

Research Article

Designing Energy-Efficient Topologies for Wireless Sensor Network: Neural Approach

Chiranjib Patra,¹ Anjan Guha Roy,² Samiran Chattopadhyay,² and Parama Bhaumik²

¹ Calcutta Institute of Engineering and Management, 24/1A Chandi Ghosh Road, Tollygunge, Kolkata 700040, India

² Department of Information Technology, Jadavpur University, Kolkata 700032, India

Correspondence should be addressed to Chiranjib Patra, chiranjibpatra@gmail.com

Received 3 August 2010; Accepted 1 December 2010

Copyright © 2010 Chiranjib Patra et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Preserving energy or battery power of wireless sensor network is of major concern. As such type of network, the sensors are deployed in an ad hoc manner, without any deterministic way. This paper is concerned with applying standard routing protocols into wireless sensor network by using topology modified by neural network which proves to be energy efficient as compared with unmodified topology. Neural network has been proved to be a powerful tool in the distributed environment. Here, to capture the true distributed nature of the Wireless Sensor Network (WSN), neural network's Self-Organizing Feature Map (SOFM) is used.

1. Introduction

Sensors in the Wireless Sensor Networks mainly use batteries. Very often these batteries are nonrechargeable or nonreplaceable due to the geographical location of those sensors. Therefore, energy preservation of sensor nodes is a crucial issue to reduce the quick exhaustion of the energy of sensor nodes and thereby to prolong the overall network lifetime. The limited resources of the sensor nodes need to be spent judiciously so that it requires the minimum energy for this energy-consuming task. Several techniques have been proposed so far, which emphasize the energy-efficient routing protocol. Most of them tend to utilize the same single optimal path for each communication time [1, 2]. A straight line routing protocol has been mentioned in [3] which achieves routing in WSN without broadcasting. But using only a single path is prone to node and link failures because of the depletion of batteries. In case of failure due to any reason, an alternative route has to be discovered for maintaining the continuous transmission from source to destination. This technique cost some extra energy for the route discovery process. Using multiple paths in wireless sensor network can enhance the overall efficiency, reliability, and integrity of the network. It can prove to be an effective way of even distribution of traffic load over the network. Most of the multiple paths routing protocol till date are based

on the classic on-demand single path routing methods [4, 5] such as AODV and DSR. They have their own technique of selecting multiple routes. Some papers are concerned with the node energy while constructing multiple paths [6, 7].

But all these approaches mentioned above are well applicable for static topologies and have to pass through some common problems. They flood the route request to the network at a time over the whole network, thus increasing the overhead and probability of route congestion becomes high. They also unnecessarily waste the sensor node's energy. Moreover, when several alternative paths transmit data packets arbitrarily, there exists a probability of high packet loss rate even if node-disjoint multipaths are used and which results in a suboptimal Computation-Communication tradeoff. For dynamic topology an efficient distributed method to form a weighted connected dominating set (the backbone) could be achieved through constant approximation ratio on cost optimization [8]. Whereas some researchers believe the shape of the topology is important as topology control for 3D sensor networks; it is proved that some of the structures could guarantee the power efficiency of all paths.

Any kind of time development (be it deterministic or essentially probabilistic) which is analyzable in terms of probability deserves the name of stochastic process. This process offers a simple, robust, and ultralow-power solution

for many sensor network applications. A “stochastic” sensor network is proposed in [9] in which a sensor node operates normally and consumes stored energy in the wake mode until the energy is depleted, and then ceases processing and reverts back to the sleep mode while scavenging the environment for usable sources of energy. When the recharging process is complete, the node resumes normal operation. This behavior of having asynchronized wake-sleep modes among sensor nodes constitutes stochastic sensor network. One of the biggest problems with stochastic sensor networks are redundant packet transmissions to sustain network traffic via stochastic flooding [10].

