{elec} = 50 \text{ nJ/bit}$ to run the transmitter or receiver or receiver circuitry, and $\varepsilon_{amp} = 100 \text{ pJ/bit/m}^2$ for the transmit amplifier to achieve an acceptable $\frac{E_b}{N_o}$ (figure 3). This way, to transmit

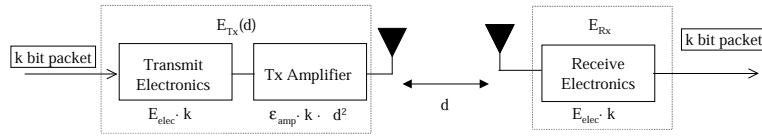


Fig. 3. Energy model.

a k -bit message a distance d using this radio model⁶, the radio expends:

$$E_{Tx}(k, d) = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^2 \quad (3)$$

⁶ We assume the radio propagation model.

and to receive this message, the radio expends:

$$E_{Rx}(k) = E_{elec} \cdot k \quad (4)$$

We assume that the radio channel is symmetric, and that our simulation is *event-driven*, that is, sensors only transmit data if some event occurs in the environment. Due to transmission distance from a sensor node to the base station is large on a global scale, the transmission energy is much more higher than the received energy. In this network topology, as detailed in section 2, the most energy-efficient protocol is the minimum-transmission-energy.

- Average Delay. This metric measures the average one-way latency observed between transmitting an event and receiving it at each sink.

We study these metrics as a function of sensor network size. The results are shown in figure 4.

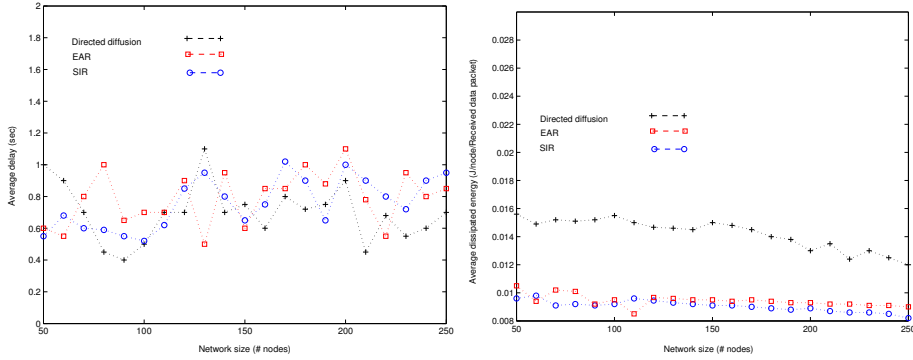


Fig. 4. Average latency and average dissipated energy in a scenario with no simultaneous node failure.

Experiment #2: 20 % simultaneous node failures. The purpose of this experiment is to evaluate the introduction of AI techniques in a scenario where there is a 20 % of simultaneous node failures. This means that at any instant, 20 % of the nodes in the network are unusable because of noise, collision or battery failure influence.

To simulate these situations we create a WSN with 250 nodes. Amongst all of them, we select 20 % of the nodes (50) to introduce one of the following effects:

- S/N ratio degradation. Due to battery energy loss, the radio transmitter power decays. Consequently, the S/N ratio in its neighbors radio receivers is degraded, causing no detections with a certain probability, P . In this situation, we can assume that the node affected by the lack of energy is prone to failure with probability P .

- In many actual occasions, sensor nodes are exposed to high level of noise, caused by inductive motors. Furthermore, the radio frequency band⁷ is shared with other applications that can interfere with our WSN.

In these scenario we analyze the problem studied described in experiment #1 with the three paradigms related. The results are shown in figure 6.

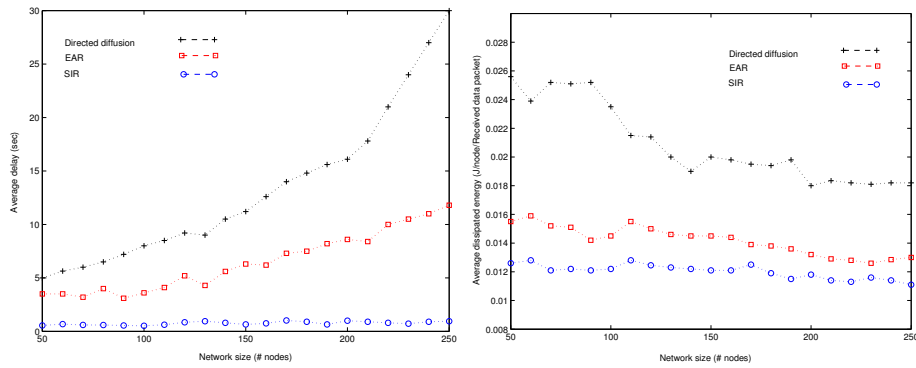


Fig. 5. Average latency and average dissipated energy in a scenario with 20 % simultaneous node failures.

