ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2014.0130

ICTACT JOURNAL ON COMMUNICATION TECHNOLOGY: SPECIAL ISSUE ON ADVANCES IN WIRELESS SENSOR NETWORKS, JUNE 2014, VOLUME: 05, ISSUE: 02

AUTONOMOUS MOVEMENT OF MOBILE RELAYS IN WIRELESS SENSOR NETWORK TO ENHANCE NETWORK LIFETIME

Regis W. Anne[1] and Elijah Blessing Rajsingh[2]
[1] *Department of Computer Applications, Sri Krishna College of Engineering and Technology, India*
E-mail: regisanne@skcet.ac.in
[2] *School of Computer Science and Technology, Karunya University, India*
E-mail: elijahblessing@karunya.edu

Abstract

Wireless Sensor Network (WSN) is a collection of intelligent sensors that communicate and coordinate in an energy constrained environment. Due to the limited energy supply of the sensors, lifetime problems arise in static WSN. One of the potential solutions to this problem is a mobility-assisted WSN. Mobility can be introduced by adding extra entities like mobile sinks, mobile cluster-head or mobile relays (MRs) to the existing static WSN. The entities can significantly improve the functionality and performance of the WSN by making it flexible to failures, ease data collection, increase energy efficiency, enhance connectivity, improve coverage and prolong network lifetime. The need of today's applications demands mobility-assisted WSN instead of the traditional static WSN. In this paper, we control the movement of MRs to maximize network lifetime. A distributed algorithm for controlling the movement of MRs is given and its performance is validated for the network parameters such as network lifetime and average residual energy. The proposed method shows that lifetime can be improved compared to that of a static network.

Keywords:

Mobile Relays, Network Lifetime, Mobility, Simulations

1. INTRODUCTION AND RELATED WORK

Wireless Sensor Network (WSN) is a collection of intelligent sensors that communicate and coordinate in an energy-constrained environment. One of the critical research issues is the management of energy consumption for maximizing network lifetime in Wireless Sensor Networks. In a static WSN, the nodes in the sink's proximity are heavily burdened because of the many-to-one traffic pattern. So they dissipate their energy very quickly than the other nodes which are farther away towards the sink's periphery. Therefore, the sink's neighbors tend to die faster and form an energy-hole [2] around the sink resulting in network partitioning. The energy-hole problem causes the network lifetime to be shortened. However, this scenario can be changed by considering the results of [2] and [3]. The authors in [2] showed that the sensors farther away from the sink have a great amount of energy left unused, which can be up to 93% of the total initial energy [2]. [3] Proved that in a corona model, 90% of the total network energy is unused when the first node depletes their energy and this can be even higher for very large networks. MRs are potential fixes to these problems. The MRs are resource-rich nodes which can lessen the traffic load of the heavily burdened and energy-constrained sensor nodes. So in a static WSN, the problems can be modified as,

(i) The premature death of sensors in the sink's proximity which can be delayed or postponed by incorporating MRs

(ii) The huge amount of unutilized energy can be used energy-wisely to increase network lifetime by moving the MRs to energy less regions.

The idea of introducing an MR in a flat architecture was introduced in [4] for the connectivity problems. The authors in [5] framed an integer linear programming problem to maximize network lifetime allowing node mobility and MR. The MRs were first deployed in a hierarchical architecture in [6] for load balancing in clusters. And in [7], the authors have added fault tolerance to a static network with minimum number of relay nodes. The MR collects the data from the nodes when it is in their range and transports it to the sink. This causes transmission delays due to the limitation of the speed of MR [8]. So rendezvous points can be selected and the nodes can send the data via multi-hop to the points. The MR can tour the points to collect data and transport it to the sink [9].

In [10], a heterogeneous architecture with few energy-rich MRs and a large number of static nodes is proposed. The MR is introduced to lift the burden of the bottle-neck nodes around the static sink. A single MR increases the lifetime of the network by a factor of four when compared to a static network. Therefore, in a densely deployed sensor field of radius R hops, we require $O(R)$ MRs to achieve the same performance as the MS.

