Using Artificial Intelligence in Wireless Sensor Routing Protocols

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

Abstract. This paper represents a dissertation about how an artificial intelligence technique can be applied to wireless sensor networks. Due to the constraints on data processing and power consumption, the use of artificial intelligence has been historically discarded in these kind of networks. 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.

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

Goals like efficient energy management, high reliability and availability, communication security, and robustness have become very important issues to be considered in wireless sensor networks (WSN). This is one of the many reasons why we can not neglect the study of the collision effects and the noise influence.

We present in this paper a new routing algorithm which introduces artificial intelligence (AI) techniques to measure the quality of service (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 topology. 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.

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 [1], [2]. In our research we consider a random distribution of sensors.

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 [3], 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 [4] 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

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

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.

SOM is an unsupervised neural network. The neurons are organized in an unidirectional two layers architecture. The first one is the input or 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 $\boldsymbol{x}(t) \in \boldsymbol{\mathcal{R}}^m$ from the sensorial space. The second layer is usually formed by a rectangular grid with nxx$n'y$ neurons. Each neuron (i, j) is represented by an m-dimensional weight or reference vector called *synapsis*, $\boldsymbol{w'}_{ij} = [w'_{ij1}, w'_{ij2}, \ldots, w'_{ijm}]$, where m is the dimension of the input vector $\boldsymbol{x}(t)$. 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: the ***learning phase***, in which, neurons from the second layer compete for the privilege of learning among each other, while the correct answer(s) is (are) not known; and the ***execution phase***, in which every neuron (i, j) calculates the similarity between the input vector $\boldsymbol{x}(t)$, $\{x_k \mid 1 \leq k \leq m\}$ and its own synaptic-weight-vector $\boldsymbol{w'}_{ij}$.

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.

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** [5].

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 [6]. 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.

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.

$$d(v_i) = d(v_j) \cdot qos \tag{1}$$

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 [7]. Every node in OLIMPO implements a neural network (SOM) running the execution phase (online processing).

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 we can determine the signal-to-noise ratio at the detector input. This signal-to-noise ratio can be expressed as an associated BER (Bit Error Rate). 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.

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.

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.

4.1 Learning Phase

In order to organize the neurons in a two dimensional map, we need a set of input samples $\boldsymbol{x}(t)$=[latency(t), throughput(t), error-rate(t), 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 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 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(%). 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. 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 $\boldsymbol{w'}_{ij} = [w'_{ij1}, w'_{ij2}, \ldots, w'_{ij4}]$ is formed, where every synapsis identifies a connection between input and output layer.

In order to quantify the QoS level, we 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, e 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.

4.2 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. 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.*

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 were 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 declare 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: the average dissipated energy, which computes the average work done by a node a in delivering useful tracking information to the sinks (this metric also indicates the overall lifetime of sensor nodes); and the average delay, which 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 figures 1.a and 1.b.

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

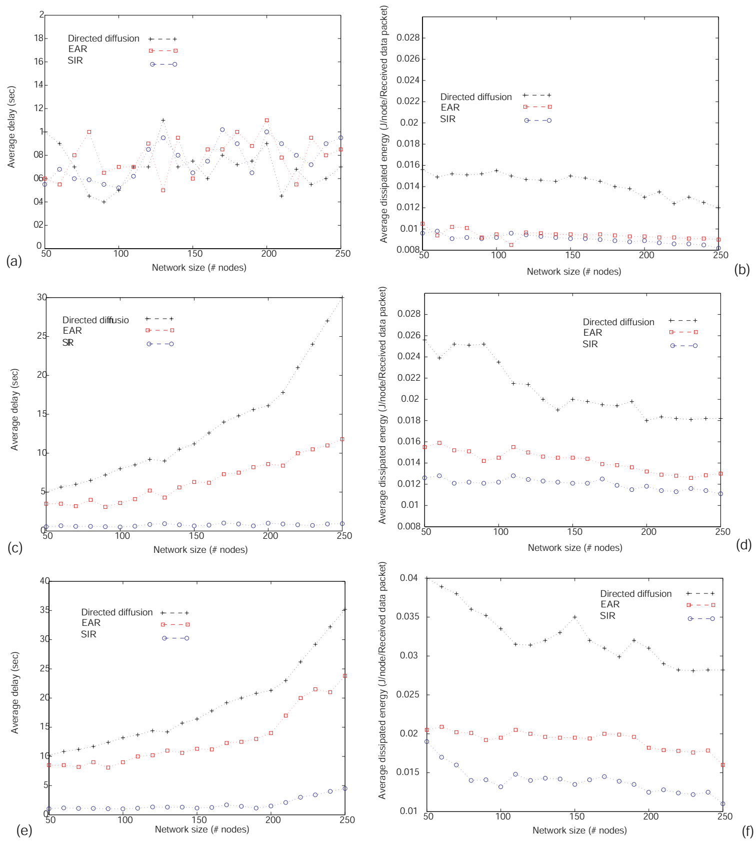

Fig. 1. Average latency and average dissipated energy in a scenario with no simultaneous node failure [(a) and (b)]; with 20 % simultaneous node failures [(c) and (d)]; and with 40 % simultaneous node failures [(e) and (f)]

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 band 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 figures 1.c and 1.d.

Experiment #3: 40 % simultaneous node failures. This experiment simulates a scenario with a 40 % of simultaneous node failures. The results are shown in figures 1.e and 1.f.

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, 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. K. Aspnes, D. Goldenberg, and Y. Yang. On the computational complexity of sensor network location. *Lectune Notes In Computer Science, Springer Verlag*, 3121:235–246, July 2004.
2. S. Saginbekov and I. Korpeoglu. An energy efficient scatternet formation algorithm for bluetooth-based sensor networks. In E. Çayircy, Ş. Baydere, and P. Havinga, editors, *Proceedings of the Second Europen Workshop on Wireless Sensor Networks*, pages 207–216, Istanbul, Turkey, February 2005. IEEE, IEEE Press.
3. 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.
4. R.C. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proceeedings of IEEE WCNC*, pages 17–21, Orlando, FL, USA, 2002.
5. 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.
6. 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.
7. J. Barbancho, F.J. Molina, D. León, J. Ropero, and A. Barbancho. OLIMPO, an ad-hoc wireless sensor network simulator for public utilities applications. In E. Çayircy, Ş. Baydere, and P. Havinga, editors, *Proceedings of the Second Europen Workshop on Wireless Sensor Networks*, pages 419–424, Istanbul, Turkey, February 2005. IEEE, IEEE Press.