Experiment #3: 40 % simultaneous node failures. This experiment simulates a scenario with a 40 % of simultaneous node failures.

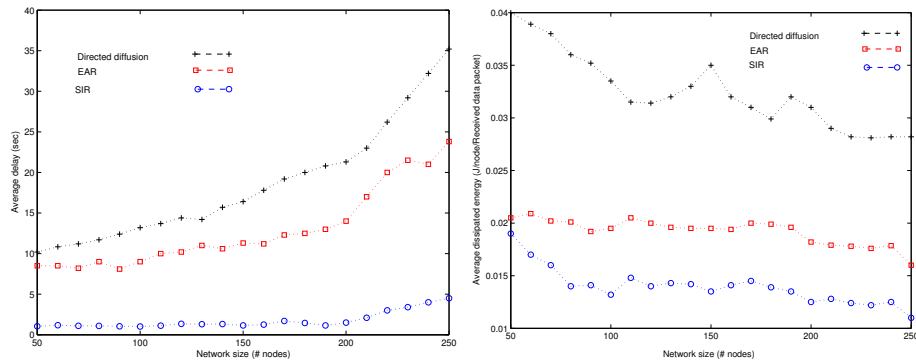


Fig. 6. Average latency and average dissipated energy in a scenario with 40 % simultaneous node failures.

⁷ The use of this band is regulated in Europe by the European Conference of Postal and Telecommunications Administrations (CEPT) and the European Telecommunications Standards Institute (ETSI) by the technical standard *EN 300 220-1*.

5 Conclusion and future works

After comparing the results obtained with every routing paradigm, we can conclude that the differences are important when there is a significant percentage of node failures. Thus, while the average delay goes up with the number of sensors in directed diffusion and EAR, it maintains a low level of delay in SIR. The cause of this effect can be found in the fact that while directed diffusion and EAR elect the intermediate nodes using rules based on the propagation of the interest, SIR elects the intermediate nodes running an AI-algorithm. Thus, the path created by SIR avoids the election of intermediate nodes that are prone to failure because of battery draining, interference or noisy environment. Furthermore, the average dissipated energy is less in SIR when the number of nodes in the sensor goes up. We again find the reason in the effect of the election of the intermediate nodes in SIR. The use of AI in every sensor dynamically varies the assignment of this node role, distributing the energy consumption through the network. When the number of nodes is increased, the number of possible paths is increased too. Furthermore, when the percentage of node failures goes up (from 20 % to 40 %) SIR becomes the best suited protocol for these kinds of scenarios.

Although the results obtained with the inclusion of AI techniques in WSN are important and encouraging, we must take in account some relevant remarks:

- What is the price WSNs have to pay for introducing AI techniques? Although the computational payment for implementing the neural network in a sensor is inapreciable, as detailed in table 1, the tradeoff associated with this implementation is the increase of the overhead. However, in typical SCADA applications, WSNs don't have to attend high level of data traffic. Consequently, the network can support an increase on the overhead.
- Nodes failures can be provoked by the following reasons:
 - Sensor battery draining.
 - Noise originating at industrial environments.
 - Interference in the sensor surroundings.

These phenomena provoke an influence on the average dissipated energy.

SIR has been presented in this paper as an innovative QoS-driven routing algorithm based on artificial intelligence. This routing protocol can be used over wireless sensor networks standard protocols, such as IEEE 802.15.4 and Bluetooth[®], and over other well known protocols such as *Arachne*, SMACS, PicoRadio, etc.

The inclusion of AI techniques (e.g. neural networks) in wireless sensor networks has been proved to be an useful tool to improve network performances.

The great effort made to implement a SOM algorithm inside a sensor node means that the use of artificial intelligence techniques can improve the WSN performance. According to this idea, we are working on the design of new protocols using these kinds of tools.

