Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 5040475, 11 pages
http://dx.doi.org/10.1155/2016/5040475

Research Article

Balanced Cluster Head Selection Based on Modified *k*-Means in a Distributed Wireless Sensor Network

Sasikumar Periyasamy, Sibaram Khara, and Shankar Thangavelu

SENSE, VIT University, Vellore, Tamil Nadu 632 014, India

Correspondence should be addressed to Sasikumar Periyasamy; sasikumar.p@vit.ac.in

Received 29 October 2015; Revised 2 February 2016; Accepted 11 February 2016

Academic Editor: Antonio Lazaro

A major problem with Wireless Sensor Networks (WSNs) is the maximization of effective network lifetime through minimization of energy usage in the network nodes. A modified *k*-means (*Mk*-means) algorithm for clustering was proposed which includes three cluster heads (simultaneously chosen) for each cluster. These cluster heads (CHs) use a load sharing mechanism to rotate as the active cluster head, which conserves residual energy of the nodes, thereby extending network lifetime. Moreover, it reduces the number of times reclustering has to be done and significantly increases the number of data packets sent during network operation. The results show that *Mk*-means (modified *k*-means) algorithm was found to outperform the existing clustering algorithms owing to its unique multiple cluster head methodology.

1. Introduction

Distributed wireless microsensor networks have become very important components in the day-to-day world of data computing. Miniature in size is the key factor for designing microsensors. The battery in the sensor node is a major constraint of the miniaturization process. It can only be reduced to a particular degree due to energy density. In addition to the improvements to the energy density factor, consumption of energy can also be reduced in sensors. This approach requires the implementation and usage of low power hardware. Also the operational lifetime of sensor nodes can be extended by optimizing applications, operating systems, and protocols used for communication. In certain scenarios, the on-board components were switched off when they are not sensing or transmitting data. That is, they are idle.

Categorizing sensor nodes to groups (by forming clusters) has become common among researchers in recent years. This clustering approach led to the development of more number of clustering algorithms. Clustering guides the nodes to organize themselves into multiple clusters, with one sensor node acting as the CH. And the rest of the noncluster head nodes transmit sensed information to their respective CHs, while the CHs aggregate the collected information and send it to the distant BS.

The current scheme of clustering addresses the self-organization and scalability properties with rotational based CH selection. Thereby a group is led by master (CH) node with energy conservation paradigm to send the data to Base Station (BS). We propose here the modified *k*-means algorithm with more than one CH in a cluster to lead the group. The sensor nodes in the cluster will choose one of the CHs to send the data either distance based or schedule based. So that the time and energy spent during the reclustering phase are reduced by decreasing the number of reclustering rounds. Further, various applications demand extremely efficient support for top-*k* queries in getting data from WSN. Hence, adding the above processing gradually reduces the data traffic when data forwarding happens on query/requirement based to the BS.

The rest of the paper is organized as follows. Detailed literature survey on selected clustering algorithm is presented in Section 2, followed by *k*-means clustering algorithm and *Mk*-means clustering algorithms in Sections 3 and 4, respectively. In Section 5, the performance is analyzed comparing the algorithm proposed through simulations. Finally, conclusions are presented in Section 6.

2. Literature Survey

The last few years have witnessed a rapid increase in the interest in the usage of WSN in various applications like disaster management, habitat monitoring, and military surveillance. The sensor nodes in these environments were deployed randomly and expected to operate independently in unsupervised area. In this larger number of sensor nodes scalability is addressed by generally grouping the sensor nodes into multiple nonoverlapping clusters. In [1], a detailed taxonomy and generalized classification of various clustering schemes were presented.

The clustering process aims to extend the lifespan of sensor nodes [2–5]. Thereby by increasing the lifetime of the CH, one will be able to minimize the need for reclustering and less number of overheads exchanged between a microsensor node and CH. The real challenge considered in the field of clustering algorithms is how to influence the longevity of sensor nodes in a WSN, through comparison of initial energy with the residual energy of the node during each cycle or round of communication [3]. To simplify, the depleted battery causes the sensor nodes to fail soon.

Linked Cluster Algorithm (LCA). Baker and Ephremides [6, 7] are the first authors to consider clustering in WSN. They focused on creating an efficient topology of network that had the capability to handle the mobility of the nodes. By performing network clustering [1], the CHs are expected to form a backbone network to which cluster members can connect while on the move. The objective of the proposed distributed algorithm is to form many clusters such that a CH is independently connected to all the sensor nodes in its own cluster. LCA is thus designed to maximize connectivity of the network. But this algorithm assumes that the nodes are synchronized to a master clock and medium access which is time-based.

Adaptive Clustering. Lin and Gerla [8] analyzed the role of support of multimedia applications in a general multihop ad hoc mobile network using CDMA based medium arbitration [1]. To minimize the delivery of data the network is clustered and a distinct code would be assigned to the cluster. Similar to [6, 9], an ID-based selection scheme of clusters is employed. Like LCA, in this too, a one-hop intracluster topology can be established. A CH decides the codes of communication with the neighboring CHs. This algorithm tries to control the size of the cluster optimally by balancing the interest in the reuse of channels spatially, which can be enhanced by having small clusters, and a certain data delivery delay, which decreases by bypassing the intercluster routing technique, that is, large cluster sizes. Similarly as in LCA [1], TDMA is used in intracluster communication. Each cluster would use a unique code resulting in the simple implementation of the same. Great potential for reaching up to the QoS requirements can be often found in these multimedia applications.

Weighted Clustering Algorithm (WCA). WCA selects a particular node as a CH depending on the number of surrounding neighbors, power of transmission, battery-life, and rate of mobility of the node [10, 11]. WCA also restricts the number of sensor nodes in a particular cluster, thereby preventing degradation of the performance of the MAC protocol.

Distributed Clustering Algorithm (DCA). In DCA wireless sensor nodes are assigned various weights on preset criteria. The more is the weight of the node, the better it would be for playing the role of a CH. The major advantage of this technique [12] is that, by representing the nodes with the help of certain weights, with parameters which are related to the mobility of the nodes, we can choose to give the role of CHs to those nodes that are better qualified. A very interesting result regarding this is that the complexity pertaining to the time parameter of the DCA is limited by a parameter of the network that depends on the topology of the network (that may change due to mobility of the sensor nodes) rather than on the network size. The DCA [12] is very much apt for clustering "quasi-static" ad hoc networks.

k-Medoids Clustering Algorithm. The k-medoids algorithm is related very closely to the k-means algorithm and the medoid shift algorithm. k-means and k-medoids algorithms have the ability to split the WSN to groups and both of these algorithms try to minimize the distance between points designated to be in a cluster and a specific point designated as the center of that cluster. As compared to the k-means algorithm, k-medoids choose data points as centers (exemplars) and they work with any arbitrary matrix comprising of distances between data points [13]. k-medoids [14] is a conventional partitioning technique of data clustering which easily clusters the n-object data set into k clusters known previously. A beneficial method of determining k is the silhouette. It is much more tolerant to noise and outliers as compared to the k-means because it tries to reduce a sum of pairwise differences instead of a sum of squared Euclidean distances. The most commonly known implementation of the k-medoids clustering is the Partitioning around Medoids (PAM) algorithm.

