

Prediction-based Clustering Algorithm for Mobile Wireless Sensor Networks

Andrey Koucheryavy

Ahmed Salim

Advisor of General Director
Central Science Research Telecommunication Institute
Moscow , Russia
Kouch@Komset.ru

PhD Student
Telecommunications University
Saint-Petersburg, Russia
Salimxpress@hotmail.com

Abstract

In recent years, a new paradigm of sensor networks, mobile sensor networks (MSNs), are emerging. They inherit all the properties from static wireless sensor networks (WSNs) and meanwhile have their own uniqueness, node mobility. Cluster-based architecture in MSNs have the advantage of providing scalable and resource efficient solutions. Thus, finding an optimal way to generate clusters is an important topic in MSNs. In this paper, we present a prediction-based clustering algorithm for MSNs. Our proposed algorithm uses adequate criterion for cluster-head election in conjunction with heuristic predictors to generate steady and balanced clusters. The results of the simulation experiments clearly show the benefit of the proposed algorithm in terms of network life time.

1 Introduction

A Wireless Sensor Network (WSN) consists of a large set of wireless sensors deployed over a geographic area known as the sensors field, to monitor specific characteristics of the environment or characteristics of specific subjects immerse in that environment. Wireless sensors are small devices with limited resources: limited battery power, low memory, little computing capability, very low data rates, low bandwidth processing, variable link quality, etc. However, despite their constraints, when the sensors are deployed in large numbers they can provide us with a very real picture of the field being sensed. Wireless sensors collect the data related to that phenomena and forward it to a central unit for processing and analysis. Examples of the data that can be collected include temperature, humidity, light conditions, seismic activities, etc. This data can be used to detect certain events and to trigger activities. For example, sensors distributed over a large woodland could automatically raise an alarm if a fire has broken out somewhere [15]. When creating a WSN, in most cases the sensors are deployed randomly in the sensor field. Once deployed, the sensors communicate wirelessly

among each other, and send the collected data towards a central point called the base station or sink. This central data collection point allows the processing of the collected data and the connection of the WSN with other networks, either WSN or other types of networks like a wired corporate network or the Internet.

The use of the WSN potential provides efficient and costs effective solutions for several problems. However, it is necessary to implement mechanisms or procedures to deal with the sensors' constraints. Due to the limitation of their physical size, the sensors tend to have storage space, energy supply and communication bandwidth so limited that every possible means of reducing the usage of these resources is aggressively sought.

The use of clustering techniques has been proposed to help solve some of those constraints, by allowing the organization of the sensors in a hierarchical manner, grouping them into clusters and assigning specific tasks to the sensors in the clusters, before forwarding the information to higher levels. A cluster in WSN means to interconnect the sensor nodes hierarchically in order to improve their performance. Clusters provide reliability by having sensors taking over the workload of failing sensors. Clusters also improve data aggregation mechanisms, reduce the workload of each sensor to save energy and thus increase the overall lifetime of the system. The creation of clusters in a WSN field is generally done taking into account the proximity between the sensors, measured through the radio frequency signal they emit. Clustering helps solve some of the sensors' constraints by: reducing the cost of transmitting data to base stations, reducing the power consumption in the devices, facilitating the gathering of sensed data, maximizing the routing process execution and allowing scalability, i.e., the concept of clustering is very useful in different contexts of WSN.

Mobility in wireless sensor networks has attracted a lot of attention in the recent years and has introduced unique challenges in aspects like resource management, coverage, routing protocols, security, etc. Thus, the next evolutionary

step for sensor networks is to handle the mobility effect in all its forms. Most of clustering protocols currently used or proposed for WSN assume that the nodes are stationary [12]. However, in applications like habitat monitoring (animal tracking) or search and rescue, that assumption makes those clustering mechanisms invalid, since the static nature of sensors is not real. One of the reasons for the sensors to be taken as stationary is because that assumption facilitates the simplification of the clustering protocols, making them have a very low overhead. It also avoids having to manage the mobility patterns of the sensors and allows saving more energy, since the localization information that the network has to manage is very low.