In [11], a three-tier architecture is proposed and analyzed to collect sensor data in sparse sensor networks. The MULES, which acts as MR, moves randomly to collect the data from the sensors, buffer and drop the packets at the wired access points. The sensors do not send the data by means of multi-hop communication to the sink, but passes the data to the MULES as they pass by, which will result in considerable energy savings. This approach increases the latency of the packets as the sensors has to wait for a MULE to pass by its location.

In this paper, we alleviate the burden of heavily loaded and energy-less nodes by providing them with MRs which take up their responsibility. We consider multiple MRs which are coordinated and moved wisely and sojourned near these nodes. Once the MRs is placed, they take up the tasks of the nodes and act as their replacement.

2. NETWORK MODEL

In case of a static network the sink is at the center of a square area with the communication range R. We assume N sensors with transmission radius r_s uniformly and randomly deployed in a square area of size $L \times L$. The topology of the network can be modeled as a fully connected graph $G(V, E)$ over the area $A = L^2$. The sensor set V can be denoted by $V = \{n_0, n_1, n_2, \ldots n_{N-1}\}$. E is the set of wireless links between the sensor nodes n_i and n_j. The nodes can communicate with each other directly if the sensor n_j is within the transmission radius r_s of n_i. Each sensor has a data generation rate G_i, the number of data units generated per time unit. A time unit is considered to be one data gathering round. In a data gathering round, each sensor generates one data unit and all the data units are relayed to the sink for processing. The scheme is independent of the underlying Medium Access Control (MAC) protocol.

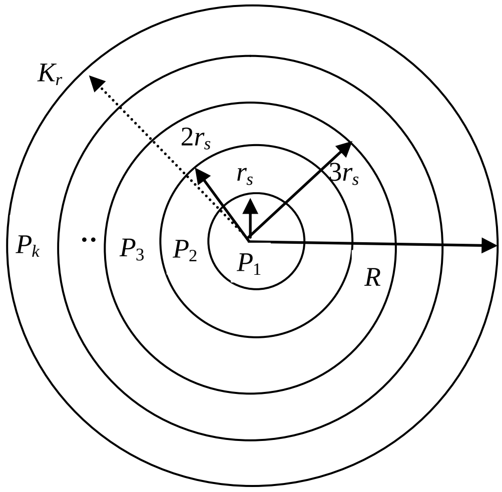

Fig.1. WSN into different logical sets

The static sensors are divided into different logical sets based on their distance from the sink as in Fig.1 and as proposed by [10]. The set P_1 contains sensors that can reach the sink within one hop and are distance r_s from the center of the sink. The set P_2 contains sensors that can reach the sink at least within two hops and whose distances are beyond r_s and within $2r_s$. The number of such sets can be determined by, $q = \left[\frac{R}{r_s}\right]$ where, $k \in \{1,2,3,\ldots q\}$. In general, the set P_k contains all sensors that can reach the sink within k hops and whose distances are beyond $(k-1)r_s$ and within kr_s. A sensor in the P_k set is represented as $n_{i,k}$ otherwise it is n_i. The union of all P_k is the sensor set V given by $V = \bigcup_{k \le q} P_k$. The number of sensors in the set P_k is $|P_k|$. We have considered ideal MAC protocol so there is no energy wastage due to collisions and retransmissions. We define the network lifetime as the time until which the first sensor $n_i \in N$ runs out of energy in the WSN as given by [4]. The Network lifetime Z, is highly dependent on the lifetime of each of the sensors which constitutes the network and is given by,

$$z = \min_{i \in N, k \in q} \tau_{i,k} \tag{1}$$

The lifetime of each individual sensor contributes to the network lifetime. So in order to prolong the network lifetime i.e. maximize Z, we have to prevent the early death of individual sensors and use the unutilized energy efficiently. The MRs can be moved to nodes that are going to die a premature death and can prolong their lifetime thereby prolonging the network lifetime.

3. PROBLEM STATEMENT AND OPTIMIZATION MODEL

The problem to maximize network lifetime using MRs can be stated as: Given an static sink, a collection of N static sensors and M MRs uniformly and randomly deployed in an area A for a data logging application with locations and energy for the sensors known a priori, our goal is to find the optimal sojourn times for the MRs at the locations of the nodes such that the network lifetime is maximized.