HEED. In [2], the authors have proposed the hybrid, energy-efficient, distributed (HEED) WSN clustering algorithm to prolong the lifetime of the network. It supports scalable data aggregation. In HEED, the CHs are probabilistically chosen based on the parameter called residual energy and then the sensor nodes join the network clusters according to their respective level of power. The HEED clustering can be divided into 3 main stages: (1) tentative distribution of CHs, (2) election of CH and iterative balancing of the same, and (3) finalization and establishment of membership. HEED is completely distributed. All the data has to be transmitted between the sensor nodes, or known locally [15].

LEACH-C. LEACH-C [16] is a modified LEACH using centralized clustering control, which is the same steady-state phase algorithm as LEACH [16]. LEACH-C is different from LEACH only in its setup phase. In the setup phase, in LEACH-C, every wireless sensor node would send information regarding its current location (probably determined using GPS) and its residual energy level to the BS. The BS will then elect the number of optimal CHs for the network and then the network is configured into clusters for the current round. Later, the BS would broadcast an advertisement message to all the other sensor nodes in the network; this message would contain the IDs of the various CHs and ID of all the member nodes of a cluster.

3. *k*-Means Clustering Algorithm

k-means is the simplest algorithm used for unsupervised clustering. This algorithm partitions the data set into *k* clusters using the Euclidian distance mean, resulting in maximizing intracluster similarity and minimizing intercluster similarity. *k*-means is iterative in nature [17]. It follows the following steps:

(1) Arbitrarily generate *k* points (cluster centers) [17], *k* being the number of clusters desired.

(2) Calculate the distance between each of the data points to each of the centers, and assign each point to the closest center.

(3) Calculate the new cluster center by calculating the mean value of all data points in the respective cluster.

(4) With the new centers, repeat step (2) [17]. If the assignment of cluster for the data points changes, repeat step (3); else stop the process.

The distance between the data points is calculated using Euclidean distance defined by

$$\text{Dist}\left(X_1, X_2\right) = \sqrt{\sum_{i=1}^{n}\left(x_{1i} - x_{2i}\right)^2}. \tag{1}$$

3.1. Types of k-Means Clustering in WSNs

3.1.1. Centralized k-Means Clustering [17]. When a central authority makes the various decisions and partitions the group of sensor nodes into clusters without the involvement of any other nodes, it is defined as the centralized way of clustering. In this type of clustering, the centralized authority gets the necessary information required for carrying out the clustering process from the individual sensor nodes.

Cluster Head Selection. From [17] among the sensor nodes which are at the first level of distance and the next level of distance from the centroid, we take the nodes with the highest energy and select the one which is the nearest as the CH.

Declaration of Cluster Head. Upon completion of the clustering process by the central node the central node, it sends back the information of its cluster and respective CHs to each and every node individually [17]. Thus, every node is aware of its cluster and its CH.

3.1.2. Distributed k-Means Clustering. In this type of clustering, every node gets all the necessary information for clustering from all other nodes. Since the *k*-means algorithm [17] is based on the basic principle of Euclidian distances and residual energies of the sensor nodes (for choosing CH), the information of the location of nodes and their respective residual energies is obtained by every node by exchanging messages among themselves. After collecting the information about all the nodes each node runs the algorithm (*k*-means). The *k*-means algorithm [17] for clustering and the algorithm for choosing CH are very much the same as the algorithms used in centralized clustering. As every node runs the same algorithm, every node knows its parent cluster and its CH. So here there is no process of Declaration of Cluster Head as in centralized. Thus, the distributed clustering process is complete.

Initially *k* cluster centers are randomly chosen and each of the nodes is assigned a point to the nearest center. Then the clusters are updated by finding the mean of the various member patterns, and the same steps are repeated until the algorithm converges [18]. Typical convergence criteria would include the most number of successive iterations and difference on the value of the function of distortion [18].

The detailed algorithm (in a WSN scenario) is as follows:

(1) Consider the inputs:

  (i) *k*-number of clusters, set of data points (node locations) are $\{x_1, \ldots, x_n\}$.

  (ii) Association of x_i will be done only to one cluster.

  (iii) $c(i)$ denotes ith iteration in the clustering process.

(2) Place *k* centroids in random places $\{c_1, \ldots, c_k\}$:

$$c_k = \frac{\sum_{i:C(i)=k} x_i}{N_k}, \quad k = 1, \ldots, K. \tag{2}$$

(3) Repeat until convergence:

  (a) For each point x_i,

    (i) find the nearest centroid

$$C(i) = \underset{1\le k\le K}{\operatorname{argmin}} \left\|x_i - c_k\right\|^2, \quad i = 1, \ldots, N. \tag{3}$$

    (ii) Assign the point x_i to centroid c_j which is at the least distance.

  (b) For each cluster $j = 1, \ldots, k$;

$$c_j(a) = \frac{1}{n_{j:x_i\to c_j}} \sum x_i(a) \quad \text{for } a = 1, \ldots, d, \tag{4}$$

  where a is the attribute (numerical) value of the data set.

    (i) A new centroid c_j = means of all points of x_i assigned to cluster j in previous step.

(4) Stop when the iterations converge: no node changes its cluster in successive iterations.

3.2. Network Model and Radio Energy Model

3.2.1. Basic Assumptions. In our proposed model we assume a sensor network model as given in [19] and build upon it with the following properties:

(a) The BS is a high energy node and is located far away from the extremities of the sensor network.

(b) The network is homogenous, with all sensor nodes having the same computational and transmission capabilities. In other words, all nodes are capable of acting as CH nodes.

(c) The sensor nodes have a fixed energy supply with uniform initial energy.

(d) The sensor nodes can (if required) vary the power with which they transmit signals according to the received signal strength indication of a particular node.

(e) The sensor node scans its environment at a fixed rate and will contain data to be sent to the BS at all intervals.