In this paper, we propose a Distributed Clustering Algorithm (DCA) which is based on the use of predicted combined criteria metric. In DCA algorithm, the sensor calculates its combined criteria metric according to connectivity, coverage, mobility and residual energy, after that each sensor node uses its history combined criteria information to predict the current combined criteria based on heuristic predictors. The sensor with the highest Predicted combined criteria in its r – hop neighborhood will become the cluster-head.

The rest of this paper is organized as follows. In section 2, the related work is discussed. The criteria and the heuristic predictors are introduced in Section 3. Section 4 exhibits the details of our approach. In Section 5, we evaluate the performance of our approach via simulations and compare the results with LEACH-M. Finally, Section 6 concludes the paper.

2 Related Research

In this section we discuss the related work that has been done in these areas.

Mobile sensor networks (MSNs) are a new paradigm of WSNs that emerged a few years ago. In MSNs, nodes are attached to certain mobile platforms and therefore obtain the ability to geographically relocate themselves as needed. Note that, unlike another mobility-enriched type of wireless networks, mobile ad hoc [5, 6] networks (MANETs), where node mobility is a uncontrolled interferential factor, MSNs have it as a controllable asset [9, 10], by which they can be tuned more intelligent and more flexible and adaptive to their monitored environments than static WSNs. Kansal et al [2] implement controlled node mobility to provide a sustainable WSN, reviewing the potential advantages. However, the approach assumes mobility is restricted to predefined cableways, and would not be suitable for the random deployment applications. Small et al [7] consider a similar approach by analyzing the WSN performance issues from the development of an infestation model, where sensor

nodes would use cableways to produce three dimensional movements, with a view of optimizing coverage.

The use of clusters for transmitting data to the base station leverages the advantages of small transmit distances for most nodes, and having a smaller number of nodes (Cluster Head) transmitting data to the base station [13]. During the establishment of the cluster, it is necessary to take into account aspects like: cluster size and form, how to select the cluster head, how to control inter-cluster and intra-cluster collisions, and energy saving issues. The design of the clustering process [14] is one of the more important issues for the correct functioning of the WSN, due to the probed efficiency of using a hierarchical scheme for communications between the network elements. Many clustering algorithms and protocols have been proposed for WSN. A clustering algorithm splits the network into disjoint sets of nodes each centering on a chosen cluster-head. A good clustering imposes a regular, high-level structure on the network [3, 4]. In general, a clustering algorithm should partition the network so that the nodes inside each cluster have high correlation. This correlation can be spatial, temporal or based on the data collected by the sensor.

Probably, the first and the better known cluster-based routing protocol proposed was the Low Energy Adaptive Clustering Hierarchy (LEACH) [8]. LEACH is a self-organizing and adaptive clustering protocol. The energy cost of data packets transmitting is distributed among all sensor nodes in a random way. In LEACH, the nodes organize themselves into local clusters, with one node serving as a local base station or cluster head. In this cluster scheme, if the cluster heads are chosen a priori and fixed throughout the system lifetime, it is easy to see that the unlucky sensor nodes chosen to be cluster heads would die quickly. Thus, LEACH includes randomized rotation of high-energy cluster head nodes such that they rotate among the various sensor nodes in order not to drain the battery of a fixed group of nodes. In addition, LEACH performs local data fusing to compress the amount of data being sent from cluster heads to the base station, further reducing energy dissipation and prolong the system lifetime. The hierarchical topology of the network is similar to our approach. However, all the sensors including the CH are static in LEACH.

LEACH-Mobile, in short LEACH-M [11] is a variation of LEACH, which support node mobility. In LEACH-M mobile nodes declare the membership of a cluster as they move, and to confirm whether a mobile sensor node is able to communicate with a specific cluster head within a time slot allocated in TDMA schedule. This ensures dynamic joining and leaving of non-cluster mobile nodes in the steady state phase, unlike [8] where cluster membership is fixed after cluster formation. However, in LEACH-M, clusters are dynamically formed every time the sensor moves, giving rise to chance of overhead in the cluster

maintenance.

