International Journal on Advances in Intelligent Systems, vol 3 no 1 & 2, year 2010, http://www.iariajournals.org/intelligent_systems/

Using Convolution Filters for Energy Efficient Routing Algorithm in Sensor Networks

Yaser Yousef
University of Haute-Alsace
34 rue du Grillenbreit
F-68000 COLMAR, France
yaser.yousef@uha.fr

Marc Gilg
University of Haute-Alsace
34 rue du Grillenbreit
F-68000 COLMAR, France
marc.gilg@uha.fr

Pascal Lorenz
University of Haute-Alsace
34 rue du Grillenbreit
F-68000 COLMAR, France
pascal.lorenz@uha.fr

Abstract

Several protocols and algorithms are used for energy efficient routing in Wireless Sensor Networks (WSNs). These protocols enable not only to reduce communication latency in these networks but also to maximize the network lifetime that is directly related to energy sensors due to the constraint of the batteries. Some of these protocols use clustering or avoiding holes to reach this goal. We have proposed, in this paper, a new hybrid algorithm based on clustering and convolution filters for efficient routing in terms of energy. For this reason, we present the network as a grayscale image. Our algorithm happens in two steps, each of these steps uses a different convolution filter. One kernel filter is used to determine the nodes that participate in the routing by using the mean filter. Another kernel is to pass packets from a source to the destination by using the gradient to select the routes over the most efficient battery.

Keywords - wireless sensor network, energy efficient routing, clustering, matrix convolution, kernel filter.

1. Introduction

During the last decade, wireless communications have emerged enabling users an access to information and electronic services regardless of their geographical positions. New applications of these technologies appear constantly. They have attracted great interest among individuals, businesses and industry.

One of these wireless communications is the Wireless sensor networks which are considered as a specific type of ad hoc network [1]. Indeed, the WSN share with MANET (Mobile Ad hoc Networks) several properties, such as the lack of infrastructure and wireless communications. But one of the key differences between the two architectures is the scope.

Wireless sensor networks are composed by a set of embedded processing units, called nodes, communicating via wireless links.

The goal of a WSN is to collect a set of environmental parameters surrounding nodes, such as temperature or pressure of the atmosphere in order to transport them to the point of treatment.

Unfortunately, the WSN are not a perfect solution, because of their low cost and their deployment in hostile areas, the sensors are quite fragile and vulnerable to various forms of failure: breakdown, non-rechargeable batteries... etc.

As sensor nodes are equipped with limited battery energy, it is also necessary to minimize their energy efficiency. The factor "energy" is at the center of the research around the sensor networks. This energy is mainly due to the air interface.

A sensor, often has limited energy and battery replacement is impossible. This means that the lifetime of a sensor depends greatly on the life of the battery. In a sensor network (multi-jumps) each node collects data and sends / transmits values. The failure of some nodes requires a change of network topology and the re-routing of packets. All these operations consume energy and therefore the current research focuses primarily on ways to reduce consumption.

The routing protocols for WSN have been widely studied, and various studies have been published. These protocols should be able to choose paths to route packets based on the actual condition of the network by increasing its lifetime and reducing the convergence time.

Clustering is one of the major approaches to treat the structure of a wireless sensor networks. It allows the formation of a virtual backbone that improves the use of limited resources such as bandwidth and energy. Furthermore, clustering helps to achieve multiplexing between different clusters. In addition, it improves the performance of routing algorithms. Several protocols use this preventive approach.

In a context of mobility and economy of node energy resources, self-configuring network is a problem in sensor networks. To preserve the battery of a node, it is necessary that its radio transmitters switch off as often as possible transition. Then, the problem of the synchronization of the nodes and the distribution waking periods appears. MAC layer is a solution that allows nodes to have sleep phases, without affecting communication. Several solutions at the MAC layer have been proposed to save energy. The role of MAC protocols is to organize the medium access between nodes wishing to communicate and to allow proper coordination between the nodes to share access to the medium but also to minimize node energy dissipation.

In this paper, we present an algorithm to improve and to complete our previous work in [2] and [3]. In this algorithm, we present the wireless sensor network as a grayscale image to achieve energy efficient routing. The pixels in this image represent the nodes in the WSN. We cut the image into sub-units or areas around a central node to determine the energy distribution in an area around this node. White regions in this image represent regions of the network that have sensors with a full battery. Unlike black regions which have nodes with empty batteries. We have used matrices and products of convolution in this algorithm using two filters. These two filters are used in the techniques of image processing; the first (the mean filter) is effective in identifying the nodes that will participate in the routing system. The second is to perform the routing of packets through the routes which have more energy efficiency (through the nodes that have the greatest capacity).

In section 2, we present a reminder of our previous work. Energy sector is explained in section 3. The Sections 4 and 5 describe our proposal. Results are presented in section 5. These are tested by the simulator OMNET++.

2. RELATED WORK

Several protocols have been proposed to conserve energy in WSNs. Scheduling protocols are one of these protocols, they are efficient in energy use because they avoid collisions and overhearing. They do not allow peer-to-peer communication and generally require cluster nodes. The inter-cluster communication is achieved by the two approaches: TDMA and FDMA [4].

Others, such as SMACS (Self-organizing Medium Access Control for Sensor Networks) proposed by Sohrabi and Pottie [5], which is a MAC protocol combining TDMA and FDMA in which the neighboring nodes randomly choose a slot and a frequency defining link.

Several solutions at the MAC layer have been proposed [4, 5, 6]. This layer allows nodes to have sleeping phases. For the MAC layer, two topologies are supported by the 802.15.4 protocol, the star topology and the peer to peer topology. In the star topology, communications are established directly between the central node (coordinator) and the sensors. The coordinator is the node that initiates and manages the network communications. The topology peer to peer allows each network node to communicate with any other node.

S-MAC (Sensor MAC) [7] is a similar protocol than 802.11. It uses the medium access CSMA / CA RTS / CTS (Request-To-Send, Clear-To-Send) that avoids collisions and hidden node problem [8]. The protocol establishes a mechanism for a standby distribution to each node in order to reduce energy consumption and extend its life. Each node should coordinate and exchange information with its neighbors to choose its own "sleep / active" cycle.

Other types of energy efficient routing protocol in WSNs have been created as the hierarchical protocols with clustering, cluster-head, aggregation and data fusion to reduce the total energy of the network by limiting communications between nodes. Consequently, the routing is done between clusters and not between nodes [9, 10, 11].