Keeping the degree of complexity of the above mentioned approach in mind, the concept of SOFM from neural network is taken. The unique property of SOFM is mapping of continuous input space of some certain distribution function to a discrete output space. This discrete output space consists of the modified topology of the network that will be used for information dissemination [11].

The rest of this paper is organized as follows; we briefly discuss Self-Organizing Feature Map (SOFM) network model and assumptions in Section 2 and algorithm formulation in Section 3. Here we also have a target system which will be discussed in Section 4 and have discussed implementation details through proper interpretation of the output graphs from PROWLER in Section 5. Finally we conclude in Section 6, pointing out future directions of research.

2. SOFM Network Model and Assumptions

The network under consideration consists of N number of sensors nodes scattered over 2-dimensional space. For network analysis with Kohonen’s Self-organizing map, let us assume that the Kohonen’s layer consists of N neurons. Here we represent a neuron of the neural network to a sensor node of wireless sensor network (WSN). In this context, we will use the terms neurons and sensor nodes to describe the same thing. Furthermore we assume the two-dimensional lattice of Kohonen’s map represent the area of a wireless sensor network. Input spatial data of the 2-dimensional space as described is assumed to follow Poisson’s distribution function for obvious reasons [10]. From the concept of Kohonen’s Self-organizing feature maps (SOFM) these input spatial data act as input vectors learn to classify according to how they are grouped in the input space. They differ from competitive layers in that neighboring neurons in the self-organizing map learn to recognize neighboring sections of the input space. Thus, self-organizing maps learn both the distribution (as do competitive layers) and topology of the input vectors they are trained on.

Learning in a self-organizing feature map (SOFM) occurs for one vector at a time, independent of whether the network is trained directly or whether it is trained adaptively.

First the network identifies the winning neuron. Then the weights of the winning neuron, and the other neurons in its neighborhood, are moved closer to the input vector at

each learning step. The winning neuron’s weights are altered proportional to the learning rate. The learning rate and the neighborhood distance used to determine which neurons in the winning neuron’s neighborhood are altered during training.

Thus through Kohonen’s learning the winning neuron will be selected which will be treated as a speaker node for a region.

Now this speaker node of described network has a specific coverage region. All the neighbor nodes reside at a certain distance from a speaker node. It is found during the cooperative process of SOFM that the topological neighborhood function satisfies the requirement of Gaussian function [11]. So it may be concluded that the speaker node is surrounded by those nodes which fall in the Gaussian range as decided by previous competitive process. Lastly in the synaptic adoption process enables the exited neighboring nodes to increase their individual values of the discriminant function in relation to the input pattern through suitable adjustments applied to their synaptic weights. The adjustments made are such that the response of the speaker nodes to the similar input pattern is enhanced.

Mathematically it can be simplified as follows.

Step 1. Initialization. Choose random values for the initial weights $w_j(0)$.

Step 2. Finding the Speaker. Find the winning unit j^* at using the minimum-distance Euclidean criterion $j^* = \arg \min \|x_j(t) - w_j\|$, $j = 1, \dots, N$, where $x_j(t)$ represents the input pattern, N is the total number of unit, and $\|\cdot\|$ indicates the Euclidean norm.

Step 3. Weights Updating. Adjust the weights of the winner and its neighbors, using the following rule: $w_j(t+1) = w_j(t) + \alpha N_{j^*}(t)(x_j(t) - w_j(t))$, where α is a positive constant and $N_{j^*}(t)$ is the topological neighborhood function of the winner unit j^* at time t . The neighborhood function is traditionally implemented as a Gaussian (bell-shaped) function:

$$N_{j^*}(t) = \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^* \exp \left\{ -\frac{(j^* - j)^2}{2\sigma^2} \right\} \quad (1)$$

with σ a parameter indicating the width of the function, and thus the radius in which the neighbors of the winning unit are allowed to update their prototype vectors significantly. It should be emphasized that the success of the map formation is critically dependent on how the values of the main parameters (i.e., α and $N_{j^*}(t)$), initial values of weight vectors, and the number of iterations are prespecified. The Kohonen’s SOM mainly has implementations based on a single-processor, centralized method.