The main difference between our work and LEACH-M is that our work provides remarkable extending network lifetime and provides stable cluster formation decisions based on predicted combined criterion metric

3. Predicted combined criterion metric

In this section, we will start by giving our criteria, which are used to determine the combined criterion metric, then we introduce three heuristic predictors which are used to predict the combined criterion metric. The details of connectivity criterion are presented as follows.

3.1 Connectivity Criterion

Connectivity gives a measure of the communication capabilities; it is termed as the ability of any node to communicate with any other node in the network.

Consider a Mobile wireless sensor network (MSN) comprised of a large number of sensors, denoted by $S = \{s_1, s_2, \dots, s_n\}$. The sensors are distributed over a large target field and they are designed to detect specified events. The communication region of sensor s defines the area in which the sensor s can communicate directly with other sensor nodes. The maximum distance between s_i and any other node s_j , where s_j is in communication range of s_i , is called the communication range (R_i) of sensor s_i . The set of communication neighbors of node s_i is represented by $N(s_i)$ and defined by the following equation:

$$N(s_i) = \{s_j : d(s_i, s_j) \leq R_i\}, \quad (1)$$

where d is the Euclidean distance between s_i and s_j . Then we can get the connectivity criterion $ConC(s_i)$ at time t by the following equation:

$$ConC(s_i) = N(s_i). \quad (2)$$

3.2 Coverage Criterion

Most of cluster head selection approaches are designed with the goal to provide balanced energy consumption among sensor nodes, but at the same time, these approaches do not consider the network's requirement for full coverage over expended periods of time [16].

The *sensing range* (r) of a sensor s indicates the maximum distance between sensor s and any point p in the sensing region of sensor s . A point p is covered or monitored by a sensor node s if the Euclidean distance between p and s is less than the sensing range of sensor s .

Definition 1 (A Segment): The segment Seg_{ij} is the section of s_i 's sensing perimeter that is covered by the sensing range of node s_j . The segment Seg_{ij} is represented in terms

angels identified by closed interval $[\theta_{ij}^s, \theta_{ij}^e]$, where θ^s, θ^e are the start and end angle of the segment Seg_{ij} start and end point.

Consider two nodes s_i and s_j with distance d from each other. Then, by using the Cosine Rule, the length L_{ij} of the segment Seg_{ij} is

$$L_{ij} = \arccos\left\{\frac{r_i^2 + d^2 - r_j^2}{2d * r_i}\right\} \quad (3)$$

The start and end angles can be calculated by, $\theta_{ij}^s = \theta_{ij} - L_{ij}/2$ and $\theta_{ij}^e = \theta_{ij} + L_{ij}/2$ where θ_{ij} is angular coordinate of node s_j that denotes the direction of s_j relative to arbitrary polar-axis.

Definition 2 (A Segment List): A segment list of node s_i , denoted by SL_i , is a set of segments such that, (a) A segment in SL_i is not subsumed in any other segment in SL_i . (b) Every segment in SL_i overlaps with exactly two other segments one from each side, except two segments (the first and the last segments) that each one of them overlap only with a single segment in SL_i .

Then we can get the coverage criterion $CovC(s_i)$ at time t by the following equation:

$$CovC(s_i) = \frac{Cov(s_i)}{2\pi} \quad (4)$$

where $Cov(s_i)$ is the perimeter coverage of node s_i and given by the following equation:

$$Cov(s_i) = \sum_{\forall Seg_{ij} \in SL_i} |Seg_{ij}| - \sum_{\forall Seg_{ij}, Seg_{ik} \in SL_i: j \neq k} |Seg_{ij} \cap Seg_{ik}| \quad (5)$$

3.3 Mobility Criterion

Mobility is an important factor in deciding the cluster-heads. In order to avoid frequent cluster-head changes, it is desirable to elect a cluster-head that does not move very quickly. When the cluster-head moves fast, the nodes may be detached from the cluster-head and as a result, a reaffiliation occurs. Reaffiliation takes place when one of the ordinary nodes moves out of a cluster and joins another existing cluster. In this case, the amount of information exchange between the node and the corresponding cluster-head is local and relatively small [17].