In [12, 13], protocols have been deployed to maximize the lifetime of the network by avoiding regions with less battery to choose the route that passes through the regions with high battery capacity.

As mentioned above, this paper is an improvement of our previous work in [1] and [2].

In [1] and [2], we used an image processing algorithm to get information about energy repartition in the network. To do this, we represented the network energy capacity by a grayscale image. The pixel with coordinates X, Y represents the sensors located at X, Y and the gray level is defined by the energy capacity C of this sensor. Any sensor with a full battery is represented by a white pixel, and the sensor with an empty battery is referred to by a black pixel. In this representation, the adaptive algorithm routes packets to the light part of the image to preserve the battery capacity.

An adaptive routing algorithm based on image processing algorithm is represented in [1]. It utilized the gradient to select an energy efficient path. In this algorithm, we obtained the energy gradient in an image by the convolution product computed by the Sobel algorithm which will be detailed in Section 4.

In [2], we used the convolution matrix and clustering to choose the more economical energy path to minimize energy consumption in the entire network.

3. NETWORK MODEL AND POWER MODEL

Before presenting our algorithm, we must mention that we work in a sensor network that has a limited capacity C. The sensors are often in isolated locations, and in some application domains, it is almost impossible to replace their batteries. So, we must conserve their capacities because the failure of a sensor could cutoff communications in the entire network. This capacity must be used effectively to extend the network lifetime. These sensors communicate with each other through wireless links in which packets go in one direction or in an alternative direction but not simultaneously. This type of connection provides a bidirectional link using the full capacity of the network. Each node is capable of receiving and transmitting packets to a particular node chosen by the administrator called CH (Cluster-Head) who will coordinate the nodes. These packets are transmitted as UDP messages.

Energy is consumed primarily by reception and transmission operations [2, 3], which is used the following energy model:

$$E(t+\Delta t) = E(t) - E_{tran} * \Delta t_{tran} - E_{recv} * \Delta t_{recv} - E_{idle} * \Delta t_{idle} - E_{sleep} * \Delta t_{sleep} \quad (2)$$

$$\Delta t = \Delta t_{tran} + \Delta t_{recv} + \Delta t_{Eidle} + \Delta t_{sleep} \quad (3)$$

E_{tran}, E_{recv} denote energy consumption for transmission and reception.

E_{idle}, idle energy, refers to the energy spent in idle mode.

E_{sleep} denotes energy expenditure in sleep mode.

Δt_{tran} , Δt_{recv} denote the time of transmission, reception energy.

Δt_{Eidle} denotes the time of energy spent in idle mode.

Δt_{sleep} denotes the time of energy spent in sleep mode.

4. Energy sectors and associated matrix

After representing a wireless sensor network as a grayscale image, we can cut the neighborhood of a node into sectors in order to compare the energy of this node relative to its neighbors. We cut the region around a pixel (sensor) in eight or in twenty-four sectors as shown in Figures 1.

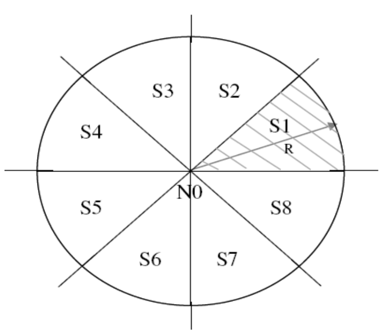

Figure 1. Energy sector

The goal of this representation is to know the distribution of energy in the network [14, 15]. We can see that the network as an image with white and black areas. Consequently, we represent the wireless sensor network like a map of pixels. Each pixel is identified by both an x and a y coordinate as shown in Figure2. In the context of digital images, we can use a grid of rows and columns in which is reserved a place to store the value of each pixel of the image. In mathematics this type of chart is called a matrix.

$P_{x-1,y-1}$ b	$P_{x,y-1}$ b	$P_{x+1,y-1}$ b
$P_{x-1,y}$ b	$P_{x,y}$ a	$P_{x+1,y}$ b
$P_{x-1,y+1}$ b	$P_{x,y+1}$ b	$P_{x+1,y+1}$ b

Figure 2. Pixel's image

In our case, the value related to the brightness of a pixel represents the energy capacity of a node. Hence, each sector will have the mean energy used to create the energy matrix M. This matrix will be used to calculate the product convolution [16].

$$M = \begin{pmatrix} s_3 & s_2 & s_1 \\ s_4 & s_0 & s_8 \\ s_5 & s_6 & s_7 \end{pmatrix}$$

5. Matrix convolution

A convolution is a mathematical operation which is used to multiply matrices together. In our case, we multiply two matrices:

The first is the image matrix that represents the energy values of nodes around a central node (energy matrix).

The second is a matrix called the kernel, noted as K. This matrix is the "heart" of all the changes that will affect the image. The kernel will act on each pixel, i.e. on each element of the matrix image [17].

The kernel is composed by the square matrix [k] of 3x3 elements. We apply a convolution filter to multiply each pixel of the image matrix M by the kernel [k]. To calculate the value of a pixel (x, y) of the image matrix, we multiply its value by the pixel central kernel K (2.2) shown in the Figure 3 and summing up the product value of adjacent pixels.

K (1,1)	K (2,1)	K (3,1)
K (2,1)	K (2,2)	K (2,3)
K (3,1)	K (3,2)	K (3,3)

Figure 3. Kernel matrix

Then the convolution product of two n+1×p+1 matrix K and M is defined by the formula:

$$K * M = \sum_{i=0}^{n} \sum_{j=0}^{p} K_{i,j} \times M_{n-i,p-j} \qquad (1)$$

To perform the convolution product of two functions we take an average of these two functions. It is widely used in mathematics to approximate and stabilize functions.

6. KERNEL MATRIX

This work will proceed in two phases and each phase uses a different filter to find the most economical energy path from the source to the destination. The two kernels used are:

6.1 1D : 1 kernel matrix K (mean filter)

As shown in Figure1, we have eight energy sectors. In our proposal, we will use the following kernel matrix:

$$K = \begin{pmatrix} 1/12 & 1/12 & 1/12 \\ 1/12 & 4/12 & 1/12 \\ 1/12 & 1/12 & 1/12 \end{pmatrix}$$

In this way, we will use the mean filter that is simple, intuitive and easy to implement for smoothing images. This filter represents the shape and size of the neighborhood to be sampled when calculating the mean. It replaces each pixel by the average values of adjacent pixels and the central pixel. Therefore, it smoothes the signal and the filter works as a low-pass one. This filter enables any sensor of the network to take an average across its neighborhood. In this configuration, the central node will contribute for half weight compared to the eight surrounding regions. Doing this, a high energy capacity central node will be chosen to forward the packets, even if it is located in a poor energy region.

