Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 6 (4): 389-398, 2014
ISSN 1815-7432 / DOI: 10.3923/rjit.2014.389.398

Load Balancing Clustering in WSN using MABC

N. Ashok Babu and R. Kumar
Department of Electronics and Communication Engineering, Sri Ramaswamy Memorial (SRM) University, Kattankulathur, 603203, Kanchipuram, India

Corresponding Author: N. Ashok Babu, Department of Electronics and Communication Engineering, Sri Ramaswamy Memorial (SRM) University, Kattankulathur, 603203, Kanchipuram, India Tel: +91 9843782289 Fax: +91 44-27453622

ABSTRACT

Wireless Sensor Networks (WSN) are composed of a large array of independent sensors called nodes. These nodes are inexpensive and are used in data gathering usually in hostile environments. They collect data from physical or environmental conditions, such as temperature and pressure etc. and send it to a base station. They are limited in efficiency, secure routing and load sharing. To overcome these issues, clustering is performed in WSN which partitions the network into groups. Each group selects a cluster head which gathers data from all the nodes and sends it to the base station. These cluster heads are selected using various algorithms which are based on residual energy and distance between nodes and base station. The proposed algorithm is based on the modified version of the Artificial Bee Colony (ABC) algorithm which is adapted to allow quick convergence and improved search area in selection of cluster heads. It is used to effectuate the improvement in load balancing of clusters in wireless sensor network which also improves the network lifetime. The simulation results show that the proposed approach is more efficient than other distributed algorithms. This technique can be easily extended to large scale wireless sensor networks.

Key words: Load balance, network lifetime, routing protocol, WSN, cluster based routing

INTRODUCTION

Wireless Sensor Networks (WSN) consist of small intelligent sensors which are deployed to collect physical data from environments without human intervention. They are usually packed with small batteries and have limited lifetime. They are usually placed in hostile environments for tactical purposes. They are used in applications such as object tracking, intrusion detection, environmental monitoring, traffic control, inventory management in factory and health related applications and so on (Ahmed *et al*., 2005; Kumar *et al*., 2013). The data from these sensor nodes is critical in most of the cases. Since the sensors have limited life time due to their small battery capacity, energy efficient techniques are required to prolong their lifetime.

Various methods have been proposed to prolong the lifetime of the wireless sensor networks. The nodes are usually placed in dense manner (Sharma and Mittal, 2013). Closely spaced nodes may collect similar data. It becomes redundant for a number of nodes to send similar data. Several algorithms have been presented to increase the life time of the network by limiting the nodes from sending redundant data. Certain nodes can be made inactive for a period of time. The energy required to send the data to the base station increases squarely with distance from the base station to the nodes. Therefore, nodes which are far away, lose energy quickly while transmitting data to the base station than the nodes which are nearer to the base station.

This led to the method of clustering which split the entire network into groups of nodes which are nearer in distance and transmit similar data. Not all nodes transmit data to the base station, a single node in a cluster aggregates the collected data from all the nodes in the cluster and transmits data to the base station. This node is called as a Cluster Head (CH) (Velrani, 2013). Cluster heads tend to lose more energy than normal nodes because they transmit data to a larger distance than the normal nodes. Therefore, cluster heads are swapped periodically for each round. The energy consumption in a cluster is highly related to the selection of cluster head and they had to be in a rotational basis. The selection of cluster head also determines the load sharing between the clusters and network lifetime.

In this study, an algorithm using MABC is proposed which elects the cluster heads in the network using the fitness value of each node and thus the load in the network is balanced uniformly by increasing the network lifetime.

The deployment of Wireless Sensor Networks are designed in such a way that they can withstand the limiting conditions of the network such as battery life and mobility etc. Various algorithms have been proposed which are much effective in overcoming these limitations.

Among them, three algorithms LEACH, PEGASIS and swarm intelligence algorithm Artificial Bee Colony (ABC) are considered in this study.

Low Energy Adaptive Clustering Hierarchy (LEACH) is one of the major improvements in wireless sensor networks clustering technology (Dasgupta and Dutta, 2012). The LEACH manages load stress among the cluster heads by rotating the cluster heads in each round. The nodes which are not a cluster head in the previous round are used as a pool in which the cluster head is selected based on the probability function.