(f) We assume that the sensor nodes are stationary in our study because mobile nodes add extended complexity to the clustering mechanism and they involve knowledge of the geographic location of the nodes.

(g) The sensor nodes in general have location information such as a GPS support or geographic hash table information.

(h) The communication takes place over a symmetric propagation channel.

(i) The CHs perform data compression to reduce the amount of bits transmitted to the BS.

(j) The sensor nodes are capable of computational functions and the algorithm proceeds in a distributed fashion with all nodes individually collecting data.

(k) The BS indicates all nodes to reinitiate clustering when all the CH nodes in the network have insufficient energy.

3.2.2. Radio/Energy Model. In the recent history there has been lot of research in the area of ultralow power radios. If the assumptions about the radio characteristics like the energy dissipation during transmission and reception modes are changed, this will affect the advantages of different algorithms [19]. Hence, in our study, we use the simple first-order radio model that was used in the LEACH algorithm, so that the results and advantages of the new algorithm can be understood.

We assume that the radio dissipates E_{elec} amount of energy to transmit receiver circuitry and E_{amp} for the transmitter amplifier to achieve an acceptable signal to noise ratio. An inverse squared energy loss is assumed due to channel transmission. To transmit a k-bit message for a distance of d using the above-proposed model [20] the radio would expend

Table 1: Energy dissipated by various operations in radio model.

Operation	Energy dissipated
Transmitter electronics ($E_{\text{Tx-elec}}$) Receiver electronics ($E_{\text{Rx-elec}}$) ($E_{\text{Tx-elec}} = E_{\text{Rx-elec}} = E_{\text{elec}}$)	50 nJ/bit
Transmit amplifier (E_{fs}) Transmit amplifier (E_{mp})	100 pJ/bit/m^2 0.0013 pJ/bit/m^2
Data aggregation and processing (E_{DA})	5 nJ/bit/signal

$$E_{\text{Tx}} = \begin{cases} kE_{\text{elec}} + k\epsilon_{\text{fs}}d^2, & d \le d_{\text{crossover}} \\ kE_{\text{elec}} + k\epsilon_{\text{mp}}d^4, & d > d_{\text{crossover}}, \end{cases} \tag{5}$$

where $d_{\text{crossover}}$ [20] is the distance after which we switch from friss's free space propagation model to multipath fading model. This crossover distance is calculated as follows:

$$d_{\text{crossover}} = \frac{\epsilon_{\text{fs}}}{\epsilon_{\text{mp}}}. \tag{6}$$

And to receive the message it will expend

$$E_{\text{Rx}}(k) = E_{\text{Rx-elec}}(l) = kE_{\text{elec}}. \tag{7}$$

Additionally, each CH node will aggregate the data packets received from its M sensor nodes before sending a single data packet after each communication round. The energy expended in this process is given by

$$M * E_{\text{DA}} * k. \tag{8}$$

Table 1 gives the standard values of the parameters as found in [19, 21]. For our study we have adopted these parameters and proceeded with the algorithm design. Any algorithm's efficiency will be tested by its minimizations of transmit and receive operations for each message and the distance between the nodes. According to our assumptions stated above, the radio channel is considered symmetric so energy required for transmitting a message from node A to node B is the same for transmission from node B to node A. We have also assumed that the sensor nodes will always have k-bit data packets to transmit to the CHs.

4. Modified *k*-Means Algorithm

The modified k-means algorithm (Mk-means) is built upon the foundation of the k-means algorithm itself, with a major modification to ensure load balancing and extension of network lifetime. The algorithm proceeds in a series of logical steps similar to k-means and adds a final step after stable clusters have been achieved, detailed in Figure 1 schedule of cluster activity in each cycle, where the cycle indicates the removal of black listed nodes.

4.1. Setup Phase. The algorithm begins with a setup phase in which the BS randomly assigns three ($k = 3$) nodes as the initial centroids in the sensing area. The remainder

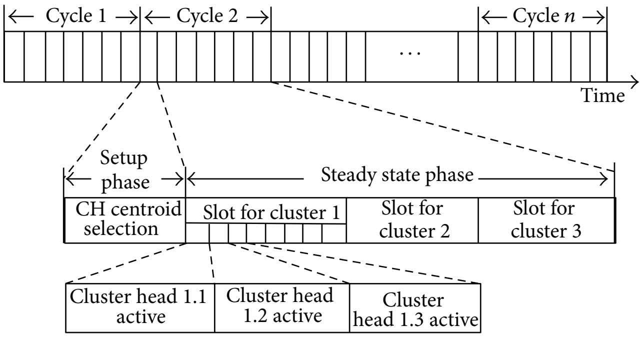

Figure 1: Schedule of cluster activity in each round.

of the clustering setup proceeds in a distributed fashion with nodes communicating with each other to establish the optimum means within each initially assigned cluster. At the end, the final CH node assigns its two nearest nodes as CHs too. Hence, in each cluster three nodes act as CHs on a load sharing basis and distribute the energy dissipation that would otherwise have occurred at a single CH node.

4.2. Network Operation Phase. After three stable clusters have been established in a distributed fashion amongst all the nodes, network operation can proceed. This is the operation area which is of actual interest, for time spent during clustering is a waste as far as data collection and sensing for the application is concerned. We have considered three variations of this network model in order to firmly establish the superiority of the *Mk*-means algorithm in various applications and energy saving parameters.

(1) A simple network: the CH node simply forwards each data packet it receives to the BS.

(2) A smart network with data aggregation: a user defined threshold is established at each sensor node, so that it only sends data selectively. The CH node transmits only one aggregated packet at the end of each round.

(3) A smart network with data aggregation and top-*k* query: the CH only transmits the user defined top-*k* query result to the BS.

The clustering exists as long as a predefined energy criterion is met by all nodes within a given cluster. If that condition is not satisfied, reclustering is initiated by the BS.

4.2.1. Simple Network. After the clustering phase stable clusters have been established, network operation can begin. When a node is established as the final CH, it broadcasts a TDMA frame to all sensor nodes in its cluster. The nodes communicate with the head on the basis of this frame, thus eliminating the multiple channel access and data collision problem. In the simple network, we assume that a node always has a data packet to send to the CH at every instant. Also, the CH node immediately forwards the data packet to the BS. An important issue faced here is the rotation of the multiple CH nodes in each cluster. Even though there are three CH nodes for every node cluster, only one node acts as a CH at a time. We have implemented a CH rotation mechanism based on load received, that is, number of transmissions. After every round of communication, the CH node is switched to one of the other nodes in the cluster and this information is broadcasted to the network. Additionally, when a CH node is not acting as a CH node, it acts as a simple sensor forwarding data to the currently active CH node. We define one round of communication as each sensor node within the network transmitting its data to its CH.