References

1. E. Çayirci, T. Çöplü, and O. Emiroğlu. Power aware many to many routing in wireless sensor and actuator networks. In E. Çayirci, Ş. Baydere, and P. Havinga, editors, *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pages 236–245, Istanbul, Turkey, February 2005. IEEE, IEEE Press.
2. I.F. Akyildiz, Y. Su, W. Sankarasubramaniam, and E. Çayirci. Wireless sensor networks: A survey. *Computer Networks, Elsevier*, 38:393–422, December 2002.
3. H. Karl and A. Willig. A short survey of wireless sensor networks. TKN, Technical Report Series, Berlin, October 2003.
4. J. Barbancho, F.J. Molina, D. León, J. Roperó, and A. Barbancho. OLIMPO, an ad-hoc wireless sensor network simulator for public utilities applications. In E. Çayirci, Ş. Baydere, and P. Havinga, editors, *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pages 419–424, Istanbul, Turkey, February 2005. IEEE, IEEE Press.
5. F.J. Molina, J. Barbancho, and J. Luque. Automated meter reading and SCADA application. *Lecture Notes in Computer Science, Springer Verlag*, 2865:223–234, October 2003.
6. E.M. Royer and C-K Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, 4(2):46–55, 1999.
7. J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: A survey. *Wireless Sensor Networks, IEEE Wireless Communications*, 4:6–28, December 2004.
8. G. Calinescu and P. Wan. Range assignment for high connectivity in wireless ad hoc networks. *Lecture Notes In Computer Science, Springer Verlag*, 2865:235–246, October 2003.
9. K. Aspnes, D. Goldenberg, and Y. Yang. On the computational complexity of sensor network location. *Lecture Notes In Computer Science, Springer Verlag*, 3121:235–246, July 2003.
10. H. Stratil. Distributed construction of an underlay in wireless networks. In E. Çayirci, Ş. Baydere, and P. Havinga, editors, *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pages 176–187, Istanbul, Turkey, February 2005. IEEE, IEEE Press.
11. S. Saginbekov and I. Korpeoglu. An energy efficient scatternet formation algorithm for bluetooth-based sensor networks. In E. Çayirci, Ş. Baydere, and P. Havinga, editors, *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pages 207–216, Istanbul, Turkey, February 2005. IEEE, IEEE Press.
12. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM Mobicom 2000*, pages 56–67, Boston, MA, USA, 2000.
13. R.C. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proceedings of IEEE WCNC*, pages 17–21, Orlando, FL, USA, 2002.
14. J. Barbancho, C. León, F.J. Molina, and A. Barbancho. SIR: A new wireless sensor network routing protocol based on artificial intelligence. *Lecture Notes in Computer Science, Springer Verlag*, 3842:271–275, January 2006.
15. C. Aurrecoechea, A.T. Campbell, and L. Hauw. A survey of QoS architectures. *Multimedia Systems, Springer-Verlag*, 6:138–151, 1998.
16. B. Sabata, S. Chatterjee, M. Davis, J.J. Sydir, and T.F. Lawrence. Taxonomy for QoS specifications. In *Proceedings of the third International Workshop on Object-Oriented Real-Time Dependable Systems*, pages 100–107. IEEE, IEEE Press, 1997.

17. R. Iyer and L. Kleinrock. QoS control for sensor networks. In *IEEE International Conference on Communications, ICC'03*, volume 1, pages 517–521. IEEE Press, May 2003.
18. J. Kay and J. Frolik. Quality of service analysis and control for wireless sensor networks. In *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 359–368. IEEE Press, October 2004.
19. M.A. Perillo and W.B. Heinzelman. Sensor management policies to provide application QoS. *Ad Hoc Networks*, 1:235–246, 2003.
20. V. Rakocevic, M. Rajarajan, K. McCalla, and C. Boumitri. QoS constraints in bluetooth-based wireless sensor networks. *Lecture Notes In Computer Science, Springer Verlag*, 3266:214–223, 2004.
21. T. Kohonen. The self-organizing map. In *Proceedings of the IEEE*, volume 78, pages 1464–1480, 1990.
22. A. Bensky. *Short-range Wireless Communication. Fundamentals of RF System Design and Application*. Elsevier, second edition, Oxford, UK 2004.
23. W. R. Heinzelman, Al Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of 33rd Annu. Hawaii Int. Conf. on System Sciences*, pages 3005–3014, Hawaii, USA, 2000.