The decision of whether a node is elevated to a cluster head is made dynamically at each interval (Shukla, 2013). The cluster head is selected by:

$$T(n) = \frac{p}{1-\left(r \bmod \frac{1}{p}\right)} \text{ if } n \in G \tag{1}$$

where, n is the given node, P is the probability of a node being elected as a cluster head, r is the current round number and G is the set of nodes that have not been elected as cluster heads in the last 1/p rounds.

Although, LEACH provides a way in which the load can be shared with the nodes, it does not consider the energy of the nodes and at the distance they are from the base station. Hence, the load variance in LEACH algorithm is higher due to its randomness in selection of the cluster head.

Power Efficient Gathering Systems in Sensor Intelligence Systems (PEGASIS) is a grid based system which proposes the formation of node chains which results in lesser transmission distance for nodes. At any instance of time, only one node will be transmitting towards the base station which results in energy savings (Guo *et al*., 2010; Lindsey and Raghavendra, 2002).

PEGASIS enforces the concept of energy savings at the node level rather than the hierarchical level. Data aggregation occurs at every node in the cluster. Eventhough, the energy consumed is lesser, the load variance is not uniform since the nodes at the end of the chain always handle larger number of data than the nodes at the start of the chain. Also, the residual energy of the nodes is not considered. Hence, there is a possibility of increased rate of dead nodes at the end of the rounds.

ABC algorithm was first introduced by Karaboga (2005). It has been inspired from the collective behavior of bees as seeking food (nectar). Bees return to their hive when they find food and inform other bees about the position and quality of the food with a series of particular movements. In this algorithm, there are three groups of bees, the first group is referred to as employed bees and they look around to find a better food source. The second group consists of the onlooker bees which will analyze the information received from the employed bees about the location and quality of potential food sources and will choose a food source. The last group includes scout bees whose duty is to find new food sources (Karaboga and Basturk, 2007). The algorithm is summarized as follows:

- **Step 1 (Initialization):** Half of the population is assigned to the solution space. Food is randomly generated and it's fitness value is calculated. Employed bees take this portion of the population
- **Step 2 (Move the onlookers):** The possibility of any food source, being chosen by the onlookers, is calculated and they will move to their new positions
- **Step 3 (Moving the scouts):** If there is no improvement in the fitness value of the employed bees after a certain number of iterations, known as limit, the respective food source will be ignored and the employed bees will be changed into scout bees and will seek for new food sources
- **Step 4 (Updating the best position):** The fitness of the new food source is compared with the old food source and in case the new food source is better then it will replace the position of the old food source
- **Step 5 (Termination checking):** In case the termination constraint of the algorithm (number of iterations or a certain fitness value) is satisfied, the algorithm is finished otherwise it repeats again from step 2

The algorithm is used to select the cluster head based on a fitness function. For each round, selections of possible cluster heads for next round are found out. These selections are evaluated with the fitness function. The best selection set is chosen as cluster heads for the next round. When these clusters are formed, the cluster heads for next round is chosen evaluating a new set of selections by the algorithm.

This method provides an improved way of selecting the cluster heads but swarm intelligence algorithms are slower in convergence which limits their operation in a real time environment.

Artificial Bees Algorithm is a stochastic meta-heuristic algorithm which generates random solutions and evaluates them with the fitness function. In ABC algorithm, new random solutions are formed by changing only one parameter at a time. This results slower convergence rates and reduced randomness. To improve these areas, the following modifications are made in the algorithm (Karaboga *et al*., 2014).

In ABC, only one control parameter is available for controlling the randomness in the new solution formed. This is called the 'limit'. This single control parameter is not sufficient to provoke greater randomness in the new solution that is generated. Hence, the modification parameter is introduced called 'frequency of perturbation' (Akay and Karaboga, 2012).

In the basic version of ABC, only one parameter is changed at a time. This is improved by introducing Modification Rate (MR). This control parameter decides whether the entire parameter set of the solution is to be varied. If a uniformly distributed random number, which varies from

0-1 ($0 \le R_{ij} \le 1$), is lesser than the modification rate, then the entire parameter set is changed to a new random parameter set. The parameter set, here, refers to the cluster head set whose fitness is evaluated using the fitness function. Mathematically, this is expressed as:

$$v_{ij} = \begin{cases} x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right), & \text{if } R_{ij} < MR \\ x_{ij}, & \text{otherwise} \end{cases} \tag{2}$$

where, x_{ij} is the parameter set, v_{ij} is the new formed parameter set, ϕ_{ij} is the perturbation magnitude and k = {1, 2, ..., SN} is randomly chosen index that has to be different from i. The value of MR is between 0 and 1. The higher value of MR increases the probability of random parameter set being generated which creates too much diversity in the population and a lower value reduces the randomness in the population. The solution set is the set of node numbers which are evaluated with fitness so that the solution set with higher value of fitness is selected for the next round.

