
Julio Barbancho, Carlos León, Javier Molina and Antonio Barbancho

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

Abstract. Currently, Wireless Sensor Networks (WSNs) are formed by hundreds of low energy and low cost micro-electro-mechanical systems. The radio range of this nodes is a few meters. Routing and low power consumption have become important research issues to interconnect this kind of networks. However, conventional Quality of Service routing models, are not suitable for ad hoc sensor networks, due to the dynamic nature of such systems. New features must be analysed to redefine the QoS. This paper describes the concept of routing in sensor networks and introduces a new QoS-driven routing algorithm, named SIR: Sensor Intelligence Routing. Due to the complexity for discovering a path between source and sink we have designed an artificial neural network based on Kohonen self organizing features map. Every node implements this artificial neural network forming a distributed intelligence and ubiquitous computing system. Extensive simulations on our wireless sensor network simulator, OLIMPO, have been carried out to study the efficiency of this new protocol and its performance has been evaluated. The results offer a way to keep on working on.

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

1 Introduction

The deployment of hundreds or thousands of sensor nodes in a wide area, makes possible a huge variety of applications. 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. In this sense, we can not neglect the study of the collision effects and the noise influence.

Many research centers in the whole world (specially in Europe and USA) have focused their investigations in this kind of networks. Erdal Akyildiz 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 features a network communication system should have. A description of the defined network topology is given. Section 3, introduces our specific QoS-driven routing algorithm based on artificial intelligence. This algorithm performance is evaluated by simulation in section 4. Concluding remarks and future works are given on section 5.

2 Network design issues and system architecture

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. Among all these aspects we discuss, in this section, topology and scalability aspects and routing issues, in section 3.

The protocol stack proposed by our research group is based on the OSI model, as depicted in figure 1. In the lower layers we can use the well known IEEE wireless sensor network standard 802.15.4 or our own protocol Arachne. 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 [4], we would like to 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 study this problem [5],[6]. In this section, we model this scenario in which sensors are working, and in section 3 we formalize the routing algorithm proposed to solve this problem, SIR.

2.1 Sensor Network Topology

Due to the desire to cover a large area, a communication strategy is needed. There are different communication paradigms to solve this problem. In GSM1,

1 Global System Mobile.
Fig. 1. Protocol Stack: Arachne and IEEE 802.15.4.
the topology is based on a cellular organization. Every base station forms a vertex of an hexagonal region. This topology has two main features: cell perimeter - area covered by one station ratio is minimum [7], as we can see in figure 2.a (furthermore, the number of handovers is decreased when a mobile station moves through another cell); and, base stations use sectorial antennas with a beam width of 60 degrees per each antenna. Consequently, every base station needs six antennas to cover the whole cell.

![Hexagonal topology](image)

**Fig. 2.** Hexagonal topology (a) in a GSM network and (b) in our wireless sensor network.

Many studies approach the problem of high connectivity in wireless ad hoc networks [8], [9], [10]. In our research we consider a cellular paradigm based on a hexagonal structure like in GSM, although other topologies (random, rectangular, etc.) can be used. However, due to the need of connectivity among nodes, we have modified this topology in the following way:

- Every hexagon’s apothem has been reduced in such a way that we can establish communications among cells allocating a node at every vertex of the hexagon, as it is shown in figure 2.b. With this scheme, we assume that every node can work as a gateway, routing data to a faraway destiny.
- We have consider omnidirectional antennas. This implies that every node needs only one antenna.

Though our topology guarantees a higher connectivity, we have to consider the problem of scalability, introduced in 2.2.
2.2 Scalability

There are several studies [11],[12] related with the optimal density of nodes in a wireless sensor network. This variable depends on the applications the network can support. However, a high density is expected in this kind of system. Bulusu et al. provided in [13] a method to estimate the value of the density of sensor nodes, \( \mu \), in the network:

\[
\mu(R) = \frac{N\pi R^2}{A}
\]

where \( N \) is the number of scattered sensor nodes in region \( A \); and \( R \), the radio transmission range.

\( \mu \) gives an idea about how many nodes there are within the transmission radius of every node.

3 SIR: Sensor Intelligence Routing

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\(^2\). In this situation, the problem is solved by a technique called network backbone formation.