The adjustments of the speaker nodes produce results that is, spatial coordinates, which are not amongst any of the input coordinates. So in order to map the speaker coordinates with respect to input data we use nearest- k neighbor algorithm. The value of k for nearest neighbor algorithm is

incremented by one iteratively until the redundancies in list of participating nodes are removed.

3. Proposed Algorithm

3.1. Proposed SOFM Topology Building (SOFMTB) Algorithm.

The following are the steps of the SOFM topology building algorithm.

Step 1. Spatial coordinates for sensor distribution on the field is taken as input vectors.

Step 2. SOFM algorithm is used to train the spatial coordinates with number of neurons equal to desired number of speaker nodes at the output.

Step 3. An array of new set points in spatial dimension is returned.

Step 4. Returned array is mapped for real spatial coordinates with the help of K -nearest neighbor algorithm with respect to speaker nodes.

Step 5. The output contains duplicated spatial coordinates, to remove these duplicity, we iteratively run the K -nearest-neighbors algorithm for subsequent values of K until the duplicity is removed.

Step 6. An array of nonduplicated set points in spatial dimension is returned.

Step 7. This topology is presented to the link layer protocol such as spanning tree protocol.

An alternative representation is given in Algorithm 1.

4. Target System

A very successful, low-cost prototype field-node (mote) family was developed at Berkeley. The used variant (MICA) of the Berkeley motes includes an 8-bit, 4 MHz Atmel ATMEGA103 microcontroller, 128 kB program memory, 4 KB RAM, and an RFM TR1000 radio chip capable of providing 50 kbit/s transmission rate at 916.5 MHz. The motes can also accommodate a set of interchangeable sensors (temperature, light, magneto, sound, etc.) [12].

The motes use a small operating system called TinyOS, designed to provide the necessary services in despite of the very limited hardware resources. It contains a complete network stack with bit-level error correction, medium access layer, network messaging layer, and timing.

The Medium Access Control layer uses a simple Carrier Sense Multiple Access protocol: it waits for a random duration before trying to transmit a packet and then waits for a random backoff interval if the channel was found busy. It keeps trying until the transmission can be performed. This simple approach is not as effective as the more sophisticated protocols (e.g., IEEE 802.11, [13]) in terms of collision

```

Algorithm SOFMTB ( Input_Vector )
//Input_Vector file contains spatial coordinates for
//sensor distribution on the field.
{
    Load Invec:= Input_Vector;
    SOFM:= Create_SOM(n, m);
//Define a SOM whose input data points vary from -n to n,
//with m nodes.
    SOFM:= Train(SOFM, Invec);
//Train the SOM with value from Invec.
FOR i:= 1 to m
    Save new set points post training in array x;
    Set Actual_data := Invec;
    Set Ideal_data := x;
WHILE (duplicate coordinate)
    t:= KNearestNeighbor (Actual_data, Ideal_data);
//Ideal data points are mapped with input iteratively by
//means of KNN until the duplicities are removed.
    topology:= t;
    Set Node_IDs from 1 to m;
    Present the topology to link layer protocol such as
spanning tree;
}

```

ALGORITHM 1: Annexure-I.

avoidance, but it certainly consumes less energy and the communication overhead is much smaller.

5. Implementation Details

Due to stochastic nature of the environment, a useful performance metric is typically not the result of a single experiment, but rather an average value, a minimum or maximum. Thus a single function call of the optimizer algorithm can be very expensive. Other problems include no prior knowledge of error surface so that efficient error surface calculation could be determined as the result the number of experiments to be done is not known.

In order to overcome such problems and keeping the considerations of the target system we use PROWLER—PROBABILISTIC WIRELESS NETWORK SIMULATOR V1.25 with a test bed of 50 sensors placed in a matrix of 10×10 sq units, the positions of various sensors were recorded previously and supplied to the simulator as input.