6.2 2D: 2 kernel matrices Kx,Ky (Sobel)

In our routing, after selecting the active nodes, we have to select the most economical energy routes. In this step we use the following matrices:

$$K_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

These matrices are used by the algorithm of Sobel [18] to obtain the gradient in grayscale image. The Sobel filter uses two 3x3 kernels, one for the horizontal axis (X) and the other for the vertical (Y). Each core is a gradient filter, which is them combined to create the final image.

The energy gradient $G_{i.i} = (G_x,G_y)$ at sensor i, j used by Sobel, it is obtained by the following equations:

$$G_x = M_{i,j} * K_x \qquad (4)$$

$$G_y = M_{i,j} * K_y \qquad (5)$$

Where $M_{i.i}$ is a 3 × 3 sub matrix of the grayscale image that represents the sensors energy capacities.

In the Sobel algorithm $G_{i,j}$ represents the gradient of gray intensity at pixel i, j. This intensity is related to the battery capacity of sensor i, j by definition.

The norm of the gradient is given by:

$$\|G\| = \sqrt{G_x^2 + G_y^2} \qquad (6)$$

and the direction of the gradient is given by:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \qquad (7)$$

The product of multiplying matrix M by matrices K_x, K_y on the two axes x and y is then:

- On the x-axis shown in Figure.4:

$G_x = M_{i,j} * K_x$

$$= \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * \begin{pmatrix} s_3 & s_2 & s_1 \\ s_4 & s_0 & s_8 \\ s_5 & s_6 & s_7 \end{pmatrix}$$

$$= S_7+2S_8+S_1-S_3-2S_4-S_5 \qquad (8)$$

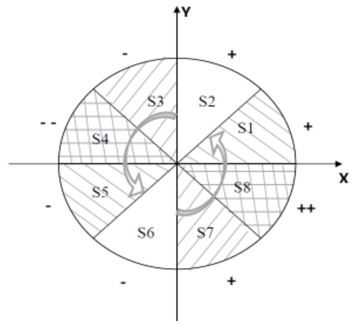

Figure 4. Gradient direction x

We can notice that the direction of the gradient on the x-axis is influenced by six values including three positive values (S_7, $2S_8$, S_1) and three negative ($-S_3$, $-2S_4$, $-S_5$). The central energy sector on each side (S_8 and S_4) is the double of the energy of its neighboring areas.

⬩ On the y-axis shown in Figure.5:

$G_y = M_{i,j} * K_y$

$$= \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * \begin{pmatrix} s_3 & s_2 & s_1 \\ s_4 & s_0 & s_8 \\ s_5 & s_6 & s_7 \end{pmatrix}$$

$$= S_5+2S_6+S_7-S_1-2S_2-S_3 \qquad (9)$$

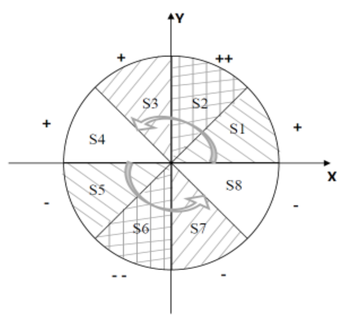

Figure 5. Gradient direction y

We can notice that the direction of the gradient on the y-axis is influenced by six values including three positive values (S_5, $2S_6$, S_7) and three negative ($-S_1$, $-2S_2$, $-S_3$). And the central energy sector to each side (S6 and S2) is the double of the energy of its surrounding areas.

These two kernels will be used in the algorithm that we will explain in the next section.

7. Hybrid Algorithm

The failure of a sensor can have serious consequences on the total life of the network, so it is necessary to involve only the nodes that have a capacity greater than a certain threshold (called ET). The sensors that do not participate in routing will be dormant for the period of pending instructions from HC to wake up and participate in routing.

7.1 Parameters of our algorithm

- X_i, Y_i: positions of sensors N_i.
- CH: Cluster-Head.
- BC: Battery capacities.
- ET: Energy Threshold, if sensor's battery is less than this value, the forwarding packets over UDP will not happen.
- SEID: Sending Energy Information Delay, the delay between two information packets.
- T: Time of energy packet sent (current time – last time).
- $T_{sleep:}$ sleep time.
- R: Range, for energy sectors.
- K: Kernel matrix.
- M: Energy matrix of N_i.
- ERP: Energy Routing Parameter: K*M, the average energy around the node.

This work will proceed in three phases in order to find the most economical energy path from source to destination.

7.2 Energy Routing Algorithm steps

7.2.1 Exchange energy information and convolution computations

1. Sending energy information:
   If (BC_i change) then
   If (T > SEID) and node is sleeping
   Then send (BC_i and X_i, Y_i) over UDP to CH.
2. Convolution computations (running on CH):
   a) Obtain energy information from Ni.
   b) update energy table :
      - IP address of N_i
      - X_i, Y_i
      - BC_i
   c) Compute:
      - Compute energy matrix: M of N_i
      - $ERP_i = K*Ms_i$
      - G_{xi} , G_{yi}
   d) Send ERP to all sensors over UDP.

7.2.2 Waking state and sleeping state

3. On each sensor: routing packet and IP routing process:
   For each sensor, get ERP, G_{xi} ,G_{yi} over UDP.
   If (ERP < ET)
   Then sleep during t_{sleep}
   And if (sent packet is not a control packet)

Then sensor is set in standby mode

4. Sensor wake-up during time t_{sleep} then it request a new K*M value
   If (k*M> ET) then end standby mode

7.2.3 Routing packets

5. Compute vector $V=(K_x*M-S_x,K_y*M-S_y)$ at sensor (S_x,S_y)
6. Compute cos α=cos (V,G) where G is next-hop vector and

$$COS\left(V_{x',y'},G_{x,y}\right)=\frac{\vec{V}_{x',y'}\times\vec{G}_{x,y}}{\left\|\vec{V}_{x',y'}\right\|\times\left\|\vec{G}_{x,y}\right\|} \quad (10)$$

7. Compute the minimums values of cos α.
8. Select the path with the maximal of the minimums values of cos α.

The two following charts represent our algorithm:

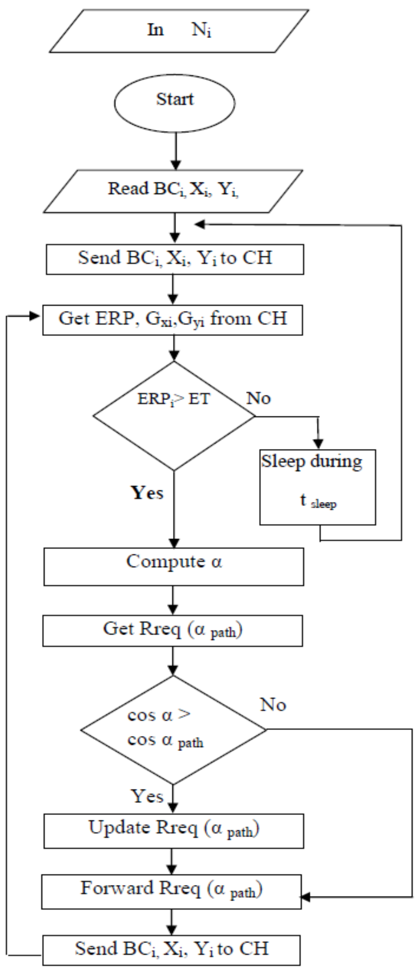

Figure 6. Running on N_i

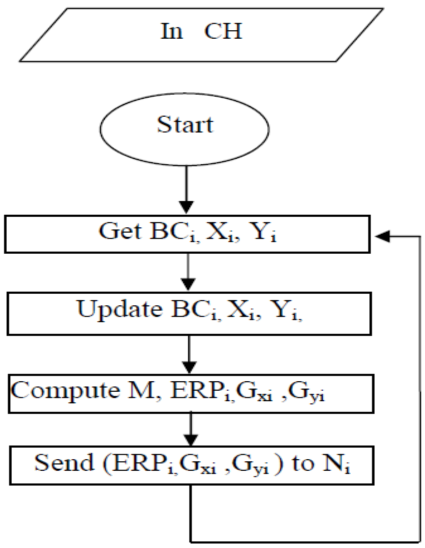

Figure 7. Running on CH

Therefore, we can resume the previous three phases of this algorithm as:

1D is used to obtain the energy information of each node by the C-Hs.

2D is used to obtain the direction of the gradient indicating the searched route through which passes the packets.

8. SIMULATION

Our work is done in a sensor network which has the topology shown in figure 8 where N1 is the source, N3, N4 are the gateways and N0 is the destination.

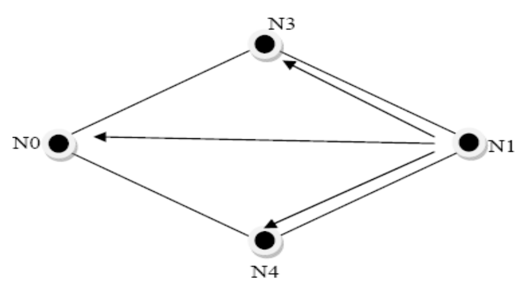

Figure 8. Network topology

Results shown in the tables 1, 2, 3, 4, 5 and 6 represent the average of twenty simulation runs performed by the simulator OMNET++ that had been modified by implementing the two kernels (Sobel and mean filter) [18, 19]. We have used the device Nano WiReach as a model that acts as a bridge to connect serial devices to 802.11b/g Wireless LANs [20]. Its power Consumption: 250mA in transmission, 190 mA in reception (typical), 8 mA in sleep and 8 mA for idle.

We have used the AODV routing protocol. Control packets that have a size of 512 Bytes are sent using UDP protocol. Node N1 sends packets as a burst of a frequency of 0.01s with a sleep delay of: 0, 50, 100 and 150s. Initial values of energy given to N0, N1, N3, and N4 in our simulations are respectively: 40mA, 20mA, 2mA, and 10mA, where N0 is the CH in our configuration. The energy threshold has the values: 0mA, 1mA, 2mA, 3mA, 4mA and 5 mA. Total simulation time is 1000 seconds.

The following three tables 1, 2 and 3 indicate the simulation results for the number of sent packets.

Sleep delay	Threshold (0)				Threshold (1)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$2.41e^{+4}$	$1.06e^{+5}$	$3.32e^{+4}$	$9.72e^{+4}$	$2.45e^{+4}$	$1.06e^{+5}$	$2.54e^{+4}$	$9.33e^{+4}$
50	$1.92e^{+4}$	$7.09e^{+4}$	$3.63e^{+4}$	$6.21e^{+4}$	$1.52e^{+4}$	$6.66e^{+4}$	$1.99e^{+4}$	$5.81e^{+4}$
100	$1.93e^{+4}$	$7.95e^{+4}$	$3.88e^{+4}$	$6.00e^{+4}$	$1.88e^{+4}$	$8.53e^{+4}$	$1.72e^{+4}$	$8.48e^{+4}$
150	$1.91e^{+4}$	$7.87e^{+4}$	$3.63e^{+4}$	$6.17e^{+4}$	$2.13e^{+4}$	$1.06e^{+5}$	$1.80e^{+4}$	$1.07e^{+5}$

Table 1: Packet sent number with ET of 0 and 1

Sleep delay	Threshold (2)				Threshold (3)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$2.12e^{+4}$	$8.50e^{+4}$	$1.22e^{+4}$	$6.02e^{+4}$	$1.82e^{+4}$	$6.75e^{+4}$	$1.25e^{+4}$	$2.36e^{+4}$
50	$2.13e^{+4}$	$1.06e^{+5}$	$2.82e^{+3}$	$1.19e^{+5}$	$7.21e^{+3}$	$5.11e^{+4}$	$1.25e^{+3}$	$3.74e^{+4}$
100	$2.08e^{+4}$	$1.06e^{+5}$	$1.88e^{+3}$	$1.21e^{+5}$	$6.85e^{+3}$	$4.53e^{+4}$	603.000	$4.01e^{+4}$
150	$1.88e^{+4}$	$9.85e^{+4}$	$1.03e^{+3}$	$1.13e^{+5}$	$6.65e^{+3}$	$4.25e^{+4}$	366.000	$4.00e^{+4}$