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. Informally, a graph is a set of objects called vertices (or nodes) connected by links called edges (or arcs) which can be directed (assigned a direction). Typically, a graph is designed as a set of dots (the vertices) connected by lines (the edges). A graph structure can be extended by assigning a weight to each edge, or by making the edges to the graph directional (\( A \) links to \( B \), but \( B \) does not necessarily link to \( A \)), technically called a digraph or a directed graph. A digraph with weighted edges is called a network.

An undirected graph or graph \( G \) is an ordered pair \( G := (V, E) \) with \( V \), a set of vertices or nodes, \( v_i \), and \( E \), a set of unordered pairs of distinct vertices, called edges or lines.

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.

\(^2\) In WSN, we often consider two kind of nodes, base stations and sensor nodes. There is usually only one base station.
In 1959, E. Dijkstra proposed an algorithm that solves the single-source shortest path problem for a directed graph with nonnegative edge weights, [14]. Unlike Dijkstra’s algorithm, another algorithm proposed by Richard Bellman and Lester R. Ford [15] can be used on graphs with negative edge weights, as long as the graph contains no negative cycle reachable from the source vertex \( s \). The presence of such cycles means there is no shortest path, since the total weight becomes lower each time the cycle is traversed.

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 this kind 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, SIR. 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 defined \( \Gamma_p(v_j) \), if \( v_j \in p \), as the subset of nodes which are predecessors or successors of node \( v_j \).

![Fig. 3. Graph terminology.](image-url)

We also assume that \( V = \{ r, v_i \} \), and that there is a subset of \( V \), \( T \), defined as \( T := V \setminus \{ r \} \). Furthermore, we can denote \( \overline{T} \) as the complementary set of \( T \), \( \overline{T} = \{ r \} \).

With this terminology, our algorithm can be described as follows in table 1.

### 3.2 Quality of Service in Wireless Sensor Networks

Once it is designed the backbone formation algorithm, we have to define the way of measuring the edge weight parameter, \( w_{ij} \). On a first approach we can
Table 1. SIR algorithm.

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

Step 2: Find a \( v_j \in T \) such as \( d(v_j) = \min \{ d(v_i) | v_i \in T \} \)
Do \( T = T - \{ v_j \} \)
Step 3: \( \forall v_i \in T \cap \Gamma(v_j) \) calculate \( t_i := d(v_i) + w_{ji} \)
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

assume that \( w_{ij} \) can be modelled with the number of hops. According to this assumption \( w_{ij} = 1 \) \( \forall i, j \in R, i \neq j \); and, for the example depicted in figure 3, \( d(v_k) = w_{ri} + w_{ij} + w_{jk} = 3 \). 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 is \( p' \), instead \( p \) (fig. 3). It is necessary to modified \( w_{ij} \) to solve this problem. The evaluation of the QoS in a specific area can be used to modified 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 [16], [17]. 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 in such a way that sensors participate equally in the network while conserving energy and maintaining the required application performance.

What is sensor network QoS? There are a variety of possible definitions. Ranjit Iyer and Leonard Kleinrock proposed in [18] 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 [19]. The same idea is discussed in [20] by Mark Perillo et al., and in [21] 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 define to measure the related QoS parameters.

Once a node has tested a neighbor link QoS, it calculates the distance to root using the obtained QoS value. The expression 2 represents the way a node \( v_i \) calculates the distance to root through node \( v_j \), where \( qos \) is a variable which
value is obtained as an output of a neural network. This tool is described in section 3.3.

\[ d(v_i) = d(v_j) \cdot qos \]  

(2)

3.3 SOM: Self Organizing Map

One of the most powerful mechanisms developed in AI is the Self-Organizing Map (SOM) model [22], 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 4). The first one is the input or sensorial layer, formed by \( m \) neurons, one per each input variable. This neurons work as buffers distributing the information sensed in the input space. The input is formed by stochastic samples \( x(t) \in \mathbb{R}^m \) from the sensorial space. The second layer is formed by a rectangular grid with \( n_x \times n_y \) neurons. Each neuron \((i, j)\) is represented by a \( m \)-dimensional weight or reference vector called \textit{synapsis},

\[ w'_{ij} = [w'_{ij1}, w'_{ij2}, \ldots, w'_{ijm}], \]

where \( m \) is the dimension of the input vector \( x(t) \), as depicted in figure 4. 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:

\[ \Theta(g) \]

\[ x(t) = [\text{latency}(t), \text{throughput}(t), \text{error-rate}(t), \text{duty-cycle}(t)] \]

\[ \text{Fig. 4. SOM architecture.} \]