Smart Network. In this network scenario, we establish a user defined data filter threshold at each sensor node. This approach is similar to that in Filter Based Aggregation (FILA) [22]. In this approach we establish a threshold (event triggered) at each sensor node. For example, if the nodes are deployed for a fire detection application and the parameter being measured is temperature. If we establish a threshold of 100 degrees at each node, this means that the sensor node will only transmit data to its CH node if it senses a temperature value above 100 degrees. Based on user defined application specific threshold values sensed data is reported. Application of a smart network ensures that we suppress the transmission of unnecessary data to the CH nodes. By establishing the filter at the sensor node the burden on the CH node is reduced to aggregate and separate data based on the threshold.

In this mechanism, the CH still transmits all data packets that it receives directly to the BS. However, the network operation will last for a longer duration as at every instant; a sensor node may or may not send a data packet to the BS. This way, for every communication round, the CH node will receive a lower number of data packets than the simple network. Hence, it will need to expend less energy per round for data transmission and this will significantly increase the amount of time that each CH node in the network will last. This will also ensure that we break the clustering mechanism later than in the simple network, therefore leading to more stable clusters and longer network operation.

Smart Network with Top-k Query. The top-*k* query [23] is an application based query mechanism in which the user requests the top-*k* data values of one or more sensing parameters from the network in order to make a decision. In this network model, we again follow the FILA [22] based mechanism for implementation of the query. The threshold at each node still remains and we establish a data aggregation model at each CH node along with a global, user defined top-*k* query.

For data aggregation, the CH node does not send each data packet as it is received. Each CH waits for a round of communication to finish, aggregates all the data packets received by it into a single data packet, and transmits this data packet to the BS. It then passes the CH duties to the next CH node. The energy required to aggregate data packets is significantly lower than the energy required for transmission. Therefore, in this methodology, we further lower the energy consumption at each layer of the network and enhance network lifetime.

Within the aggregated data packet sent to the BS after every round, the CH computes and includes the result of the top-*k* query. These transmissions result in a database established at the BS which contains the top-*k* query result and respective node id for each cluster for each round during network operation.

For each of the three methodologies described above, there were two important considerations:

(1) When to break the network operation and recluster the nodes?

(2) How to deal with nodes that die during network operation?

For the first condition, if any one of the CH nodes in a cluster falls below this threshold, the remaining two CHs divide the load amongst them and continue network operation. The CH node which falls below the threshold continues to act as a sensor node and sends data to the remaining CH nodes according to the TDMA frame allocated. The threshold defined for a node to act as a CH is 70% of its initial battery energy at the time of becoming CH. The network operation breaks and reclustering is initiated when all three CH nodes in any of the clusters are incapable of acting as such and the last node sends a message to the BS which initiates the clustering process as described in the setup phase.

Another important feature of the *Mk*-means algorithm is an inherent 3-2-1 cluster mechanism. During the multiple CH selection, there is no restriction on the CH node to choose a node from within its own cluster to act as the additional CH. While this will not make much difference during the normal network operation, it will be useful during the final stages; when a majority of the nodes are unavailable to act as CH nodes, only one large cluster will be formed with each of the multiple CH nodes spread across the network. Therefore, it will be similar to the *k*-means operation in which three clusters are formed with a single CH. In this topology, however, the network operation will only be suspended when all three nodes are unable to act as CH nodes due to a fall in battery energy below the prespecified threshold.

4.3. Postclustering Phase. This phase very rarely occurs in the *Mk*-means algorithm because of the high stability of the clusters and hence longer network operation. In this stage, no node in the network has energy sufficient to act as a CH node. The criterion taken for this is that if 65% of the nodes alive are blacklisted, the network will move to this phase.

From the battery power discharge profile [24, 25], it is found that the stability of the battery to support the operations after losing its 70% of initial energy weakens and it starts to discharge at rapid phase. Hence, the clustering process is stopped and direct transmission is initiated so that the nodes will not waste their energy in forming new clusters instead they can send useful data to BS.

To simulate a realistic environmental scenario, we have not stopped the simulation of the network when none of the nodes in it are unfit for becoming CH, owing to a paucity of battery energy. In this phase, all the remaining alive nodes in the network start sending their data directly to the BS. These highly expensive energy transmissions continue till 80% of the network nodes die, at which point, the network becomes useless in terms of the useful data it can provide (see Algorithm 1).

Table 2: Simulation parameters.

Parameter	Value
Simulation area	100 m × 100 m
Initial energy per node (E_{init})	2 J
Nodes	100
Data size	2000 bits
Base station (BS) location	(50 m, 175 m)
E_{Elec}	50 nJ/bit
Transmission amplifier type, E_{fs}	10 pJ/bit/.m^2
Transmission amplifier type, E_{mp}	0.0013 pJ/bit/m^2
Data aggregation energy	5 nJ/bit/signal
Energy threshold (ethres)	$0.7 * E_{\text{init}}$
Break parameter	65% of nodes blacklisted

5. Simulations and Analysis of Results

5.1. Simulation Environment. All the experiments are conducted for the proposed *Mk*-means algorithm versus *k*-means clustering, LEACH, LEACH-C, HEED, SECA, ECRA, PAGASIS, and NCACM using MATLAB version R2012b and C programming language [19, 26–29]. Each sensor node was assumed to have initial energy of 2 J. In the simulation run, we used the following parameter values same as in [26], as shown in Table 2.

5.2. Simulation Metrics and Analysis. In order to suitably compare the performance of the *Mk*-means algorithm with the *k*-means algorithm, we have taken a number of metrics in common with the work of [21, 26]. There are two important parameters in the simulation. One is the energy threshold, and the other is the break parameter. The energy threshold is defined at each CH. If the energy of the node falls below the threshold value, it is no longer fit to act as a CH. This parameter is fixed at 70% of the initial battery energy, or 1.4 J through empirical observations. The break parameter is defined as the number of blacklisted nodes that cause the network to break from the network operation phase and move to the postclustering phase. Through empirical observations, this parameter was fixed at 65% of nodes. When these many nodes are blacklisted, the network breaks to the postclustering phase where all nodes start sending data directly to the BS.

All the results discussed below are for *Mk*-means and *k*-means algorithms with a smart network and top-*k* based data aggregation model. The results have been shown for 10 runs of the code.