Table 2: Packet sent number with ET of 2 and 3

Sleep delay	Threshold (4)				Threshold (5)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$2.21e^{+4}$	$8.72e^{+4}$	$1.35e^{+4}$	$1.68e^{+4}$	$2.56e^{+4}$	$1.06e^{+5}$	$1.41e^{+4}$	$2.13e^{+4}$
50	325.000	$2.49e^{+4}$	306.000	41.000	321.000	$2.49e^{+4}$	305.000	42.000
100	110.000	$1.29e^{+4}$	23.000	98.000	110.000	$1.30e^{+4}$	24.000	97.000
150	92.500	$1.03e^{+4}$	83.000	19.000	92.500	$1.03e^{+4}$	82.000	18.000

Table 3: Packet sent number with ET of 4 and 5

In the three tables above, we see that there are a few packets sent by the node N0 as it is the destination. In Table 3, at a threshold of 4 and 5, we note that the number of packets that take the route between the source and the destination by both gateways (N3 and N4) tends to zero when the threshold is greater or equal to 4. In other words, the communication will be interrupted. We will focus on routing with a threshold below 4. At threshold between 2 and 3, energy has been saved because the number of packets sent by N4 decreases as shown in figure 9 and 10.

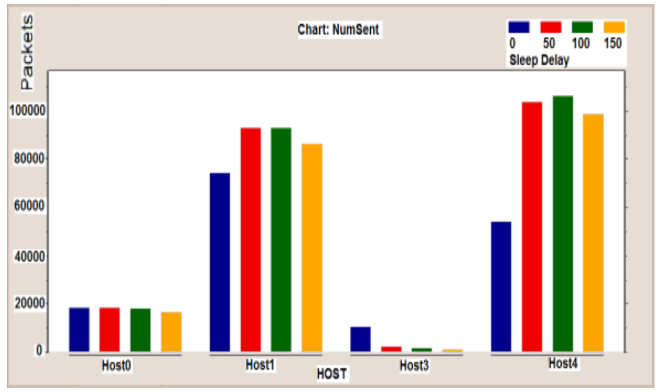

Figure 9. Packet sent number with ET of 2

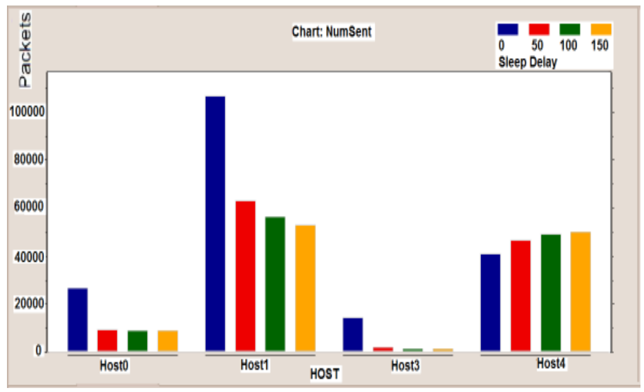

Figure 10. Packet sent number with ET of 3

The following tables represent the percentage of remaining energy in the network nodes; we note that the energy value remaining in N3 is zero because its initial value is proximate to the threshold and total simulation time is 1000s. Consequently, we will concentrate on the other gateway (node N4).

Sleep delay	Threshold (0)				Threshold (1)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	0.909	0.826	0	0.603	0.911	0.830	0	0.609
50	0.870	0.694	0	0.307	0.884	0.666	0	0.301
100	0.867	0.736	0	0.358	0.868	0.799	0	0.443
150	0.866	0.734	0	0.353	0.912	0.830	0	0.609

Table 4: Energy Left with ET of 0 and 1

Sleep delay	Threshold (2)				Threshold (3)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	0.879	0.761	0	0.426	0.839	0.771	0	0.342
50	0.912	0.831	0	0.608	0.933	0.865	0	0.724
100	0.912	0.831	0	0.608	0.936	0.867	0	0.722
150	0.907	0.800	0	0.550	0.936	0.868	0	0.723

Table 5: Energy Left with ET of 2 and 3

Sleep delay	Threshold (4)				Threshold (5)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	0.890	0.771	0	0.503	0.922	0.839	0	0.698
50	0.944	0.881	0	0.779	0.944	0.811	0	0.779
100	0.948	0.885	0	0.778	0.948	0.885	0	0.778
150	0.949	0.886	0	0.778	0.949	0.886	0	0.778

Table 6: Energy Left with ET of 4 and 5

In tables 4 and 5, we see that with a T $_{sleep}$ equal to 0 and a threshold between 0 and 3, the node N4 loses its energy. On the other hand, it retains its energy if the threshold is greater or equal to 4 as shown in tables 3 and 6, because it does not send packets (except control packets).

At a T $_{sleep}$ of 50s and at a threshold of 1, node N4 has used 70% of its energy. At a threshold of 2 it has only used 40% and at a threshold of 3 or 4 or 5 it has used 24% of its capacity. That means, we have saved 30% of energy in combination (T $_{sleep}$ = 50, ET = 2) as shown in Figure 11 and 12.