\textbf{Learning phase:} In the learning 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, \( d(w'_i, x) = \min_{i,j} \{d(w'_ij, x)\} \). Consequently, only the winning neuron is allowed to learn; and its synaptic-weight-vector is updated.

In the learning phase, weight-vectors and topological neighbors move closer to the input vector in the input space. A learning rate governs the size of the weight alterations. The update for the weight-vector of a neuron \((i, j)\) is denoted by \( w'_{ij}(t + 1) = w'_{ij}(t) + \alpha(t) \cdot h(|i - g|, t) \cdot \left( x_k(t) - w'_{ijk}(t) \right) \), where \( t \) denotes the time, \( \alpha(t) \) the learning rhythm and \( h(\cdot) \) the neighborhood function. \( h(\cdot) \) declares which are the neighbors of the current winning neuron, \( g \).

The effect of the unsupervised learning algorithm -also considered to be a weight update algorithm- is to distribute the neurons throughout the area of the \( n \)-dimensional space populated by the actual training set according to the distribution of the data.

**Execution phase:** The weights are declared fixed.

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

Next, it is declared a winning neuron, \( g = (g_1, g_2) \), with a synaptic-weight-vector, \( w'_{gj} \), similar to the input \( x \). Table 2 shows this phase implementation used at OLIMPO.

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

**Table 2.** Implementation of the winning neuron election in C++.

```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,
        d2 += aux*aux;
    d[0] = sqrt(d2);
    for (int j=0; j < 4; j++)
        float aux = d[j];
    if (aux>d[0])
        aux = d[0];
    neuron = n+1;
    return neuron;
}
```


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 [23], [24]. Every node in OLIMPO implements a neural network (SOM) running the algorithm detailed in table 2 (online processing).

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 model are integrated in the channel object of the simulation tool.

Propagation models attempt to predict he average received signal strength at a given distance from the transmitter, as well as the variability of the signal strength in close proximity to a particular location. In modelling radio wave propagation we distinguish between large and small scales.

Large-scale effects. On large scales, the interesting question is how the signal from the transmitter reaches the receiver in the first place and what is the power received relative to power transmitted. There is typically no direct line-of-sight path between the communication parties. Furthermore, we rely on various electromagnetic propagation mechanisms, such as reflection, diffraction and scattering to enable the radio communication. Sometimes we may receive both, a direct signal and another reflected signal, which interfere with each other.

The simplest propagation model is free space propagation, where there is a single unobstructed communication path. Free space propagation is a useful model for satellite and line-of-sight microwaves links. In this case, the Friis [25] free space equation (eq. 3) is used.

\[ P_r = P_t \left[ \frac{\lambda}{4\pi d L} \right]^2 G_t G_r \]  

This equation approximates the received signal power, \( P_r \), 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 \( \lambda \) is the electromagnetic wavelength. In radio communications we have to consider the receiver has a power reception sensitivity \( P_s \) and there is a power reception threshold \( P_h \) for received packet. Where \( P_h \geq P_s \).

When a node receives a signal above \( P_h \) it process the packet; this determines the transmission radius. If the received signal is above \( P_s \) but below \( P_h \), it senses the channel as busy but is unable to process the packet; this determines the interference range.

Given that a transmitter sends a signal power \( P_t \), we can define the transmission radius \( R_t \) as the distance a signal with power \( P_t \) travels until the
power decays to $P_h$; and the interference radius $R_s$ as the distance a signal travel until the power decays to $P_s$.

Using the Friis free space model (eq. 3) it is easy to obtain the following expression:

$$\frac{R_t}{R_s} = \sqrt{\frac{P_s}{P_h}}$$

Typically we choose a reference distance $d_0$ and express $P_r$ as a function of the distance, so that $P_r(d) = P_{r_0}(d/d_0)^{-n}$. The symbol $n$ called in the literature as path loss exponent. In this case $n = 2$, but when obstructions are present, $n$ is assigned a larger value, usually between 2 and 4, [25].