The frequency of perturbation controls the rate at which the randomness is imparted into the generated solution. Another control parameter called the 'magnitude of perturbation' is used to affect the degree in which the generated new solution is different from the existing solution. In basic ABC, a scout phase is available which prevents the solutions becoming stuck at a single area resulting in a localized solution. If the 'limit' is crossed, the new solutions are randomly selected in the search space. But this process does not take place in employed bee phase and onlooker bee phase.

The modification rate imparts these features in the employed bee phase and onlooker bee phase. But instead of generating an entirely new random solution set, the randomness of the solution is determined by the factor called magnitude of perturbation, a real number denoted by ϕ_{ij}. The value of ϕ_{ij} varies within the range [-1, 1] in the basic ABC while, it varies within the range [-SF, SF] in the modified ABC algorithm (Karaboga *et al.*, 2014).

The value of SF determines the speed of search. Lower value of SF may result in finer search but reduces the overall speed of convergence of the algorithm. Higher values of SF may increase the speed of convergence of the algorithm but may reduce the exploitation of the search space by the algorithm.

The magnitude of the perturbation is controlled by a control parameter called the Scaling Factor (SF). This value is set before running the algorithm. The scaling factor may be tuned automatically (ASF) which results in improved convergence behavior of the algorithm allowing difference resolutions of search at different periods of time. The SF is chosen using the following equation:

$$SF(t+1) = \begin{cases} SF(t) \times 0.85 \text{ if } \varphi(m) < 1/5 \\ SF(t)/0.85 \text{ if } \varphi(m) > 1/5 \\ SF(t) \text{ if } \varphi(m) = 1/5 \end{cases} \tag{3}$$

The new solution produced is dependent on the present solution, the difference between the present solution and the random solution (x_i and x_k) and also the magnitude of perturbation. Thus, the magnitude of perturbation controls the level of randomness of the solution and the frequency of perturbation controls the rate at which these random solutions are generated.

METHODOLOGY

MABC in cluster head selection and cluster formation: MABC is used to determine the cluster heads where each solution represents an array having k items in which every item consists of identity of the node which is to be selected as the cluster head for the next round. The fitness of cluster heads selection is stated as a fitness value which is in inverse proportion to the amount of energy consumption for a round. A certain transfer time is required for a data package and then the energy consumption is calculated by multiplying transmitting power (P^s) and the time (t).

$$E = \sum_{i=1}^{m}\left(P_i^s \cdot t\right) \tag{4}$$

$$E \geq \alpha \cdot \left(\sum_{i}^{m} d_i^2 + b^2\right) \cdot t \tag{5}$$

where, m is the No. of nodes, i is the node index, d_i is the distance between ith node and cluster head, b is the distance between cluster-head and the base station, a is the energy constant and E is the transfer energy of the cluster.

Considering multiple clusters, the calculation of minimum energy consumption emphasizing the effect of distances will be as in Shukla (2013) expressing sum of the energy consumptions of clusters.

If weight (w) is taken as the multiplication of a and t, transfer energy can be calculated as:

$$E \geq w \cdot \sum_{j=1}^{n}\left(\sum_{j=1}^{m} d_i^2 + b_j^2\right) \tag{6}$$

where, j is the cluster index, d_{ij} distance between ith node and jth cluster-head, b_j is the distance between jth cluster-head and the base station.

In order to minimize energy consumption, distances between nodes and cluster-heads and distances between cluster-heads and the base station are considered in the selection process. Since a cluster-head should have enough energy to feed the communication in the current round as a managing element, energy levels of the candidate nodes also have equal importance in the selection. A candidate cluster head should provide enough energy of receiving messages (E_{RX}) from the nodes in the cluster and transmitting the fused message (E_{TX}) to the base. According to these considerations, fitness function (f_{dist}) is expressed (simply inverse of the energy consumption) and the constraints are given where energy consumptions of E_{TX} and E_{RX} are modeled as in (Dorigo *et al.*, 1991).

$$f_{dist} = \left[w \cdot \sum_{j=1}^{n}\left(\sum_{j=1}^{m} d_i^2 + b_j^2\right)\right]^{-1} + \sum_{j}^{n}\left[\frac{b_j^2}{nnj}\left(ScE_j\right)\right] \tag{7}$$

where, i is node index, j is cluster index, E_i is energy level of the ith node, m_j is No. of nodes in the jth cluster, ScE is cluster head energy, nn_j is No. of nodes in jth cluster, d_{ij} is distance between the ith node and jth cluster.