Table.1. Sets and Parameters for the model

V: The set $V = \{n_0, n_1, n_2, \ldots n_{N-1}\}$ made up of N sensor nodes.

M: The number of MRs

L_s: Feasible sites for MRs to sojourn near node n_s, where $L_s = \{L_1, L_2, \ldots L_{|N|}\}$, $s \in N$ and $s \subset$ V.

r: variable number of data gathering rounds

$t_{i,j}^{k}$: Time span that the MR k stays at a location i for variable number of data gathering round j where $k \in M$, $i \in L_s$ and $j \in r$.

Z: Network lifetime is defined as the time when the first sensor node dies.

E_i^o : Initial energy of node n_i in joules minus the energy required for node operation [12].

e^t: Transmission energy consumed by a node to transmit a data unit.

e^r: Energy consumed at node to receive a data unit.

E_i^k : Energy consumption rate of node n_i for receiving and transmitting packets when the MRs is rooted at a feasible site L_s.

G_i: Data generation rate which is data units generated by node n_i in a time unit.

$f_{i,j}^{k}$: Transmission rate, i.e., flow of data units from node n_i to n_j when the MRs k stays at site i for a time $t_{i,j}^{k}$.

The linear programming model given below gives the optimal sojourn time for each of the MRs at each location for network lifetime maximization. This problem of finding the sojourn time and location for each MR can be designed by the formulation. The network lifetime is divided into variable number of data gathering rounds r. At the beginning a data transport tree rooted at the sink is formed to transport the data generated by the sensors via multihip to the sink for processing. The MRs are placed for a time span of $t_{i,j}^{k}$ at M locations, coinciding with the nodes location and taking up the nodes burden during which these nodes can go to a sleep mode thus saving energy. So the problem is to find the optimal time span of the M MRs at each location so that the total network lifetime is maximized. The sets and parameters for the model are described in Table.1.

$$Maximize\ Z = \min_{k\in M}\left\{\sum\nolimits_{i\in L_s}\sum\nolimits_{j\in r} t_{i,j}^{k}\right\} \qquad (2)$$

Subject to:

$$\sum\nolimits_{\forall k\neq L_s} z\left(\sum\nolimits_{i,j\in N} e^{t} f_{i,j}^{k} + e^{r} f_{i,j}^{k}\right) \le E_i^{o} \qquad (3)$$

$$\sum_{j} f_{i,j}^{k} - \sum_{j} f_{j,i}^{k} = G_i t_{i,j}^{k} \qquad (4)$$

$$t_{i,j}^{k} \ge 0 \qquad (5)$$

The objective function Eq.(2) maximizes the network lifetime The energy constraint Eq.(3) states that the energy for receiving and transmitting packets during the entire lifetime of the network should not exceed the node's initial energy. This has not taken into account the energy when a MR takes the burden of that node. The network flow constraint is given in Eq.(4) and states that the difference between the total outflow and the total inflow of a node will be equal to the data generated. Constraint Eq.(5) ensures the non-negativity of the sojourn time for the MR.

The optimization model involves gathering of entire topology of the network and feeding it as input for the maximization model which consumes more time. As the network size increases and the number of MR increases the complexity grows exponentially and it becomes computationally infeasible to solve this model. So a scalable, distributed and localized data gathering algorithm for the movement of MRs is proposed.

4. HEURISTIC FOR AUTONOMOUS MOVEMENT OF MRS

The lifetime of the network is divided into epochs. The Heuristic starts with an initialization phase where the data transport tree is constructed. This is followed by the other two phases Data Forwarding Phase and MRs Movement Phase which is called recursively till the network lifetime is over. In the Data Forwarding Phase, the data is routed to the sink using the next-hop node for an epoch. An epoch consists of variable number of data gathering rounds. In the third phase i.e, MRs Movement Phase, a feasible location for the MRs based on energy-lessness and over-loadness of the nodes are decided. A fitness function is calculated based on the residual energy and load of the node and its neighbors. This is called as the Fitness Index (F-Index). The MRs moves to the location of nodes that are prone to die early. So the MRs stays in the location and performs the duties of that node for the entire duration of that epoch. The MRs is moved to new locations in every epoch till the end of the network lifetime.