The input spatial distribution of sensors under test consideration as viewed in Matlab 7 as shown in Figure 1 is taken as the input topology file.

The topology file is modified according to the algorithm in Section 3.1. Step 2. Figure 2 describes the outcome.

Now using the Step 4 of Section 3.1 we remap the circle coordinates to the cross-coordinates as shown in Figure 3.

From Figure 3 it is clear that due to some duplicate values the number of remapped values is less than what is expected. Now by using Step 5 of Section 3.1 we can remove these duplicate values and finally obtain unique values from input topology file. Now as described in Step 7 of Section 3.1 we use PROWLER V-1.25 for simulating with Spanning Tree Protocol (STP) with center as root node.

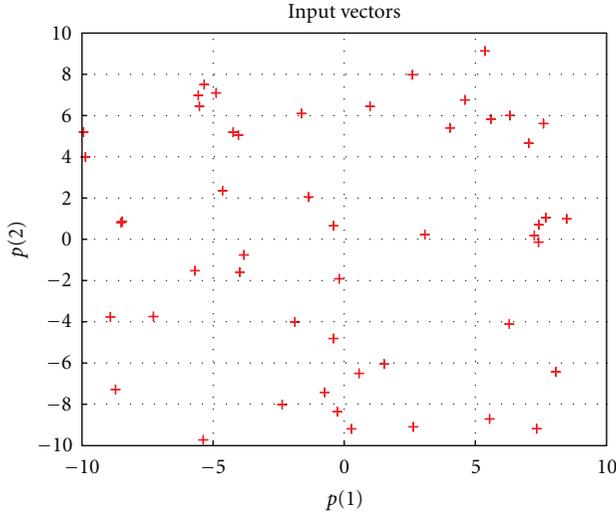


FIGURE 1: Distribution of 50 sensors as plotted in 10×10 test area.

The list of assumptions made while running the simulation on PROWLER-V1.25 [14] are as follows.

- (1) Each node has the following fields in the routing table.

xID: The identifier of the neighbor.

InLink: Quality of the directed link ($xID \rightarrow ID$).

OutLink: Quality of the directed link ($ID \rightarrow xID$).

Hop: the hop-number of mote xID .

Note: Each node is assigned with unique ID, hop number (initially NaN except the root node where its zero).

- (2) Each node wakes up periodically and transmits its *ID*, hop number, and table data. Upon receipt of message from node i , node j updates its own table.
 - (i) Updates the *InLink* property of i .
 - (ii) Updates the *Hop* property of i .
 - (iii) Updates the *OutLink* property of i , if the received table contains information about j (the *InLink* value is used).
- (3) Each node transmits the table data with certain finite probability. The transmission probability is the function of design parameter and the content of the table.
 - (a) Initially $p = P/8$.
 - (b) For all the nodes with a hop-number NaN, $p = P/8$.
 - (c) If the hop-number of the node changes, p is set to P .
 - (d) If a mote j receives a message from node i , indicating that i has no information about j , but j has a good *InLink* property of i , then node j sets $p = P$.
 - (e) After each transmitted message $p = P/2$.

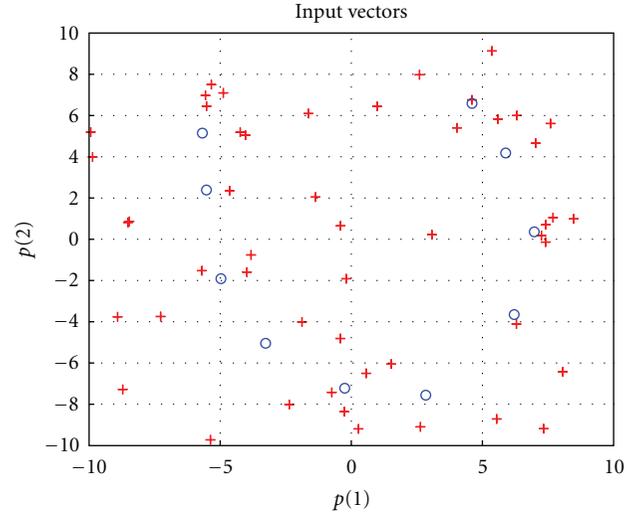


FIGURE 2: Distribution of SOFM coordinates (Circles) in the test space (with 30 neurons in test condition).