The second part of the fitness function in Guo *et al.* (2010) focuses on the energy of the cluster head as a selection criterion. For effective load sharing, the cluster sizes at greater distances from the base station must be larger than the cluster sizes nearer to the base station. Hence, the part two of the fitness function chooses the clusters with larger sizes if present at longer distances from the base station provided they have sufficient energy.

The addition of the energy as part of the fitness function makes the selection of cluster heads with sufficient energy and prevents the selection of nodes with lesser energy. This is a major improvement from distance based algorithms which occasionally results in the selection of nodes with lesser energy further increasing the number of dead nodes.

$$E_j \geq \left(mj \cdot E_{RX} + E_{TX}\right)$$

$$E_{RX} = E_{elec} \cdot k$$

$$E_{TX} = E_{elec} \cdot k + E_{RX} \cdot k \cdot b_j^2 \tag{8}$$

where, E_{TX} is transmit energy, E_{RX} is receive energy, E_{amp} is amplifier energy, E_{elec} is data aggregation energy and k is No. of bits in message.

Algorithm for building the cluster using MABC: For a sensor network with N nodes and k number of clusters, the sensor network can be clustered as follows:

- **Step 1:** Initialize population, each individual of the population has a set of node numbers which are to be elected as cluster heads
- **Step 2:** Evaluate the fitness function of each individual, where:
  - For each node n_i in the network
  - Calculate distance (n_i, CH_k) between n_i and all cluster heads CH_k
  - Assign n_i to CH_k where distance (n_i, CH_k) = $\min_{vk=1,\dots,K}$ {distance(n_i, CH_k)}
  - Compute the fitness function

Step 3: Perform the position update by the optimization algorithm
Step 4: Repeat steps 2-4 until the maximum number of cycle is reached

Cycle phase: The cluster is stable for a while until the process of re-electing cluster head is triggered in T(k). Once the re-electing phase is triggered, the algorithm generates new solution sets of size k, where k is the number of clusters (Forero *et al*., 2011). For this round, the item size k may not be the same as the value in the previous case but it is limited with boundary values and chosen randomly between these boundary values. The boundary values are updated for each round based on the number of nodes which are available for selection which includes number of alive nodes. The algorithm now generates best solution from this population by evaluating individuals using the fitness function and it is reciprocated in each round.

Measuring the load balancing in cluster: To measure the degree of load balancing, the variation of the load in the clusters σ^2 is defined as:

$$\sigma^2 = \sum_i^k \frac{\left((N/K) - l_i\right)^2}{ck} \tag{9}$$

where, k is No. of clusters formed, N is No. of nodes, l_i is load of the ith cluster (load variance is in unit of Joules^2/cluster).

Load variance is a direct measure of non-uniformity of load between cluster heads. If the load variance value is higher then there is a possibility that some cluster heads in the network are rapidly losing energy. This will affect the overall energy consumption of the network. Hence, reducing the load variance will improve the energy efficiency of the cluster and increase the network lifetime.

After finishing the setup phase, the steady state phase will start and nodes transmit data. When all the nodes within the cluster finish sending data, the intermediate nodes perform some computation on it and send it to base station using multihop communication.

RESULTS AND DISCUSSIONS

In the simulation experiments, WSNs nodes are randomly distributed in the 100×100 m area. The proposed algorithm is compared with the classical LEACH, PEGASIS and ABC by observing the network lifetime and load variance especially. The simulation was carried out in MATLAB R2012b.

The simulation parameters used in this study are detailed in Table 1. To facilitate unified comparison, the survival time was represented by the number of rounds where each round begins with a set-up phase when the clusters are organized, followed by a steady-state phase in which the data will be transmitted to the base station.