5.2.1. Increase in Network Lifetime and Effective Data Sent. The effective lifetime of a sensor network is an important metric to judge the usefulness and performance of the network. An inherent problem with MATLAB v R2012b simulation is that an accurate model for time cannot be integrated into the simulation scenario to objectively measure the network lifetime in hours or days. Therefore, we have used the total number of transmissions throughout the network operation

```
For Smart Network with Top-$k$:
Clustering Phase:
    Function 1
        for $i = 1 : \text{nodes}$
        for $j = 1 : n$ (CH nodes)
    // data generation at each node
    // threshold established globally
    // establishes a TDMA frame
    Function 2:
    // transmission from each sensor node to its respective CH after threshold check
    // CH data aggregation after each round, calculation of top-$k$ result and transmission to BS
For dead nodes:
    Function 3:
        // checks each nodes energy against 0 to establish node dying
        // removes dead nodes from all lists, including blacklist
    // CH rotation mechanism after each round of communication
    // collect energy spend for transmission and reception
    // check the Cthreshold of rotated CH to satisfy the criteria
    // if fails; current CH calls to next CH as per schedule and informs BS
    // check atleast one CH in each cluster at BS; if this criteria fails call Function 1
    //call Function 2 again
For post clustering phase:
    Function 4:
        // checks each nodes energy against 0 to establish node dying after each round
        // removes dead nodes from all lists, including blacklist
    Function 5:
        for $i = 1 : n$ (alive nodes)
        // all nodes send data directly to BS
        // energy deductions due to transmission and reception
```

Algorithm 1

as an indicator of the overall lifetime of the network. This metric is not indicative of the number of hours or days the network might exist, but it is an indirect way of gauging how much longer the *Mk*-means algorithm allows the network to function over the *k*-means algorithm. The number of transmissions is used interchangeably as time over the course of this discussion.

From Figure 2, we observe that the average increase of the number of rounds of communication over 10 runs of code is 33 times. This metric is indicative of the substantial increase in the number of times each sensor node in the network is allowed to send data before reclustering is initiated.

Additionally, the total number of data transmissions that took place in the network has also been measured to conclusively establish that *Mk*-means results in clusters which are stable and remain in operation for longer periods of time, with nodes in each cluster being able to send a lot of data before clustering is reinitiated. This metric is a direct measure of the number of data packets that were sent to the BS during the operation of the network. It represents the increase in useful data traffic, which is the most essential part of a network's lifetime.

From Figure 3, we can observe that over 10 runs of the code, the *Mk*-means algorithm outperforms the *k*-means algorithm significantly. At worst, a 61% increase was observed and 188% as the maximum increase in traffic supported. On average, the overall increase in network transmissions was found to be 136%. This metric shows that *Mk*-means significantly increases the lifetime of the sensor network as compared to *k*-means.

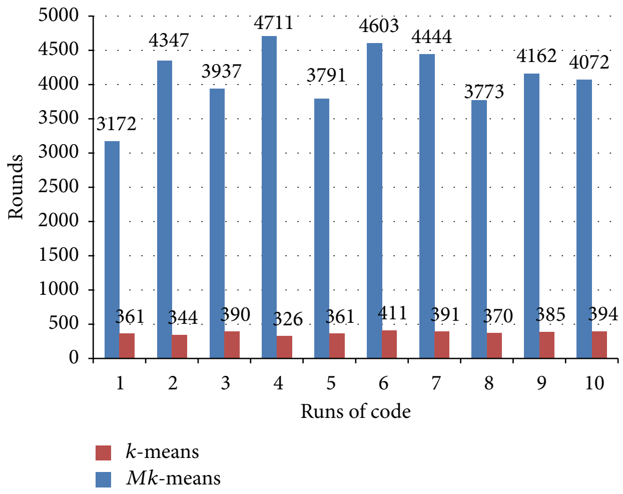

Figure 2: Rounds of communication in *Mk*-means versus *k*-means.

Besides this, there are several other metrics which were gauged in order to determine the extension of the network lifetime through *Mk*-means. For instance, we have measured the number of rounds of communication that occurred

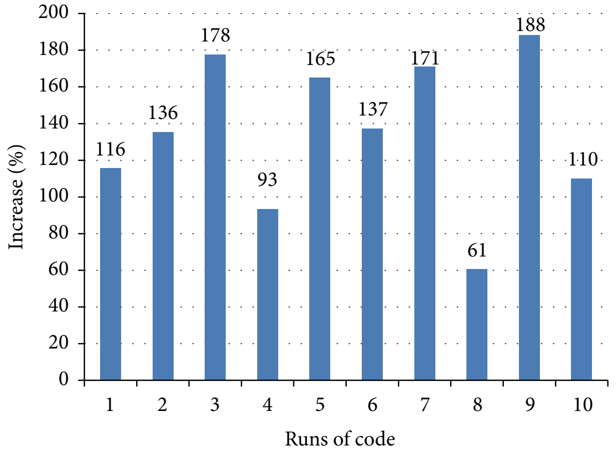

Figure 3: Total increase in transmissions on *Mk*-means versus *k*-means.

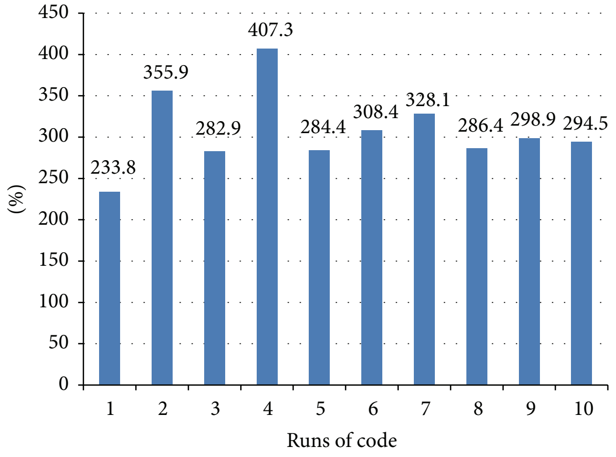

Figure 4: Percentage increase in data transmissions.

during the network operation phase. This is a standard metric and can be used to directly establish that the *Mk*-means algorithm results as shown in Figure 4, on much more stable clusters than the *k*-means algorithm, which last much longer and hence allow more number of useful data transmissions during the network operation phase. We define a round of communication as when all of the nodes in the network have had an opportunity to send data to their respective CH node.