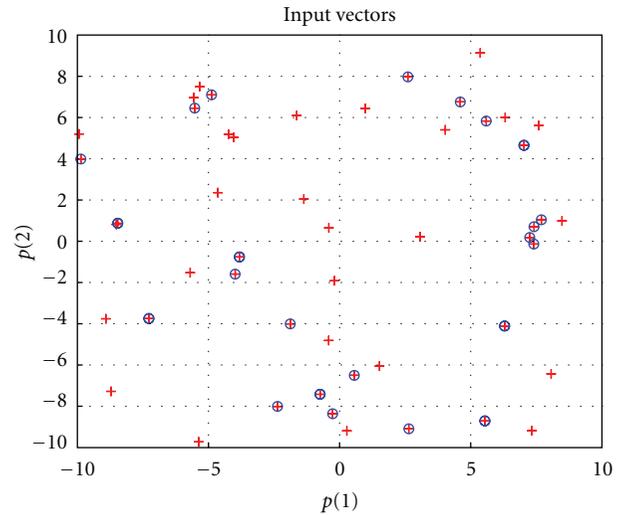


FIGURE 3: Remapped distribution of SOFM coordinates (Circles) in the test space (with 30 neurons in test condition).

Using the above considerations the spanning tree algorithm was run on the test bed; Figures 4-9 are the performance graphs obtained,

In order to interpret them we use performance metric as composed from the number of receiving motes. in the network, the more the motes receive the better and the consumed power which is proportional to the settling time, that is, time to build the spanning tree: the less power is used the better.

Considering Figure 4 we have very small region around $p = 0.3$ and $p = 0.6, 1$, the participating motes are around +90% which shows maximum coverage over the test area and the settling time is around 0.25 seconds which in terms of energy metric is appreciable. But the steady state in the receiving nodes have not yet been achieved.

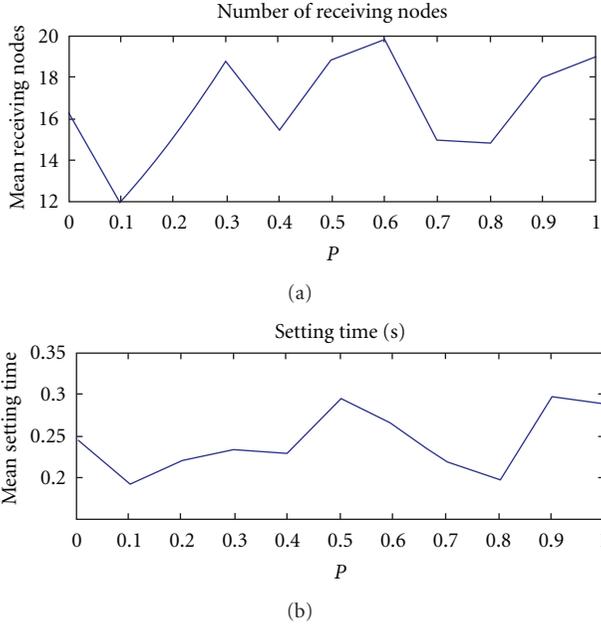


FIGURE 4: The performance graph with 20 neurons in the test condition.