Initialization Phase: Initially the sink is located at the center of the network area *A* and the MRs are placed randomly in the network. The minimum hop count value in each node called min-hop is set to infinity and the next-hop node is empty. The sink initiates the initialization phase by transmitting an information-packet to its one-hop neighbours. The information-packet format contains senders-id and hop-count .The sink broadcasts the information-packet with its hop-count value set to zero and senders-id as the sink-id to its neighbors. The one-hop neighbors of the sink which receives the information-packet process it and broadcasts it to their neighbors. So the information packet is broadcasted until it is received and processed by all the sensor nodes in the network.

Pseudocode for tree formation

```
min-hops = ∞;
senders-id=sink-id;
Tree_formation( )
Sink sends information-packet(sink-id, hcnt = 0) to its neighbours
For each sensor ni do
    if information-packet(senders-id, hcnt) message received then
        if hcnt < min-hops then
            min-hop = hcnt + 1
            next-hop = senders-id
            broadcast information-packet(senders-id= ni, min-hop) to neighbors of ni
        end if
    else
        ignore information-packet(senders-id = ni, min-hop)
    end if
end for
```

Each node that receives an information-packet increments the hop-count value in the received information-packet and compares it with its cached min-hop value, if the hop-count value is less than min-hop value then the node registers min-hop to the hop-count value and stores the next-hop node as the node from which the information-packet was received. Then the node broadcasts a new packet with its min-hop value and senders-id as the nodes-id to its neighbors. Otherwise, if the hop-count value is greater than or equal to the min-hop value, the processing node just ignores the information-packet. In the end, a minimum hop data routing tree routed at sink will be constructed and each node will know the minimum hop node by which it can reach the sink.

Data Forwarding Phase: Each node uniformly generates one packet of information during a round. This data-packet is routed to the sink through multi-hop communication by choosing min-hop node or the MRs if a node has a MR co-located near it. The pseudocode for data gathering phase during an epoch is,

Pseudocode for Data Forwarding

```
sink_id calculates the epoch time $t_k = \alpha.\min\left\{r_{i,j}^{est}\right\}$
For each round in $t_k$
    For each sensor ni do
        Generate one data_packet(sink_id, next_hop, data)
        forward data_packet(sink_id, next_hop, data) to
        sink_id
    end for
end for
sink_id calls MRs –movement-phase()
```

The sink takes the minimum (min) of all the residual energy estimate, $r_{i,k}^{est}$ of all the one-hop neighbors (P_1) and calculates an epoch time as $\alpha.\min\left\{r_{i,k}^{est}\right\}$ data-gathering rounds, where $0 < \alpha < 1$, $i \in N$ & $k = 1$. The MRs stays at the nodes location during an entire epoch.

MRs Movement Phase: The MRs report to the sink about their location at the end of an epoch. For the next epoch they are instructed by the sink of their new locations. Each node evaluates its health by means of a fitness index (F-Index) given by Eq.(8). This is based on the residual energy that is left over the load that is imposed on it. This can be calculated as sum of load index (LD-Index) and residual energy index (RE-Index) given by Eq.(6) and Eq.(7). The energy and load of one-hop neighbors are also considered in the calculation.

$$\text{LD}-\text{Index} = \text{Load on node} + \frac{\text{Total load of neighbors}}{\text{number of neighbors}} \tag{6}$$

$$\begin{aligned}\text{RE}-\text{Index} &= \text{Residual energy on node}\\ &+ \frac{\text{Residual energy of neighbors}}{\text{number of neighbors}}\end{aligned} \tag{7}$$

$$\text{F-Index} = \alpha(\text{LD-Index}) + \beta(RE - \text{Index}) \tag{8}$$

The weights are added to give importance to the factors and $\alpha + \beta = 1$. The F-Index of the node is piggy-banked to the data packet and sent to the sink at the end of an epoch. The sink sorts the packets and find out the nodes that are prone to early death. It identifies M such weak nodes.