When there is no line-of-sight link, we must rely on other mechanisms:

- Reflection occurs when the incoming wave meets a large surface with certain optical properties.
- Another phenomenon is called as diffraction. Due to this phenomenon wave can be propagated around edges or beyond the horizon.
- The last large-scale effect we have to consider is the scattering phenomenon.

**Small-scale effects.** On smaller scales we consider other problems on the reception of the radio signal, such as fading, because of the reception of multipath waves; Doppler shift, because of relative motion between the transmitter and the receiver; and Rayleigh or Ricean fading models, because of changes in the transmitted signal parameters [25].

### Table 3. Values of radio communication parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio frequency band$^1$</td>
<td>869.7 - 870 MHz</td>
</tr>
<tr>
<td>Communication bandwidth$^1$</td>
<td>25 kHz</td>
</tr>
<tr>
<td>Number of radio channels$^1$</td>
<td>1</td>
</tr>
<tr>
<td>Antenna gain$^1$</td>
<td>$G_r = 1$, $G_t = 1$</td>
</tr>
<tr>
<td>Radio transmitter power$^1$</td>
<td>$P_t = 5$ mW</td>
</tr>
<tr>
<td>Radio receiver sensibility$^1$</td>
<td>$P_s = -101$ dB</td>
</tr>
<tr>
<td>System loss$^2$</td>
<td>$L = 1$</td>
</tr>
<tr>
<td>Path loss exponent$^2$</td>
<td>$n = 2$</td>
</tr>
<tr>
<td>Large-scale effects</td>
<td>Not considered</td>
</tr>
<tr>
<td>Small-scale effects</td>
<td>Not considered</td>
</tr>
<tr>
<td>Modulation</td>
<td>Analog FM</td>
</tr>
<tr>
<td>Transmission rate</td>
<td>4800 b/s</td>
</tr>
</tbody>
</table>

$^1$Based on licensed free standard ETSI EN 301 291
$^2$Antennas are assumed to be omnidirectional.

For the purpose of this research, the values showed in table 3 have been considered. In this scenario, two sensor nodes attempting to establish a radio communication link can be 218 meters separated$^3$. According to the topology proposed in 2.1, the hexagon apothem must be reduced below the radio range. In our simulations we have assumed that the distance between every pair of sensor nodes is set up to 150 meters. With this scheme a node has six neighbors located at the same distance from it. Due to $P_s = P_h$ and according to eq. 4, $R_t = R_s$.

Consequently, in an hexagonal cell all the neighbors interfere themselves, in such a way that every node has to contend for channel access with other six nodes, as depicted in figure 2.b. We have focused our simulation on a wireless sensor

$^3$Applying these values to eq. 3.
network composed by 4000 nodes covering an area of 87 Km$^2$. This is the typical area of a European medium size city like Seville (Spain) or Zurich (Switzerland). Applying this results on equation 1, with $A = 87$ Km$^2$, $R = 218$ meters and $N = 4000$ we obtain a density around 7, which is the number of nodes which are within the transmission radius of a node.

4.2 Noise influence

Noise influence over a node has been modelled as an Additive Gaussian White Noise, (AWGN). According to $P_s = P_h$ we have considered the noise power, $N_o$, will provoke a noisy effect in receptors, only if $N_o \geq P_s$.

Noise power has been modelled as a stochastic variable with a mean value expressed as a percentage of the antenna sensibility $P_s$; and a standard deviation expressed as a percentage of its mean value.

4.3 SOM design

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, as depicted in figure 4.

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 link communication between a pair of sensor nodes can work. For that reason we have to create the special environments. These scenarios are implemented by different noise simulations. In our research we created a WSN over OLIMPO composed by 4000 sensor nodes. In this network, we chose a pair of nodes (let us denote them as $v_{800}$ and $v_{1250}$) and introduced a low power noise into one of them (e.g. $v_{1250}$).