Similar explanation goes for the Figure 5. But considering Figure 6 we can find that there is stable number of receiving nodes for $p = 0.1$ to $p = 0.8$ and the settling time is around 0.3 seconds, and considering Figure 7 we can find that there is stable number of receiving nodes for $p = 0$ to $p = 0.4$ and $p = 0.6$ to $p = 0.1$ and the settling time is around 0.35 seconds; both these cases are good example for a simple tradeoff between energy consumed and the mean receiving nodes. So for the input topology as presented, the saving in terms of nodes is 20% to 30%; these nodes can be used when the depletion of energy in other nodes occurs and hence the life time of the network can be increased. Though saving of about 20%–30% is not enough for any sensing mission but this saved percentage could reduce the number of sensors to be deployed/used for the next time. The savings in terms of nodes is absent if we use all the 50 sensors at a time, the performance characteristics as depicted in Figure 8 shows no steady region for mean receiving nodes versus p as in Figure 6 or Figure 7 but in terms of settling time Figure 7 shows the optimized settling time as compared to Figure 8.

Considering the case for Figure 6 we have the following node ID generated while running the simulation of PROWLER V-1.25 for transmission probability ranging from 0 to 1. Referring to the Figure 6 the node IDs from $p = 0.1$ to 0.8 can be used for information dissemination efficiently over the network covering the test area. The combination from $p = 0.9$ is left out for not proper coverage because less number of participating nodes are seen from Figure 6. Now if we consider time varying usage of the paths as shown in Table 1 like if each p were selected for 5 runs as done in our simulator, we would obtain the node usage (i.e., number of times a node was used for transmission throughout the simulation). The characteristics is as shown in Figure 9.

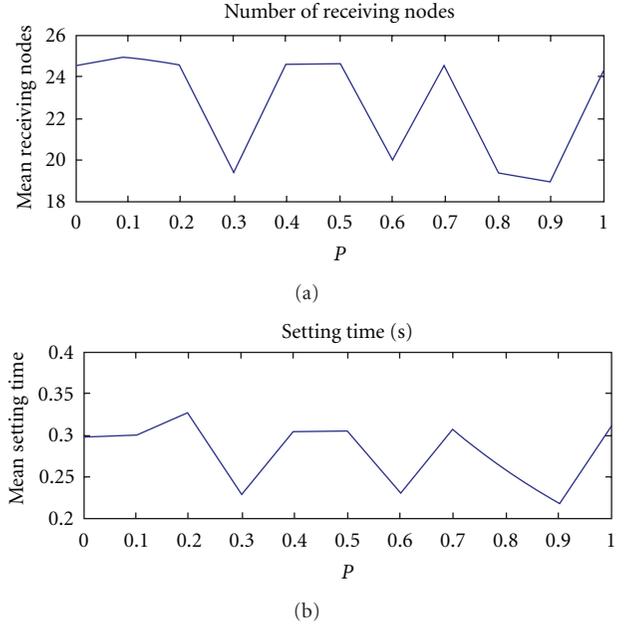


FIGURE 5: The performance graph with 25 neurons in the test condition.