The final parameter used is the number of times clustering is done in both *Mk*-means and *k*-means. This parameter establishes that *Mk*-means spends much less time during the clustering process than *k*-means does. This is a direct consequence of stable clusters being formed in *Mk*-means as compared to *k*-means. Clustering is generally the most expensive process in the network as a huge number of transmissions have to be made to and from each node. The *Mk*-means algorithm minimizes the number of clustering processes that have to be undertaken during network operation.

On average, the *Mk*-means algorithm undertakes the clustering process 17 times, while *k*-means does it 55 times as shown in Figure 5. This represents a significant decrease in the amount number of transmissions that the network has to make during clustering. As established before, clustering is an expensive process and the decrease is a significant one, thus further conserving energy at the nodes and further extending network lifetime.

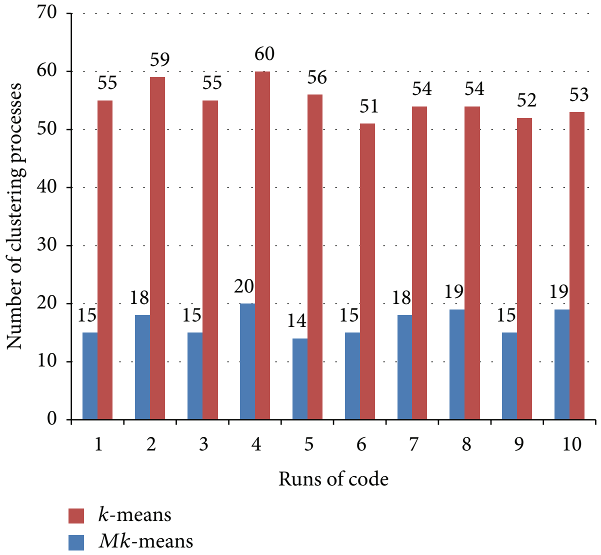

Figure 5: Total number of clustering processes in *Mk*-means versus *k*-means.

5.2.2. Number of Nodes Alive. The number of nodes alive is a well-established standard metric to measure the lifetime of a sensor network. An important parameter in this metric is the FND or First Node Dead metric, which measures the rounds of communication till the death of the first node in the network.

From Figure 6, the average number of rounds of communication after which the first node dies in *Mk*-means is 1441 and the average for *k*-means is 373. The introduction of a filter at the node level to suppresses unnecessary data transmissions, the rotation of multiple CHs in a cluster by load, and the application of a data aggregation model greatly prolong the FND criterion in *Mk*-means as compared to *k*-means with the same parameters. This is again due to the stability of the *Mk*-means clusters as compared to the *k*-means clusters.

5.2.3. Average Energy of the Network. The average energy dissipation here is plotted in pairs using the following methodology. The energy graph consists of pairs, each pair representing one clustering process and one successive network run till the clustering process breaks. The first bar of each pair represents the energy at the start of that clustering process. The next bar represents the energy at the start of the network operation phase. Thus, the fall between the bars represents the amount of energy dissipated during clustering. The fall between successive pairs represents the amount of energy that was dissipated in the previous network run. The sharper this fall, the more energy is dissipated during the network, indicating that the network ran for a significant amount of time. Having

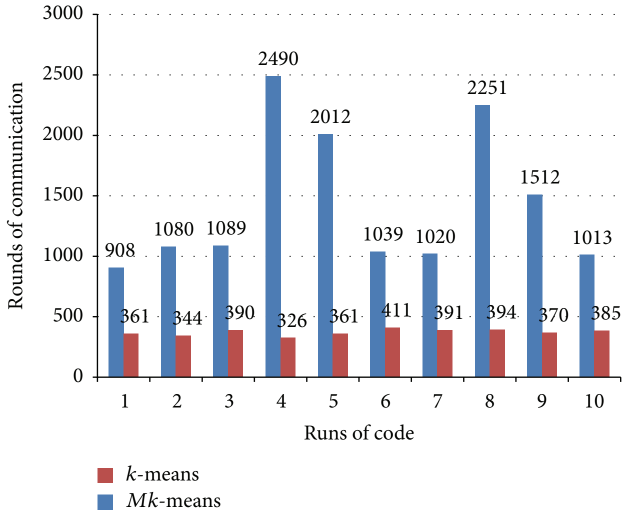

Figure 6: First Node Dead.

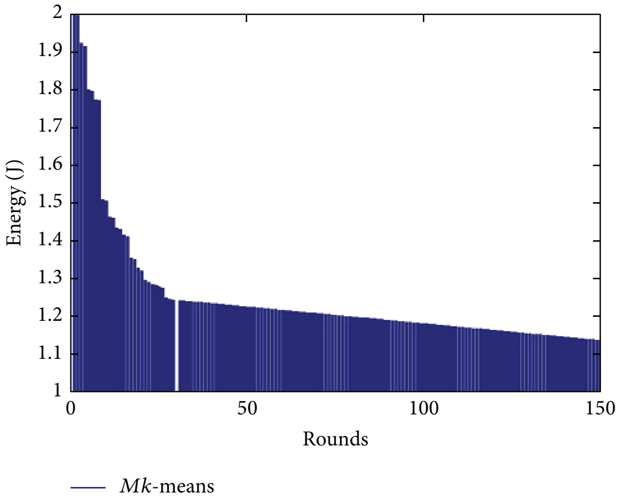

Figure 7: Average energy dissipation in *Mk*-means.

already established that *Mk*-means forms a fewer number of highly stable clusters with a significant increase in meaningful data transmission, we therefore establish that this fall between pairs is further indication of the longevity of each network operation phase.

Once the postclustering phase is initiated, the energy is calculated after every round of communication.

As observed from Figure 7, the fall between pairs is very sharp, indicating the longevity of the network. The break between the network operation phase and the postclustering phase is clearly visible on the graph.

We can contrast this energy graph with a similar plot for *k*-means as shown in Figure 8. There are significantly more pairs visible in this graph owing to the greater number of clustering processes that occur in *k*-means as compared to *Mk*-means. Further, the fall between the pairs is hardly visible which is indicative of the fact that the *k*-means clusters are unstable as compared to *Mk*-means and decay much faster, forcing the network to initiate the reclustering process more often.