Compute the mobility speed of every node s_i by the following formula:

$$MC(s_i) = \frac{\sqrt{(x_t - x_{t-1})^2 + y_t - y_{t-1})^2}}{\Delta t}, \quad (6)$$

where (x_t, y_t) and (x_{t-1}, y_{t-1}) are the coordinate positions of node s_i at time t and $t - 1$ and Δt is the interval time between t and $t - 1$.

3.4 Residual Energy Criterion

Cluster-heads are responsible for coordinating among the cluster members, aggregating their collected data, and transmitting the aggregated data to the remote sink, directly or via multi-hop transmission mode. Since cluster-heads need to receive many packets and consume a lot of power for long range transmission, they are the ones whose energy is used up most rapidly in the cluster if they are elected for a long time. Therefore, a cluster-based scheme should avoid a fixed cluster-head election scheme, because this latter with constrained energy may rapidly drain its battery power due to its heavy utilization. That can cause bottleneck failures in its cluster, and trigger the cluster-head election process again. For that, we proposed in the DCA that the cluster-head election process would be carried out periodically after each round to evenly balance energy dissipation among the sensors during the network lifetime.

Then compute the Residual Energy $REC(s_i)$ criterion of every node s_i at any time t by the following formula:

$$REC(s_i) = E(s_i), \quad (7)$$

Then from equation 6, 4, 6 and 7 we can calculate the Combined Criterion $CC(s_i)$ for node s_i at any time t by the following formula:

$$CC(s_i) = \alpha \times ConC(s_i) + \beta \times CovC(s_i) + \frac{\gamma}{1 + MC(s_i)} + \zeta \times REC(s_i), \quad (8)$$

where $\alpha + \beta + \gamma + \zeta = 1$

3.5 Predictors

In this section, we outline three heuristic predictors which are used in our experiments. Every one of them has special properties that we mentioned earlier. The Combined Criterion CC will be predicted at a current time t_c according to historical combined criterion $HCC = [(CC_1, t_1), (CC_2, t_2), \dots, (CC_n, t_n)]$ where $t_1 < t_2 < \dots < t_n$ as follows:

Single Point Predictor (SPP): This predictor always returns the last value of HCC , as shown in the following equation:

$$PCC = SPP(t_c) = CC_n \quad (9)$$

Linear Extrapolation Predictor (LEP): LEP means creating a tangent line at the end of HCC and extending it beyond that limit. LEP will only provide good results when

used to extend the graph of an approximately linear function or not too far beyond the known data []. Equation 10 shows the formula to obtain Predicated Combined Criterion PCC .

$$PCC = LEP(t_c) = \frac{t_c - t_{n-1}}{t_n - t_{n-1}} (CC_n - CC_{n-1}) + CC_{n-1} \quad (10)$$

Hybrid Predictor (HP) This prediction model is a mixture of SPP and LEP. This model is mostly useful when most parts of data is linearly predictable but rapid changes occur in some parts of sensed values (jumps). Equation 11 shows the formula to obtain Predicated Combined Criterion PCC

$$PCC = \begin{cases} CC_n, & n \text{ is odd;} \\ \frac{t_c - t_{n-1}}{t_n - t_{n-1}} (CC_n - CC_{n-1}) + CC_{n-1}, & n \text{ is even.} \end{cases} \quad (11)$$

4 Distributed Clustering Algorithm (DCA)

In this section, we present our proposed algorithm that enables to generate steady and balanced clusters. To carry out our algorithm, we assume that;

- All sensors are homogeneous with constrained energy and the same transmission range.
- The network topology changes, and sensors move with a speed ranging between 0 and 1 (m/s)
- The sensors make all decisions without reference to a centralized controller.