According to the input requirements, we had to measure the QoS metrics. In that sense, we ran a ping application 50 times at node $v_{800}$. This application pings are sent from node $v_{800}$ to node $v_{1250}$. Ping requires acknowledgment (ACK). The way node $v_{800}$ receives ACKs will determine a specific QoS environment, expressed on the four metrics elected: latency (seconds), throughput (bits/sec), error rate (%) and duty cycle(%). For example, for a mean noise power of 40 % $P_s$ and a noise standard deviation of 30 % of mean noise, the samples obtained were $[0.58, 1440, 10.95, 2.50]$. This process was repeated 100 times while increasing the noise power.

With the set of 100 input samples we trained our neural network using the learning algorithm detailed in section 3.3. This process was implemented on a personal computer using the MATLAB® neural toolbox (offline processing).

Once we had ordered the neurons on the Kohonen layer, we identified each one of the set of 100 input samples with an output layer neuron. According to
this procedure the set of 100 input samples were distributed over the SOM. We realized that input samples obtained from a similar noisy environment and with similar QoS features were allocated in a specific region of the SOM. Consequently we obtained a map formed by clusters, where every cluster corresponded with a noise level introduced at the environment and consequently a specific QoS. Furthermore, a synaptic-weight matrix \( \mathbf{w}_{ij} = [w_{ij1}, w_{ij2}, \ldots, w_{ij4}] \) is formed, where every synapsis identifies a connection between input and output layer.

In order to quantify the QoS level we studied every cluster features and assigned a value between 0.2 and 10, according to the level of noise introduced. This assignment was based on our experience as experts in networks. In that way we defined the output function \( \Theta(i, j), i \in [1, 3], j \in [1, 4] \) with twelve values corresponding with every neuron \( (i, j), i \in [1, 3], j \in [1, 4] \).

**Execution phase.** Every sensor node measures the QoS of its links collecting input samples and running the winning neuron election algorithm (table 2). For example, 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 \( \Theta \) to assign a QoS estimation, \( \text{qos} \). Finally this value is employed to modified the distance to root (eq. 2).

### 4.4 SIR performance

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

**Experiment #1:** First, a wireless sensor network with 4000 nodes is created. The network backbone is formed using SIR algorithm, as detailed in table 3. However, no SOM is applied, so, the distance from a node \( v_i \) to root \( d(v_i) \) is not modified by the neighbor link quality, \( \text{qos} \) (eq.2). We have called this algorithm as No AI algorithm.

Next, a high level of noise is introduced at nodes \( v_{100} \) and \( v_{200} \), figure 5.c. Finally, a specific node (e.g. node number \( v_{300} \)) runs the ‘Transmit clock to base station’ application (figure 1). Node \( v_{300} \) sends 10 packets to root to measure the latency. Every packet contains ‘clock’ information.

Figure 5.a represents clock latency depending on the level of the noise power introduced at nodes \( v_{100} \) and \( v_{200} \). When the mean noise is above the antenna sensibility and the noise standard deviation has a low value, clock latency grows up rapidly, decreasing the QoS.

**Experiment #2:** This experiment is similar than experiment #1, but in this case, the distance \( d(v_i) \) is modified by the neighbor link quality, using equation 2. The network backbone is formed in such a way that the path created from the node \( v_{300} \) to root does not contain the noisy nodes, figure 5.d. Figure 5.b shows SIR algorithm performance compared with No AI algorithm performance. As depicted in this figure, if the mean noise is low, both algorithm performances are excellent. In this case, the path from the node \( v_{300} \) to root contains nodes \( v_{100} \) and \( v_{200} \). However, when the mean noise
Fig. 5. (a) Clock latency measurement: No AI performance (b) Clock latency measurement: SIR versus No AI (c) Network backbone formation based on No AI algorithm (d) Network backbone formation based on SIR algorithm.
grows up above the antenna sensibility SIR algorithm performance improves. No AI algorithm performance, maintaining the QoS. In this case, the path from the node $v_{300}$ to root does not contain the noisy nodes.

5 Conclusion and future works

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, EAR, LEACH, 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. An additional advantage is the low cost the AI implementation represents, as shown in table 2.

The great effort made to implement a SOM algorithm inside a sensor node means that our research group has become a pioneer on the use of artificial intelligence techniques in WSN. According to this idea, we are working on the design of new protocols using this kind of tools.

References