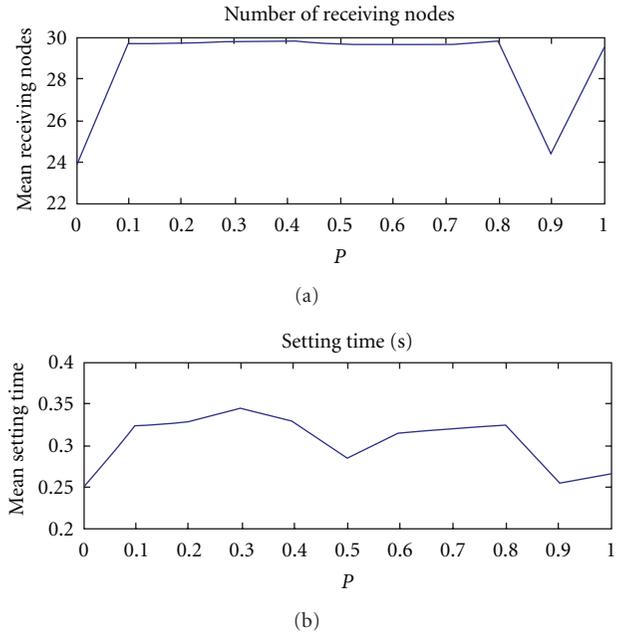


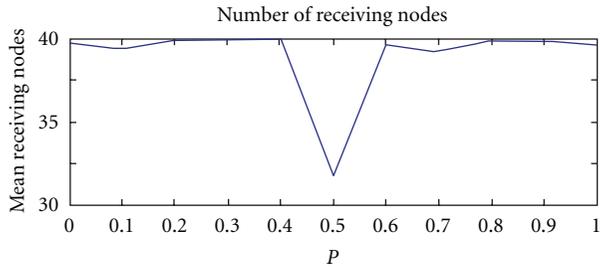
FIGURE 6: The performance graph with 30 neurons in the test condition.

6. Results and Discussions

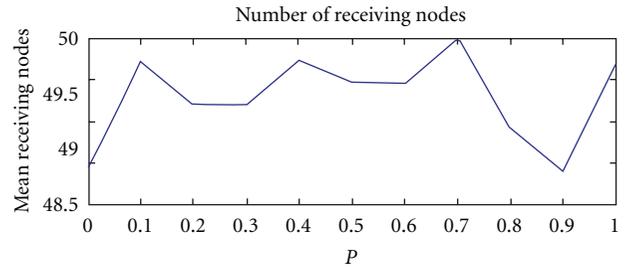
The random distribution of nodes is finally given definite energy-efficient topology building algorithm, that is, SOFM topology building algorithm. Though it may seem that the training and retraining as required by SOMTB algorithm would require energy but it can be utilized to generate alternate routing paths very easily with low energy expense

TABLE 1: The node ID's (up to 15 values are shown) corresponding to various values of p (test bed for 30 neurons).

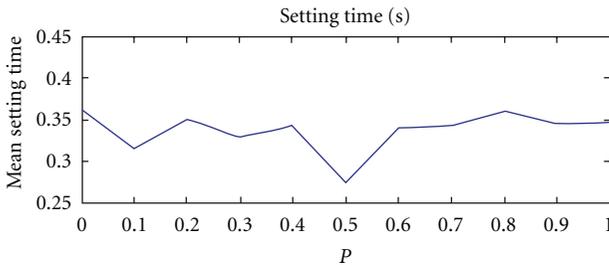
$p = 0$	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$	$p = 0.6$	$p = 0.7$	$p = 0.8$	$p = 0.9$
11	3	19	37	35	26	30	2	35	2
19	1	18	34	7	16	50	30	7	30
18	21	43	25	2	2	7	47	50	50
44	7	44	13	30	30	1	7	30	47
43	35	11	46	50	50	4	4	2	7
5	38	31	45	25	47	35	35	50	4
49	40	45	49	38	7	29	29	29	35
48	29	27	27	40	35	40	40	40	29
14	50	10	22	32	4	3	32	17	40
26	29	15	31	3	1	17	3	32	32
16	20	37	11	25	17	25	38	3	3
2	2	46	42	28	32	28	25	25	38
30	30	13	19	24	3	24	24	28	25
50	16	6	18	46	40	6	6	24	24
7	2	24	44	13	38	33	13	34	6



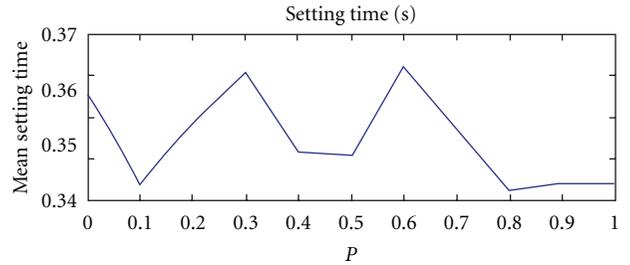
(a)



(a)



(b)



(b)

FIGURE 7: The performance graph with 40 neurons in the test condition.

FIGURE 8: The performance graph without SOFM topology building algorithm in the test condition.

which is not suitable for small pilot areas but effective for larger geographical areas.