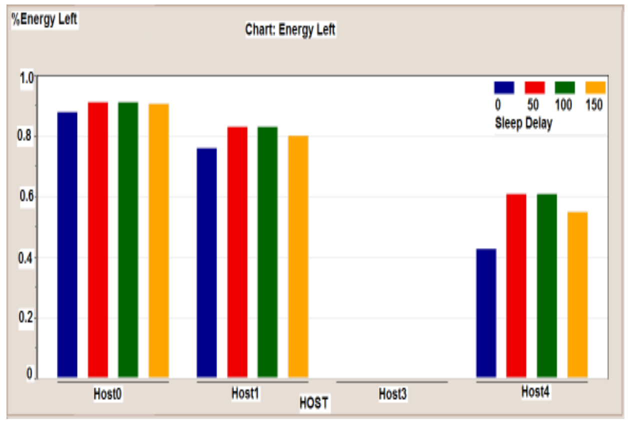

Figure 11. Energy left with ET of 2

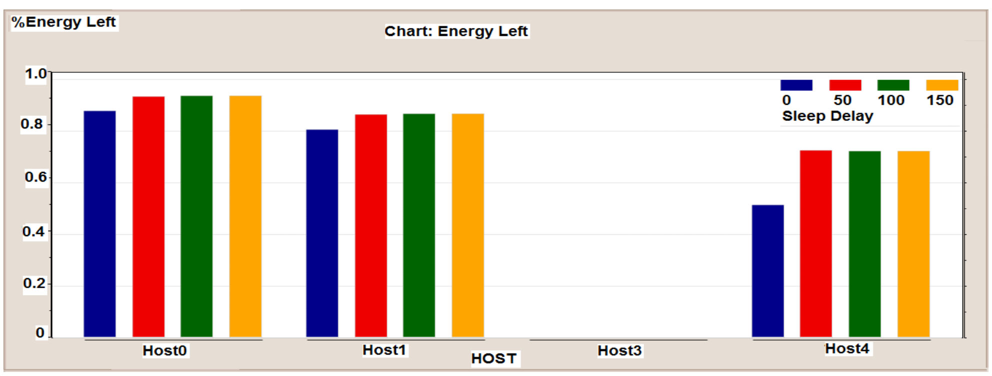

Figure 12. Energy left with ET of 3

At a T $_{sleep}$ of 100s and at a threshold of 1, node N4 has used 55% of its energy. At a threshold of 2 it has only used 40% and at a threshold of 3 or 4 or 5 it has used 24% of its capacity. That means, we have saved 15% of energy in combination (T $_{sleep}$ = 100, ET = 2).

If we increase the T $_{sleep}$ to 150s. At a threshold of 0, node N4 has used 64% of its energy. At a threshold of 2 it has only used 40% and at a threshold of 3 or 4 or 5 it has used 24% of its capacity. But the number of packets forwarded by N4 decreases dramatically. If we continue to increase the threshold, the node stops to relay packets as shown in figure 13.

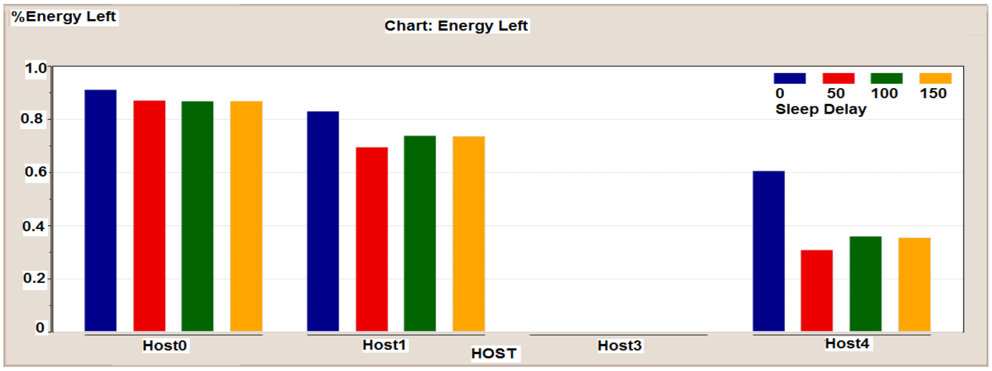

Figure 13. Energy left with ET of 0

Concerning energy consumption over time, it is clear that if we increase the sleep time, we save more energy as shown in Figures 14 and 15, which show the energy change of node N3 during the simulation time of 1000s.

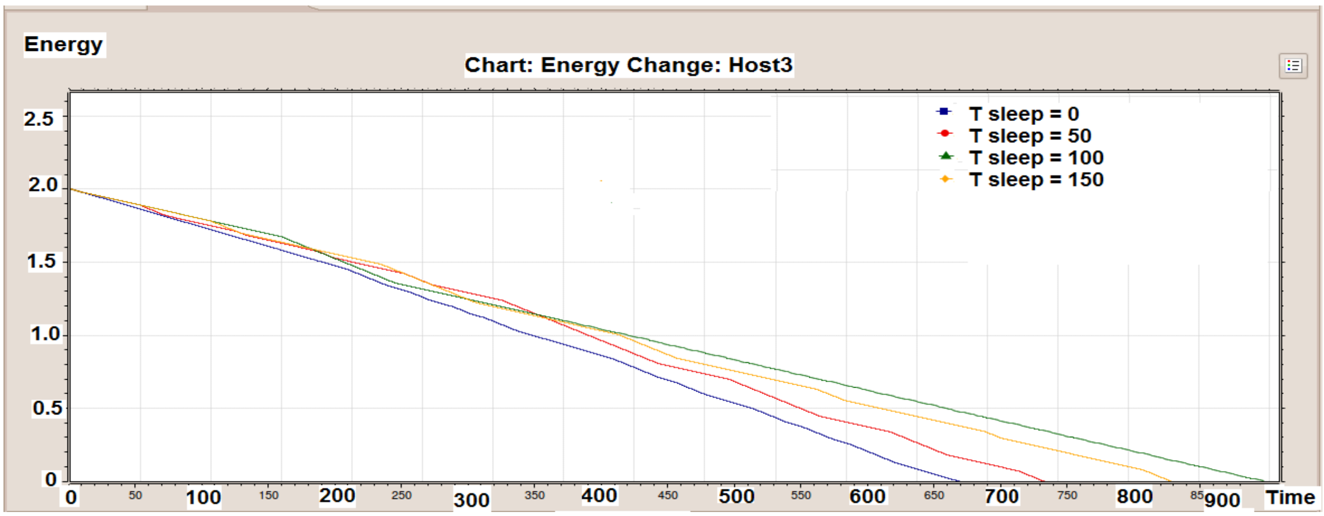

Figure 14. Energy Consumption of N3 with ET of 2

At a threshold of 2mA and a simulation time of 1000s, we remark that the node N3 stops transmitting and it is exhausted;

BC_i= 0 after 672 seconds when T_{sleep} = 0
BC_i= 0 after 739 seconds when T_{sleep} = 50
BC_i= 0 after 882 seconds when T_{sleep} = 100
BC_i= 0 after 823 seconds when T_{sleep} = 150