Pseudocode for finding locations for MRs

```
senders-id = sink-id;
broadcast-flag = false
Finding-Locations-for-MRs(senders-id)
sink broadcast infmn_packt<senders-id>
Wait for a time to get all the infmn_reply
Label:
For each  node  ni that receive  infmn_packt<senders-id > do
{
    senders-id= ni,->id;
    send  query-request<senders-id>  to  ni  's  one-hop neighbour  set P1
    For each  pi  in P1 that receive the query-request <senders-id> packet
        send query-reply <senders-id, F-Index(pi)> to node ni
    End for
    node ni calculates the F-Index(ni)
    send infmn-reply<senders-id, F-Index(ni) >  to sink using min-hop
    broadcast-flag=True
    Forward infmn_packt<senders-id> to neighbours
    go to label
}
End for
```

It also calculates the distance of each MR against each of the weak nodes location $L_s(l_x, l_y)$ using the formula and relates an MR to a weak node whose distance is minimum. The MR moves and takes the burden of the weak node and puts to rest the weak node. The formula is given in Eq.(9),

$$d = \sqrt{(x_k - l_x)^2 + (y_k - l_y)^2} \tag{9}$$

Pseudocode for movement of MRs

```
MRs_ movement phase()
{
    MRs send their location ($x_k$, $y_k$) to the sink
    Sink Si queries for Finding-Locations-for-MRs (Si)
    Sink sorts the received infmn_reply messages and identifies the weak nodes
    Sink relates an MR with an weak node
    Sink directs the  MRs to move to the location of the weak node ($l_x$, $l_y$)
}
```

The MR stays in that position for an epoch and finishes the specified number of data gathering rounds. Then the MRs move again in the next epoch where new weak nodes are identified and helped. In the next section the simulation setup and the results obtained are discussed.

5. SIMULATION AND RESULTS

The algorithms are developed using MATLAB. In all the simulations, we consider ideal MAC layer so there is no collision or retransmission of packets thus there is no wastage of energy in this layer. The energy model used is the ideal radio model [14]. The transmission energy requires additional energy to amplify the signal according to the distance from the destination. The energy consumption formula for transmitting K-bit data packet from sensor n_i to n_j, is given by Eq.(10).

$$E_{tx} = k.E_{elec} + k.E_{amp}.d_{ij}^2 \tag{10}$$

$$E_{rx} = k.E_{elec} \tag{11}$$

The parameters, E_{elec} is the energy consumed to run the transmitter or receiver circuitry, E_{amp} is the energy for the transmitter amplifier and d_{ij} is the distance between n_i and n_j.

The energy consumed in receiving a k-bit data packet is given by Eq.(11). The value of α and β is 0.5. The simulation parameters are listed in the Table.2.

Table.2. Parameters and value for simulations

Parameter	Value
Area side	100m,100m
Number of sensor nodes	100
Number of Mobile relays	10
Node Deployment	Random and uniform
Sensor Transmission radius	2m

Sink communication range	10m
Initial energy in each node	3J
E_{elec}	50 nJ/bit
E_{amp}	100 pJ/bit//m^2
Data Size	1000 bits
Data routing	Shortest path routing

The Fig.2 compares how the network lifetime varies when the initial energy of the sensors are varied at the time of deployment. In this experiment, 100 sensors are deployed in a 100 × 100 m square area and the initial energy of the sensors is varied with 2, 3, 4, and 5 joules respectively.

The results prove that as the initial energy increases there is a comparative increase in the network lifetime. The random moving MRs also increases the lifetime even though every movement of the MRs is energy unconscious. The proposed heuristic outperforms the other schemes because of wise movement of the MRs to the weak nodes and thus enhancing the network lifetime.