We introduce a parameter termed as cluster-hop, r_{hop} . Cluster-hop is defined as the maximum number of direct hops between a cluster head node and nodes at the cluster periphery. This dynamic parameter determines the dimensions of the clusters formed by the algorithm and serves as an input for the cluster formation process.

DCA consists of two phases; Information Update and Cluster formation phase. In information Update phase each node calculates its PCC . Then, based on the cluster-hop r_{hop} , the local PCC dissemination process is carried out. While in Cluster formation phase, once the Combined Criterion dissemination process is effectuated and the nodes have a weighted view of their neighborhood, the cluster organization can be performed thereby forming the physical clusters.

4.1 Pas 1: information update

Firstly, each node calculates its PCC according to one of heuristic predictors (Algorithm 1, line 1). The second step

of the algorithm is a method of disseminating the estimated *PCC* of nodes in its r_{hop} neighborhood. The step-by-step operation of the algorithm is described in algorithm 1. Once the algorithm has performed r_{hop} runs of local iterations using the exchanged messages, each node in the network will have information about the node which has the maximum *PCC* in its r_{hop} neighborhood. The synchronization between nodes for the algorithm is not required since the node verifies (Algorithm 1, lines 9-12) for *PCC* information of neighboring nodes after a pre-defined wait period. This algorithm of Predicted Combined Criterion Dissemination is efficient and allows the formation of larger clusters bounded by r_{hop} .

Algorithm 1 Predicted Combined Criteria Dissemination Algorithm (PCCD)

```

1: Calculate Combined Criterion  $PCC(s_i)$ 
2:  $Max_i.CC \leftarrow PCC(s_i)$ 
3:  $Max_i.id \leftarrow i$ 
4: while  $r_{cl} > 0$  do
5:   Transmit  $Max_i$  to  $s_j \forall s_j \in N(s_i)$ .
6:   Wait a period of time to receive the  $Max_j$  from all  $s_j \in N(s_i)$ 
7:   if all messages received then
8:     Determine maximum  $Max_j.CC$ .
9:     if  $Max_j.CC > Max_i.CC$  then
10:       $Max_i.CC = Max_j.CC$ 
11:       $Max_i.id = Max_j.id$ 
12:   end if
13: end if
14:  $r_{cl} = r_{cl} - 1$ 
15: end while

```

4.2 Phase 2: Cluster formation

After the nodes have obtained a largest *PCC* using dissemination process; we move to the cluster formation phase which is a distributed process. We do not adopt a greedy approach of cluster formation where the largest *PCC* node advertises to other neighboring nodes to join their clusters. The objective is to form nearly equal sized and distributed clusters with self-decided cluster admissions. The steps of cluster formation procedure are described in Algorithm 2. Briefly DCA algorithm runs as follows:

Each node s_i which has the largest *PCC* value consider itself as cluster head and broadcast *Cluster_Head* message contains $PCC(s_i)$ value, otherwise s_i elects the node which has the largest *PCC* value and send *My_Cluster_Head* message (Algorithm 2, lines 1-7). If node s_i received a *Cluster_Head* message, it registers the message and waits for all *Cluster_Head* messages. If there are multiple messages, choose node which has lower

CC value as cluster head and sends a *Join* message to the source node (Algorithm 2, lines 8-13). The cluster head will keep information of all the nodes in its cluster in a cluster table and updates the cluster table depending on received *Join* messages (Algorithm 2, lines 14-16).