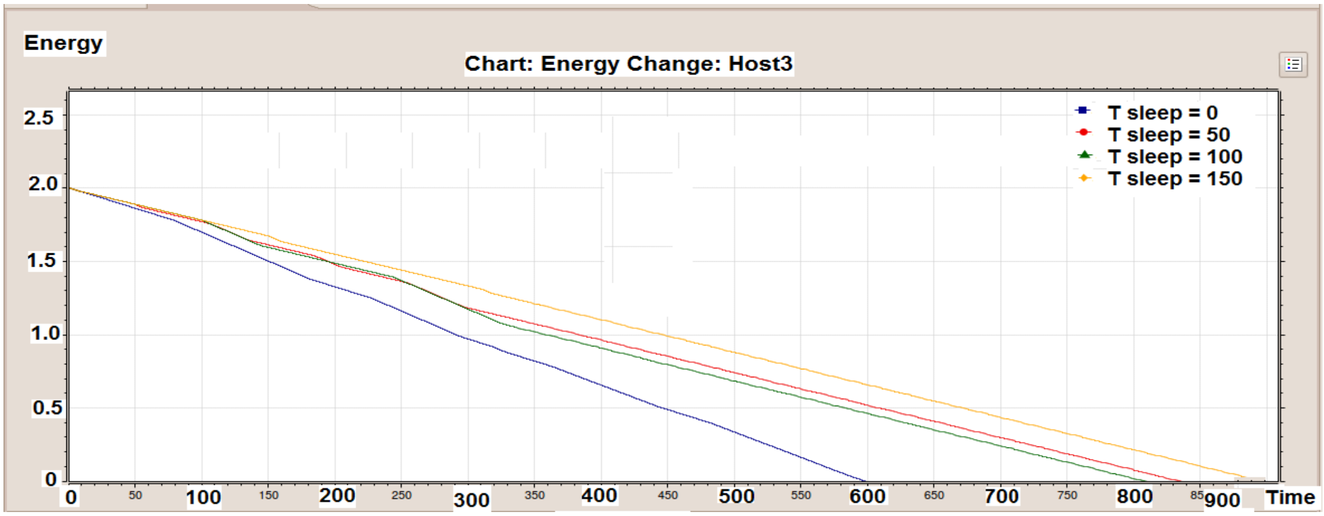

Figure 15. Energy Consumption of N3 with ET of 3

On the other side, at a threshold of 3mA;

BC_i= 0 after 598 seconds when T_{sleep} = 0
BC_i= 0 after 836 seconds when T_{sleep} = 50
BC_i= 0 after 810 seconds when T_{sleep} = 100
BC_i= 0 after 897 seconds when T_{sleep} = 150

Looking at the slope of the curve in Figures 14 and 15, we note that, the greater the sleep delay, the more the slope of curve is low (the less the node consumes its energy).

Our main objective is to extend lifetime. In the following graphs, which show the energy change of node N4 during the simulation time of 1000 seconds, we note that if we increase the sleep time to 150 and if we increase the energy threshold from 0 to 3, we achieve the best results as is shown in the following figures:

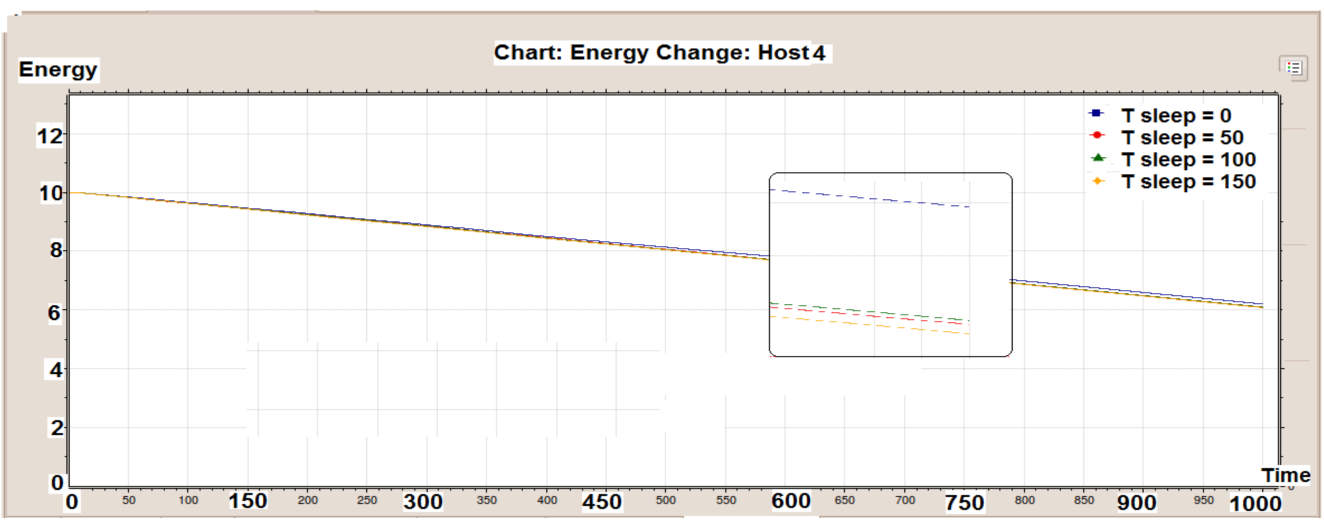

Figure 16. Energy Consumption of N4 with ET of 2

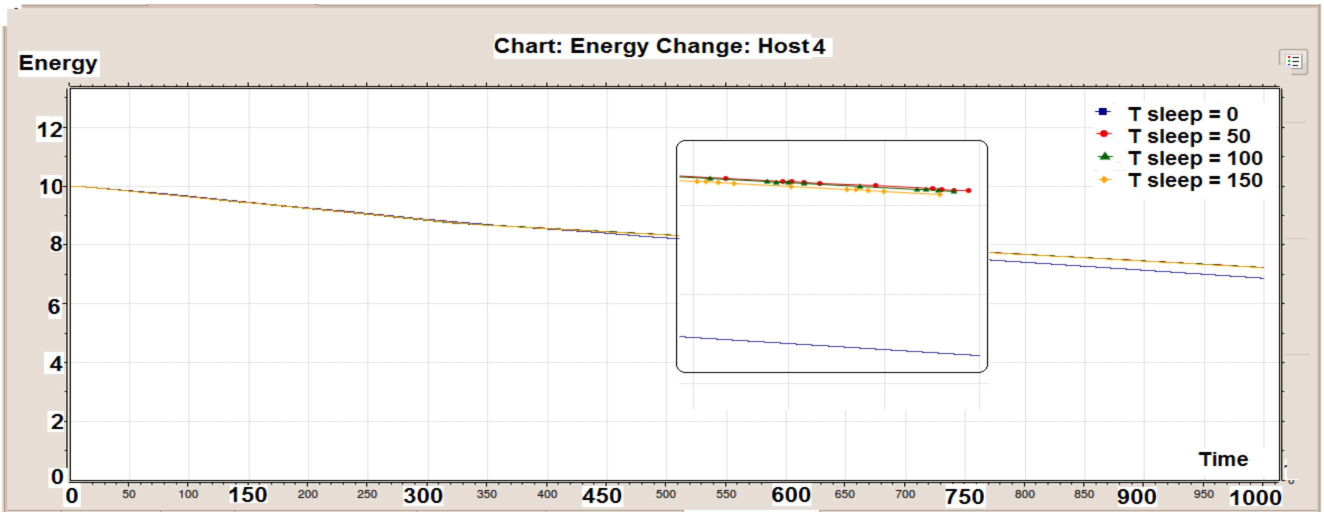

Figure 17. Energy Consumption of N4 with ET of 3

Consequently, we found that the best results are obtained with a T_{sleep} of 50 and a threshold between 1 and 2, because we have saved 30% of energy.