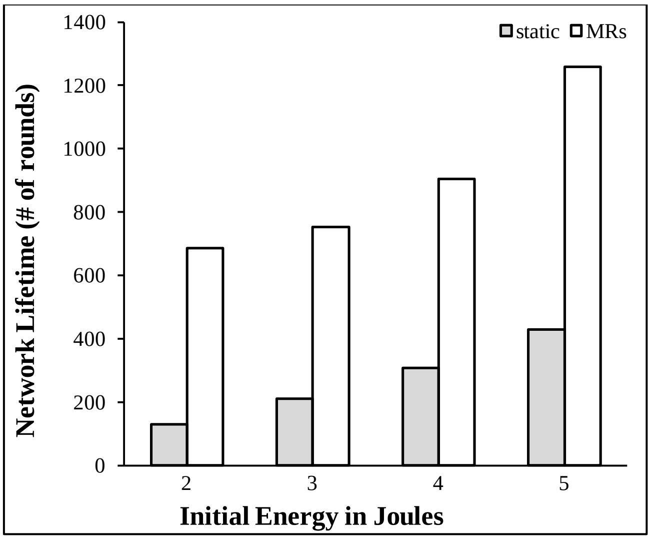

Fig.2. Network Lifetime vs. Initial Energy

The Fig.3 shows the average residual energy of the network of 500 sensors after 100, 200, 300, 400, 500, 600 and 700 rounds respectively. The average residual energy drops suddenly in the static sink case, while in the proposed scheme the energy decreases gracefully resulting in a longer network lifetime. This is because of the identification of weak nodes and enhancing its lifetime which results in increase in network lifetime.

The Fig.4 compares the network lifetime for different number of MRs deployed in 100 × 100 m with 100 sensors. The initial energy of the sensors is considered to be 3 joules. In the static case, it does not have any impact on the network lifetime and it remains constant. But in the proposed strategy, there is an immense improvement in lifetime for higher values of MRs The reason is, the MRs helps more weak nodes thereby increasing the network lifetime. The lifetime grows exponentially with the number of MRs. But there is an upper limit to the number of MRs that can be used in the network. This can be considered for further research.

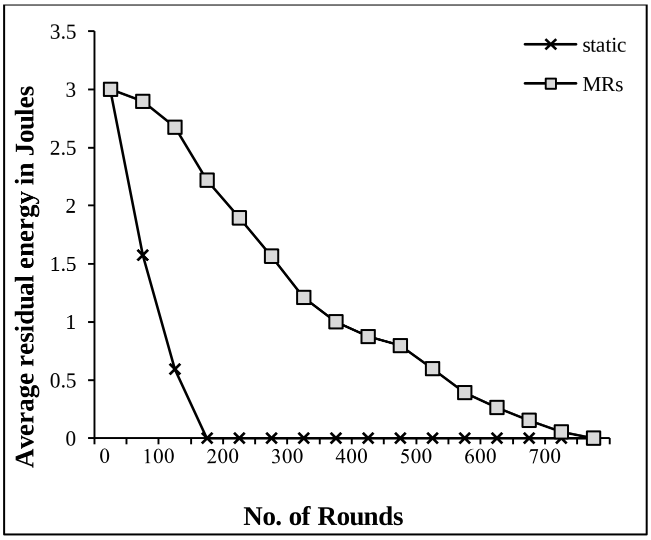

Fig.3. Number of rounds vs. Average Residual Energy

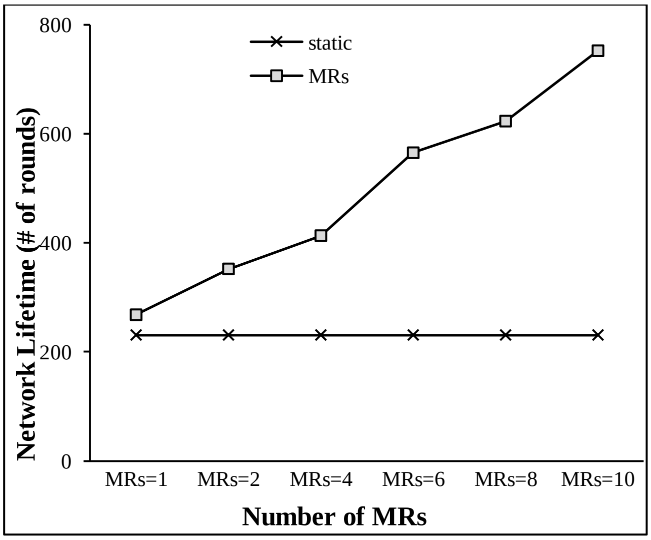

Fig.4. Network Lifetime vs. Number of MRs

6. CONCLUSION

The analysis and the simulation results show that adding MRs to a static network definitely improves the performance of a network when compared to a static network. The following conclusions can be derived from our paper.

a. Energy is used fairly among all the sensors in the network.

b. Weak nodes which might cause the network lifetime to end are identified and their life is elongated with the help of the MRs.

c. Increase in the number of MRs certainly increases the network lifetime exponentially.

d. Total energy wastage in the network is minimized when compared to the static network.