Algorithm 2 Distributed Cluster Algorithm (DCA)

```

1: if ( $Max_i.id == i$ ) OR ( My_Cluster_Head message received) then
2:    $s_i$  considered as Cluster head
3:   Broadcast a Cluster_Head( $CC(s_i)$ ) message for all neighbors
4: else
5:   Select node  $s_{j=Max_i.id}$  as cluster head
6:   Broadcast a My_Cluster_Head message to  $s_j$ 
7: end if
8: if  $s_i$  received Cluster_Head message then
9:   Wait a period of time
10:  if time expired then
11:    send Join message to  $s_j$  which has minimum CC
12:  end if
13: end if
14: if  $s_i$  received Join message from  $s_j$  then
15:   Update the cluster table by  $s_j$ 
16: end if

```

5 Simulation Results

The program has been simulated in visual C#.NET, with statistical data gathered to analyze runtime performance and scalability of the algorithm. The sensor nodes are assumed to be non-uniformly deployed at random on a two dimensional plane. We assume that the sensor nodes have the ability to adjust their transmission power according to the distance of the receiving node.

In order to measure the energy dissipation of nodes, we use the same energy parameters and radio model as discussed in [8], wherein energy consumption is mainly divided into two parts: receiving and transmitting messages. The transmission energy consumption needs additional energy to amplify the signal depending on the distance to the destination. Thus, to transmit and to receive a k -bit packet are given by equations 12 and 13, respectively. All simulation parameters are listed in Table 1.

$$E_{Tx}(k, d) = \begin{cases} kE_{elec} + k\epsilon_{fs}d^2 & d < (\epsilon_{fs}/\epsilon_{mp}) \\ kE_{elec} + k\epsilon_{mp}d^4 & d \geq (\epsilon_{fs}/\epsilon_{mp}) \end{cases} \quad (12)$$

and to receive this message, the radio expends will be

$$E_{Rx}(k) = k * E_{elec} \quad (13)$$

Table 1: Simulation parameters

Parameter	Acronym	Value
Electronics energy	E_{elec}	50 nJ/bit
Free space constant	ϵ_{fs}	10 pJ/bit/m ²
Multipath constant	ϵ_{mp}	0.0013pJ/bit/m ⁴
distance threshold	$(\epsilon_{fs}/\epsilon_{mp})$	87
Packet size	k	30 bytes
Packet rate	B	1 packet/s
Transmission range	R	$2r$
Sensing range	r	25 m
Cluster hope	r_{hop}	2

In order to carry out our experiments and study the performance of our approach, we have considered two evaluation scenarios: first scenario, we verify the effectiveness of each criterion in DCA, when using SPP, LEP and HP predictors. In second scenario we compare our method with Self-Organization Routing Protocol Supporting Mobile Nodes for Wireless Sensor Network (LEACH-M) [11]. We use network lifetime metric (the time interval from the start of operation until the death of the last alive node) to indicate the performance of our approach.

5.1 First Scenario

In this scenario, sensor nodes are randomly deployed in a region of size 200 m × 200 m. Number of deployed sensor nodes is 100 nodes. Our goal with this first set of simulations is to verify the effectiveness of connectivity criterion (i.e., $\alpha = 1, \beta = 0, \gamma = 0$, and $\zeta = 0$), coverage criterion (i.e., $\alpha = 0, \beta = 1, \gamma = 0$, and $\zeta = 0$), energy criterion (i.e., $\alpha = 0, \beta = 0, \gamma = 0$, and $\zeta = 1$) or mobility criterion (i.e., $\alpha = 0, \beta = 0, \gamma = 1$, and $\zeta = 0$) in DCA, when using SPP, LEP and HP predictors.

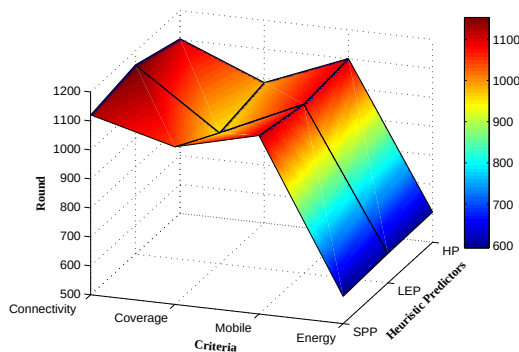


Figure 1: Network lifetime using the effectiveness of each criterion in DCA.