Moreover, these results show the good performance of our algorithm because we have extended the lifetime of the node N4.

9. Conclusion and future work

In this paper, a new idea was created by treating the wireless sensor network as an image. This image consists of regions containing nodes. Our algorithm cuts the region around the central node into eight sectors. Each sector will have a value of energy used to create an energy matrix (3 × 3). We can use this matrix in the convolution. This parameter is used in the proposed routing algorithm to choose the path that passes through the nodes of a high capacity battery.

This algorithm is based on energy efficient routing. It also uses clustering in which all calculations are done by the CH which has CPU power, memory resources and energy capacity more important than other nodes. In addition, nodes that do not participate in routing are instantly dormant in order to conserve energy. They wake up after a fixed period to either participate in routing or remain dormant.

We have developed some components of the network simulator OMNET + + to adapt our algorithm

We have successfully saved the energy of nodes, i.e. we have retained 30% of the energy of N4. Consequently, we have maximized the total network lifetime.

In future work, other techniques of image processing, such as edge detection, to avoid the holes (empty regions where there are no sensors or there's not enough of energy). We can also use other filters to reduce the number of control messages.

References

[1] M. Gilg, A. Makhlouf,and P. Lorenz, “Fairness index in single and double star Network”. International Journal On Advances in sytems and Measurements”. Vol 2 no1 , 2009, pp. 109-108.

[2] M. Gilg, Y. Yousef, and P. Lorenz, “Using image processing algorithms for energy efficient routing algorithm in sensor networks”. ADAPTIVE 2009. Athens/Glyfada, Greece, November 2009. pp. 132-136.

[3] M. Gilg, Y. Yousef, and P. Lorenz, “Using Matrix convolutions and Clustering for Energy Efficient Routing Algorithm in Sensor Networks”. AICT 2010. Barcelona, Spain, may 2009.

[4] R. Lin, Z. Wang , and Y. Sun, “Energy efficient medium access control protocols for wireless sensor networks and its state-of-art ” .in Industrial Electronics, 2004 IEEE International Symposium, May 2004.pp 669 – 674.

[5] K. Sohrabi and G.J. Pottie “Performance of a novel self-organization protocol for wireless ad-hoc sensor networks”. In Proceeding of the 50th IEEE Vehicular Technology Conference (VTC’99), vol.2, Amsterdam, September 1999. pp.1222-1226.

[6] M. Caccamo, L. Y. Zhang, L. Sha, et G. Buttazzo. “An implicit prioritized access protocol for wireless sensor networks“. IEEE Real-Time System Symposium (RTSS’02), décembre 2002.

[7] W. Ye, J. Heidemann, and D. Estrin. “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”. In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM’02), vol.3, New York, NY, USA, June 2002. pp.1567-1576.

[8] A. Rowe, R. Mangharam, and R. Rajkumar “RT-Link, A Global Time- Synchronized Link Protocol for Sensor Networks”. Elsevier Ad hoc Networks, Special Issue on Energy Efficient Design in Wireless Ad Hoc and Sensor Networks, 2007.

[9] N. V. Subramanian, “Survey on Energy-Aware Routing and Routing Protocols for Sensor

Networks". Technical Report, Computer Science, University of North Carolina, Charlotte, 2004.

[10] K. Akkaya, and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks". Journal of Ad Hoc Networks, Vol. 3, No. 3, May 2005. pp. 325-349.

[11] Al-Karaki, J.N. Kamal, A.E. " Routing techniques in wireless sensor networks: a survey. Wireless Communications", IEEE, December. 2004.

[12] D. Kim, J. J. Garcia-Luna-Aceves, K. Obraczka, J. C. Cano, and P. Manzoni, "Routing Mechanisms for Mobile Ad Hoc Networks Based on the Energy Drain Rate". In IEEE Trans. Mob. Comput, 2003. pp. 161-173.

[13] Y. Tian, F. Yu, and Y. Choi, S. Park, E. Lee, M. Jin, and S. H. Kim, "Energy efficient data dissemination protocol for detouring routing holes in wireless sensor networks". ICC 2008. IEEE International Conference on Communications, vol. 31, no. 1, May 2008. pp. 2322-2326.

[14] M. I. Razzak, S. A. Hussain, A .A. Minhas and M. Sher, "Collaborative Image Compression in Wireless Sensor Networks". International journal of computational cognition (http://www.ijcc.us), vol. 8, no. 1, march 2010.

[15] L. Zhang , Y. Lu, L. Chen, D. Dong, "Game Theoretical Algorithm for Coverage Optimization in Wireless Sensor Networks". Proceedings of the World Congress on Engineering 2008 Vol I WCE 2008, London, U.K, July 2 - 4, 2008.

[16] B. Fisher, Y. Chen, "Generalizations of Non-Commutative Neutrix Convolution Products of Functions". Georgian Math, No. 5, 1997.pp. 413-419.

[17] R.G. Lanckriet, N. Cristianini,P. Bartlett,L. El Ghaoui and M. Jordan, "Learning the Kernel Matrix with Semidefinite Programming". JMLR.org, Volume 5,December 2004, pp27 - 72 .

[18] W. Drytkiewicz, S. Sroka, V. Handziski, A. Koepke, and H. Karl, "A mobility framework for OMNeT++," in Proceedings of the 3rd International OMNeT++ Workshop, Budapest, Hungary, January 2003.

[19] J. Glaser, D. Weber, S. A. Madani, and S. Mahlknecht, "Power Aware Simulation Framework for Wireless Sensor Networks and Nodes". In EURASIP Journal on Embedded Systems, Article ID 369178, May 2008.

[20] Nano WiReach™ "Miniature embedded secure WiFi module". In Connect One Ltd. Publication Number 17-3520-11, September 2008.