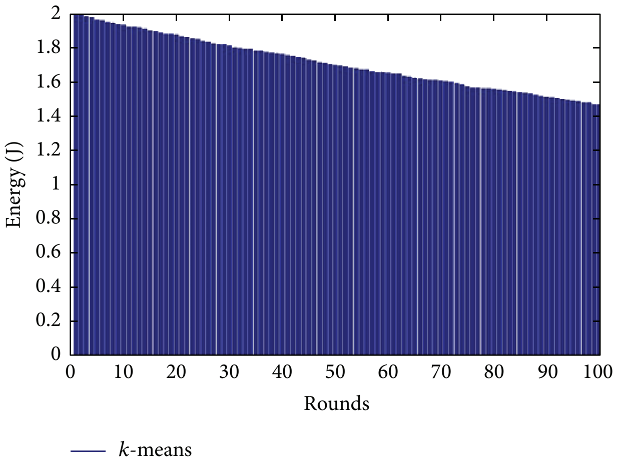

Figure 8: Average energy dissipation in *k*-means.

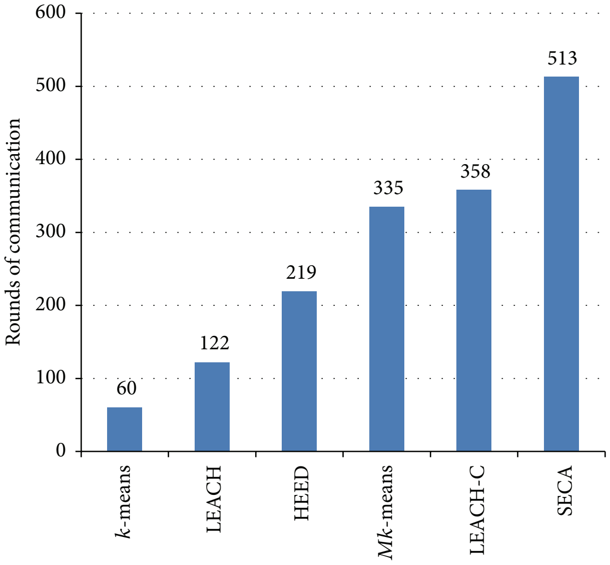

Figure 9: First Node Dead comparison for validation.

5.2.4. Validation and Comparison. In order to verify the simulation results discussed above, the base *k*-means algorithm was cross referenced against standards as in [26, 30, 31]. Our proposed *Mk*-means algorithm is built upon the same methodology as the *k*-means algorithm. Comparing the *Mk*-means and the *k*-means codes, the results for the FND (First Node Dead) metric are against the results cited in [26]. As expected, the *k*-means code performs poorly due to the high degree of energy dissipation and the lack of any sensor filter technique. This is due to its inability to form stable CHs and the small number of sensor nodes. The *Mk*-means due to its inherently stable clustering phenomenon performs well just by outstripping both LEACH and HEED and *k*-means algorithm alone, as shown in Figure 9.

But in later case the *Mk*-means algorithm with top-*k* query dramatically outperforms the existing algorithms owing to its unique multiple CH methodology as shown in Figures 10 and 11. The rotation of CHs based on load

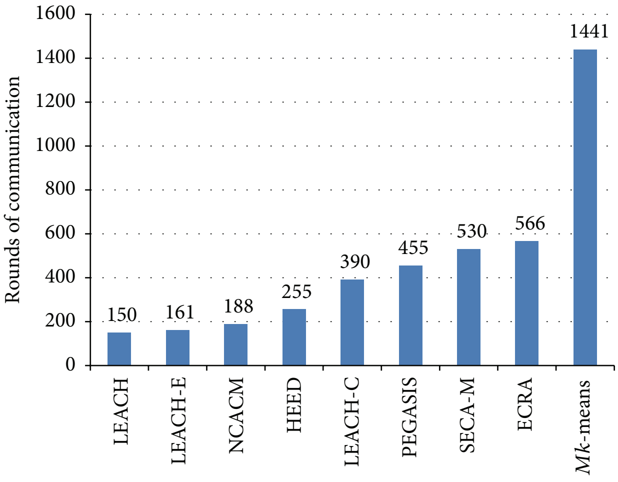

Figure 10: First Node Dead comparison.

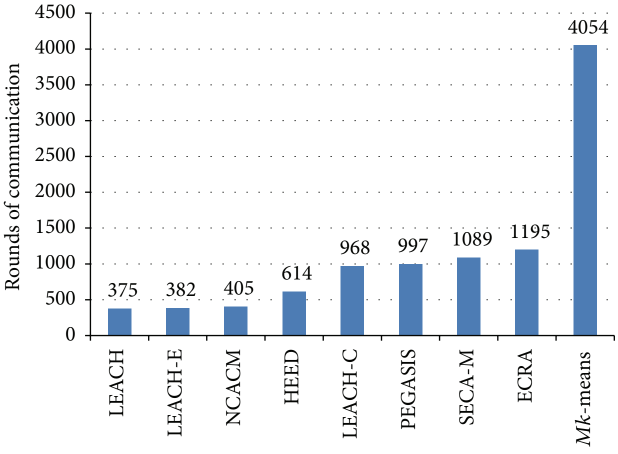

Figure 11: Half of the nodes alive.

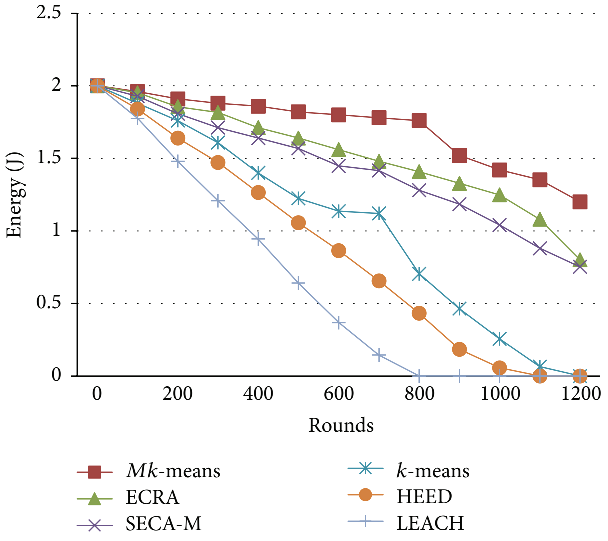

Figure 12: Total network energy.

and the filter based threshold attached to each sensor node ensures that the energy dissipation in the network is minimal. Additionally, clustering processes are minimized which further reduces the energy dissipation. The data aggregation model reduces the number of transmissions that a CH node has to make, thereby reducing the number of times a very expensive energy consuming process needs to be performed. And it shows that the about 300% of rise in sensor node life time is due to the factor that 3 CHs are elected per cluster which scales down the energy dissipation by one-third in the process of reclustering and implementation of query based data forwarding.