During the simulation process the SOMTB algorithm which forms the backbone of spantree protocol here will have the sender node forwarding the routing table to its immediate neighbor. This immediate neighbor will update its own table discarding the redundant information, thus each table will contain only the information about the immediate neighbors. This partial information in larger sense will be accumulated and delineate the complete network picture. So the routing table will remain manageable in terms of size and computation time.

Important advantages in addition to low power consumption include simplicity, inherent robustness to node or link failure, changing network geometry (in case of battery depletion), reduced redundant packet transmissions and implicit network reconfiguration. Only disadvantage is the need for sufficient Sensor density to maintain network operation. Simulations conform well and illustrate the promise of SOFMTB algorithm for applications such as event detection and monitoring over a large distributed area.

This algorithm can be extended for the use with other protocols used for WSN.

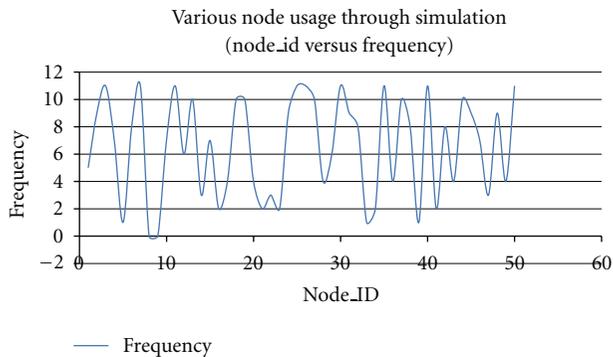


FIGURE 9: Various Node Usages through simulation (node.id versus frequency).

Showing all the promise of efficiency and low power consumption the implementation of synchronization has to be done as future work.

References

- [1] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122–1133, 2001.
- [2] C. K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 138–147, 2001.
- [3] C. F. Chou, J. J. Su, and C. Y. Chen, "Straight Line Routing for wireless sensor networks," in *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC '05)*, pp. 110–115, IEEE, June 2005.
- [4] S. J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in Ad hoc networks," in *Proceedings of the International Conference on Communications (ICC '01)*, pp. 3201–3205, 2001.
- [5] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proceedings of the 19th International Conference on Network Protocols (ICNP '01)*, pp. 14–23, November 2001.
- [6] L. He, "Energy-efficient multi-path routing with short latency and low overhead for wireless sensor networks," in *Proceedings of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, vol. 3, pp. 161–167, 2007.
- [7] A. P. Subramanian, A. J. Anto, J. Vasudevan, and P. Narayanasamy, "Multipath power sensitive routing protocol for mobile ad hoc networks," in *Proceedings of the Conference Wireless on Demand Network Systems*, vol. 2928 of *Lecture Notes in Computer Science (LNCS)*, pp. 171–183, 2004.
- [8] YU. Wang, W. Wang, and X. Y. Li, "Efficient distributed low-cost backbone formation for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 7, pp. 681–693, 2006.
- [9] S. C. Zhang, P. K. Wong, D. G. Sachs, R. Koetter, and D. L. Jones, "Stability analysis of stochastic sensor networks," in *Proceedings of the Asilomar Conference*, 2004.
- [10] L. Cai, H. Huang, S. Blackshaw, J. S. Liu, C. Cepko, and W. H. Wong, "Clustering analysis of SAGE data using a Poisson approach," *Genome Biology*, vol. 5, no. 7, p. R51, 2004.
- [11] S. Haykin, *Neural Networks A comprehensive Foundation*, Pearson Education, New Delhi, India, 2007.
- [12] Crossbow, "MICA, Wireless Measurement System Datasheet," 2009, http://stomach.v2.nl/docs/Hardware/DataSheets/Sensors/MICA_data_sheet.pdf.
- [13] ANSI/IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ANSI/IEEE std 802.11, 1999.
- [14] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, "Simulation based optimization of communication protocols for large-scale wireless sensor networks," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Mont, USA, March, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