Figure 1 the network lifetime comparison of various algorithms as simulated in MATLAB. The simulation results show that the proposed algorithm has an improved network lifetime compared

Table 1: Simulation parameters

Parameters	Value
Area	100×100 (m)
Location of base station	(50, 50)
Number of nodes	100
Initial energy	0.5 J
E_{elec}	500 nJ bit^{-1}
Packet size	4000 bits
E_{fs}	10 pJ^{-1} bit m^{-2}
E_{amp}	0.0013 pJ^{-1} bit m^{-4}

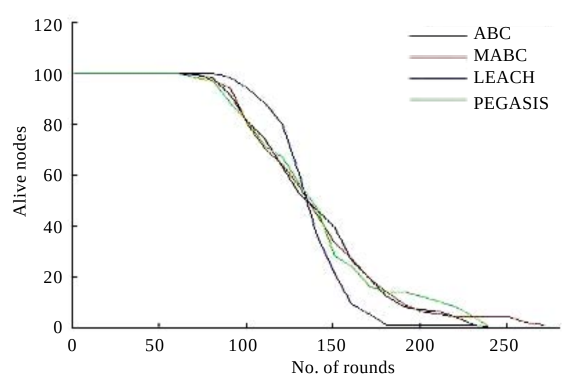

Fig. 1: Comparison of network lifetime

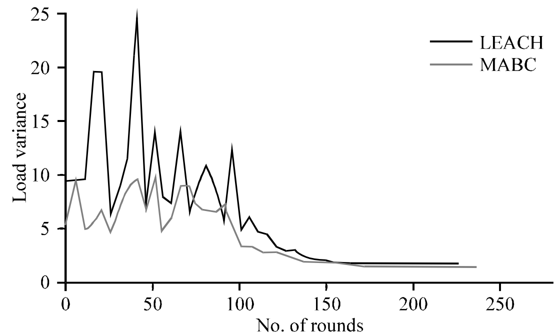

Fig. 2: Comparison of load variance between MABC and LEACH

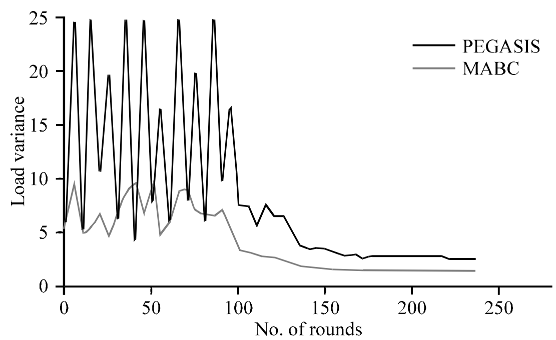

Fig. 3: Comparison of load variance between MABC and PEGASIS

to other algorithms. Even after 230 rounds, 2% of nodes in MABC algorithm are still alive whereas, all the nodes will be dead in LEACH, PEGASIS and ABC algorithm. The network life time of MABC is at least 20 rounds better than ABC. The network lifetime is nearly 80 rounds better than LEACH algorithm. MABC prolongs the network lifetime and it was also found that the alive nodes in the network are symmetrically placed in the sensing area improving the coverage.

Figure 2 shows the comparison of load variance between proposed algorithm and LEACH algorithm. The proposed algorithm was nearly twice as best to LEACH algorithm in balancing the load between clusters. There was more difference between the two algorithms in the initial stages of the network. Load variance of MABC varied from 2-10 Joules whereas LEACH varied from 2-25 Joules.

In Fig. 3, the proposed algorithm was nearly thrice as best to PEGASIS algorithm in balancing the load between the clusters. During the initial number of rounds the load variation is more in PEGASIS when compared with MABC. Load variance of MABC varied between 2-5 Joules whereas LEACH varied from 4-25 Joules.

Figure 4 shows the comparison of load variance between proposed algorithm and ABC algorithm. The difference between the two algorithms was lesser compared to previous algorithms. The proposed algorithm was nearly 10-15% better than ABC in balancing the load between clusters.

Figure 5 shows the histogram of cluster head distribution in the rounds. The distribution was uniform which translates the high energy utilization.