The advantage of using an MR when compared to other mobile entities is, if it fails, the basic network functionalities will still be working, but the network may not avail the services of MR. To conclude the above study, the MRs can definitely be added to a static network to improve the lifetime. Further it can also be used to improve the tracking quality, to keep the network connected, and to improve coverage.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless sensor networks: A survey”, *Computer Networks*, Vol. 38, No. 4, pp. 393-422, 2002.

[2] Jian Li and Prasant Mohapatra, “Analytical modeling and mitigation techniques for the energy hole problem in sensor networks”, *Pervasive and Mobile Computing*, Vol. 3, No. 3, pp. 233-254, 2007.

[3] Jie Lian, Kshirasagar Naik and Gordon B. Agnew, “Data capacity improvement of wireless sensor networks using non-uniform sensor distribution”, *International Journal of Distributed Sensor Networks*, Vol. 2, No. 2, pp. 121-145, 2006.

[4] Xiuzhen Cheng, Ding-Zhu Du, Lusheng Wang and Baogang Xu, “Relay Sensor Placement in Wireless Sensor Networks”, *Wireless Networks*, Vol. 14, No. 3, pp. 347-355, 2008.

[5] K. Dasgupta, M. Kukreja and K. Kalpakis, “Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks”, *Proceedings of Eighth IEEE International Symposium on Computers and Communication*, Vol. 1, pp. 341-348, 2003.

[6] G. Gupta and M. Younis, “Load-balanced clustering of wireless sensor networks”, *Proceedings of IEEE International Conference on Communications*, Vol. 3, pp. 1848-1852, 2003.

[7] Hai Liu, Peng-Jun Wan and Xiaohua Jia, “Fault-Tolerant Relay Node Placement in Wireless Sensor Networks”, *Proceedings of 11th Annual International Conference on Computing and Combinatorics*, pp. 230-239, 2005.

[8] Arnab Chakrabarti, Ashutosh Sabharwal and Behnaam Aazhang, “Using predictable observer mobility of power efficient design of sensor networks”, *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*, pp. 129-145, 2003.

[9] Guoliang Xing, Tian Wang, Zhihui Xie and Weijia Jia, “Rendezvous planning in wireless sensor networks with mobile elements”, *IEEE Transactions on Mobile Computing*, Vol. 7, No. 12, pp. 1430-1443, 2008.

[10] Wei Wang, Vikram Srinivasan and Kee-Chaing Chua, “Extending the Lifetime of Wireless Sensor Networks Through mobile relays”, *IEEE/ACM Transactions on Networking*, Vol. 16, No. 5, pp. 1108-1120, 2008.

[11] Rahul C. Shah, Sumit Roy, Sushant Jain and Waylon Brunette, “Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks”, *Ad Hoc Networks*, Vol. 1, No. 2-3, pp. 215-233, 2003.

[12] Z. M. Wang, S. Basagni, E. Melachrinoudis and C. Petrioli, “Exploiting sink mobility for maximizing sensor networks lifetime”, *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 287a, 2005.

[13] Wei Wang, “Increasing Lifetime of Wireless Sensor Networks”, Ph.D. diss., Department of Electrical and Computer Engineering, National University of Singapore, 2008.

[14] W. Heinzelman, “Application-specific protocol architectures for wireless networks”, Ph.D. diss., Massachusetts Institute of Technology, 2000.

[15] Stefano Basagni, Alessio Carosi, Emanuel Melachrinoudis, Chiara Petrioli and Z. Maria Wang, “Controlled sink mobility for prolonging wireless sensor networks lifetime”, *Wireless Networks*, Vol. 14, No. 6, pp. 831-858, 2008.