The performance comparison is done with 100 sensor nodes in a sensing area about 100 m × 100 m for 1200 rounds of communication. A round is considered to be the active period of one CH in the cluster. The results were plotted in Figure 12, residual energy of the network against round. From the simulations results it is found that our proposed scheme *Mk*-means with top-*k* query takes the advantages of higher residual energy throughout the simulation with stable clusters and reduced energy consumption than that of ECRA, SECA-M, *k*-means, HEED, and LEACH.

6. Conclusion

We analyzed the performance of the proposed *Mk*-means algorithm versus the *k*-means and other standard algorithms. In order to suitably compare the performance of the *Mk*-means algorithm with the *k*-means algorithm, a number of metrics were measured. *Mk*-means significantly increased the lifetime of the sensor network as compared to *k*-means. The number of times each sensor node in the network was allowed to send data before reclustering is initiated was found to be more in *Mk*-means than in *k*-means. The *Mk*-means algorithm minimizes the number of clustering processes that have to be undertaken during network operation. *Mk*-means was found to significantly decrease the amount number of transmissions that the network had to make during clustering. To conclude, the *Mk*-means algorithm was found to outperform the existing clustering algorithms owing to its unique multiple CH methodology. The rotation of CHs based on load and the filter based threshold attached to each sensor node was found to ensure that the energy dissipation in the network is minimal. Additionally, clustering processes were found to be minimized which further reduced the energy dissipation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[2] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[3] S. Yi, J. Heo, Y. Cho, and J. Hong, "PEACH: power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2842–2852, 2007.

[4] W.-T. Su, K.-M. Chang, and Y.-H. Kuo, "eHIP: an energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 1151–1168, 2007.

[5] V. Mhatre and C. Rosenberg, "Design guidelines for wireless sensor networks: communication, clustering and aggregation," *Ad Hoc Networks*, vol. 2, no. 1, pp. 45–63, 2004.

[6] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on Communications*, vol. 29, no. 11, pp. 1694–1701, 1981.

[7] D. J. Baker, A. Ephremides, and J. A. Flynn, "The design and simulation of a mobile radio network with distributed control," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 1, pp. 226–237, 2003.

[8] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[9] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[10] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[11] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1713–1723, IEEE, April 2003.

[12] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the 4th International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN '99)*, pp. 310–315, IEEE, Perth, Australia, June 1999.

[13] L. Kaufman and P. J. Rousseeuw, "Clustering by means of Medoids," in *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pp. 405–416, North-Holland Publishing Company, 1987.

[14] S. Shah and M. Singh, "Comparison of a time efficient modified K-mean algorithm with K-mean and K-medoid algorithm," in *Proceedings of the International Conference on Communication Systems and Network Technologies (CSNT '12)*, pp. 435–437, Rajkot, India, May 2012.

[15] R. E. Blace, M. Eltoweissy, and W. Abd-Almageed, "Threat aware clustering in wireless sensor networks," in *Proceedings of the IFIP Conference on Wireless Sensor and Actor Networks (WSAN '08)*, pp. 1–12, Ottawa, Canada, July 2008.

[16] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S8–S13, 2005.

[17] P. Sasikumar and S. Khara, "K-means clustering in wireless sensor networks," in *Proceedings of the IEEE 4th International Conference on Computational Intelligence and Communication Networks (CICN '12)*, pp. 140–144, Mathura, India, November 2012.

[18] X. Jin, S. Kim, J. Han, L. Cao, and Z. Yin, "GAD: general activity detection for fast clustering on large data," in *Proceedings of the SIAM International Conference on Data Mining (SDM '09)*, pp. 2–13, Sparks, Nev, USA, April 2009.

[19] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS '00)*, IEEE, Maui, Hawaii, USA, January 2000.

[20] W. N. W. Muhamad, K. Dimyati, R. Mohamad et al., "Evaluation of Stable Cluster Head Election (SCHE) routing protocol for wireless sensor networks," in *Proceedings of the IEEE International RF and Microwave Conference (RFM '08)*, pp. 101–105, IEEE, Kuala Lumpur, Malaysia, December 2008.

[21] W. Xiaoping, L. Hong, and L. Gang, "An improved routing algorithm based on LEACH protocol," in *Proceedings of the 9th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES '10)*, pp. 259–262, IEEE, Hong Kong, August 2010.

[22] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, *TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks*, UC Berkeley and Intel Research, Berkeley, Calif, USA, 2002.

[23] H. Haiping, F. Juan, W. Ruchuan, and Q. XiaoLin, "An exact top-*k* query algorithm with privacy protection in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 749049, 10 pages, 2014.

[24] A. Silva, M. Liu, and M. Moghaddam, "Power-management techniques for wireless sensor networks and similar low-power communication devices based on nonrechargeable batteries," *Journal of Computer Networks and Communications*, vol. 2012, Article ID 757291, 10 pages, 2012.

[25] S. Barcellona, M. Brenna, F. Foiadelli, M. Longo, and L. Piegari, "Analysis of ageing effect on Li-polymer batteries," *The Scientific World Journal*, vol. 2015, Article ID 979321, 8 pages, 2015.

[26] J.-Y. Chang and P.-H. Ju, "An energy-saving routing architecture with a uniform clustering algorithm for wireless body sensor networks," *Future Generation Computer Systems*, vol. 35, pp. 128–140, 2014.

[27] P. Kuila and P. K. Jana, "A novel differential evolution based clustering algorithm for wireless sensor networks," *Applied Soft Computing Journal*, vol. 25, pp. 414–425, 2014.

[28] P. Kuila and P. K. Jana, "Energy efficient clustering and routing algorithms for wireless sensor networks: particle swarm optimization approach," *Engineering Applications of Artificial Intelligence*, vol. 33, pp. 127–140, 2014.

[29] T. K. Jain, D. S. Saini, and S. V. Bhooshan, "Cluster head selection in a homogeneous wireless sensor network ensuring full connectivity with minimum isolated nodes," *Journal of Sensors*, vol. 2014, Article ID 724219, 8 pages, 2014.

[30] J. R. Patole and J. Abraham, "Design of MAP-REDUCE and K-MEANS based network clustering protocol for sensor networks," in *Proceedings of the 3rd International Conference on Computing, Communication and Networking Technologies (ICCCNT '12)*, pp. 1–5, IEEE, Coimbatore, India, July 2012.

[31] J.-Y. Chang and P.-H. Ju, "An efficient cluster-based power saving scheme for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, article 172, 2012.