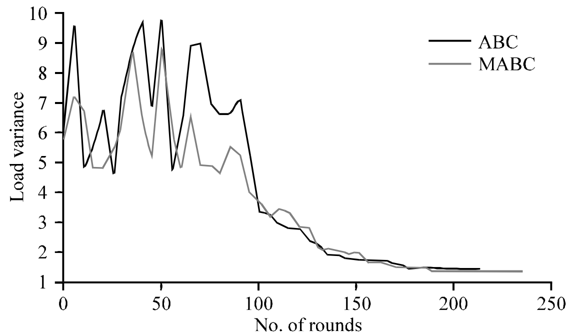

Fig. 4: Comparison of load variance between MABC and ABC

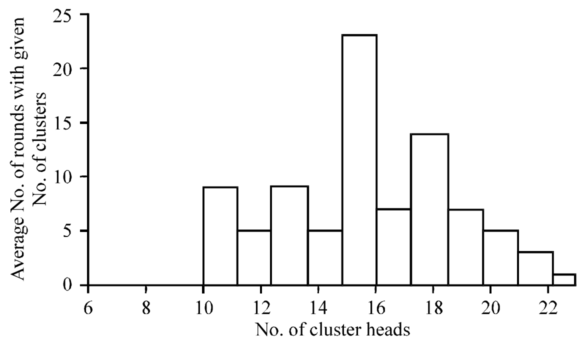

Fig. 5: Cluster head histogram

CONCLUSION

This study proposes an algorithm using MABC for CHs selection to increase the network lifetime of the WSNs. Based on information such as the number of nodes in the cluster and using the modified artificial bee colony optimization fitness function, current CHs choose the next set of CHs in each cluster. The simulation demonstrates that the proposed algorithm balances the load and increases the network lifetime. Because each node can obtain information without additional traffic or complex computations, the proposed algorithm remains simple and practical while prolonging the network lifetime. Therefore, the proposed algorithm can be an effective tool for data aggregation in WSNs.

ACKNOWLEDGMENT

We thank Department of Science and Technology (DST-India) for the financial support provided by INSPIRE Fellowship (DST 2012/650) for carrying out this study.

REFERENCES

Ahmed, N., S.S. Kanhere and S. Jha, 2005. The holes problem in wireless sensor networks: A survey. ACM SIGMOBILE Mobile Comput. Commun. Rev., 9: 4-18.

Akay, B. and D. Karaboga, 2012. A modified Artificial Bee Colony algorithm for real-parameter optimization. Inform. Sci., 192: 120-142.

Dasgupta, S. and P. Dutta, 2012. A democratic head election strategy for clustering in wireless sensor network. Int. J. Comput. Applic., 54: 32-36.

Dorigo, M., V. Maniezzo and A. Colorni, 1991. Positive feedback as a search strategy. Technical Report No. 91-016, Politecnico di Milano, Italy.

Forero, P.A., A. Cano and G.B. Giannakis, 2011. Distributed clustering using wireless sensor networks. IEEE J. Sel. Top. Sign. Process., 5: 707-724.

Guo, W., W. Zhang and G. Lu, 2010. PEGASIS protocol in wireless sensor network based on an improved ant colony algorithm. Proceedings of the 2nd International Workshop on Education Technology and Computer Science, Volume 3, March 6-7, 2010, Wuhan, China, pp: 64-67.

Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, October 2005. http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf

Karaboga, D. and B. Basturk, 2007. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. J. Global Optim., 39: 459-471.

Karaboga, D., B. Gorkemli, C. Ozturk and N. Karaboga, 2014. A comprehensive survey: Artificial Bee Colony (ABC) algorithm and applications. Artif. Intell. Rev., 42: 21-57.

Kumar, C.K.S., R. Sukumar and M. Nageswari, 2013. Sensors lifetime enhancement techniques in wireless sensor networks: A critical review. Int. J. Comput. Sci. Inform. Technol. Secur., 3: 159-164.

Lindsey, S. and C.S. Raghavendra, 2002. PEGASIS: Power-efficient gathering in sensor information systems. Proceedings of the Aerospace Conference, March 9-16, 2002, Los Angeles, CA., USA., pp: 1125-1130.

Sharma, S. and P. Mittal, 2013. Wireless sensor networks: Architecture, protocols. Int. J. Adv. Res. Comput. Sci. Software Eng., 3: 303-308.

Shukla, K.V., 2013. Research on energy efficient routing protocol LEACH for wireless sensor networks. Int. J. Eng. Res. Technol., 2: 1-5.

Velrani, K.S., 2013. Detection of false data distribution nodes in wireless sensor networks. Int. J. Eng. Comput. Sci., 68: 16-20.