Fig.1 shows the network lifetime, where network life-

time is the time until the last node dies. Connectivity criterion clearly improves network lifetime over the others criterion. It can be also observed that HP predictor improves network lifetime in the case of using energy criterion while SPP improves network lifetime in the cases of using mobility, coverage, and connectivity.

5.2 Second Scenario

In this scenario, sensor nodes are randomly deployed in a region of size 200m × 200m and vary the number of sensors between 200 and 400 with increment of 50. We performed experiments using $\alpha = \beta = \gamma = \zeta = 0.25$.

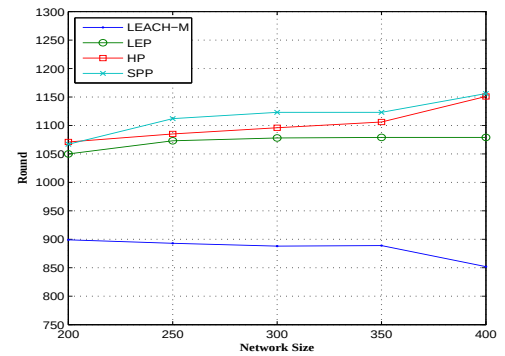


Figure 2: Network lifetime using DCA vs. LEACH-M when the sink locates at the center of network.

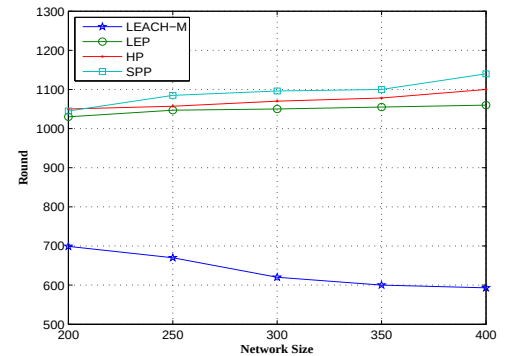


Figure 3: Network lifetime using DCA vs. LEACH-M.

In Fig.2, the sink locates at the center of network, (50,50). Simple calculus can show the sink at the centroid achieve the minimum mean squared Euclidean distance between nodes and the sink. Therefore, the main interests are consumed resources during clustering. This figure compares network lifetime to DCA corresponding to different predictors with LEACH-M. It can be observed that DCA algorithm clearly improves network lifetime over LEACH-

M clustering. This is because LEACH-M randomly selects cluster heads (and hence cluster sizes), which may result in faster death of some nodes. This is avoided in DCA because final cluster heads are selected such that they are well-distributed across.

In Fig.3, the sink is moved outside the network, DCA provided longer network lifetime than LEACH-M, which quickly decreases because it depend on the communication between the network and the sink.

In Fig.2 and Fig.3, can be observed that SPP predictor have the longest network life time and LEP predictor have shortest network lifetime with regardless of LEACH-m.

6 Summary and Conclusions

In this paper, we have proposed an approach for prediction-based clustering algorithm for MSNs. In our approach, we have presented the *Distributed Clustering Algorithm* (DCA) algorithm which uses adequate criterion for cluster-head election in conjunction with heuristic predictors to generate steady and balanced clusters. Simulations results demonstrate that the DCA algorithm provides better performance than LEACH-M in terms of network lifetime.

References

- [1] B. Ltr et al., "Heterogeneity Based Clustering Heuristic for Mobile Ad Hoc Networks" in Proc. ICC, Paris 2004.
- [2] Aman K. et al., "Sensing uncertainty reduction using low complexity actuation", In ACM Information Processing in Sensor Networks (ISPN), April 2004.
- [3] O. Younis, S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach", IEEE Transactions on Mobile Computing 3(4), 2004.
- [4] P. Kawadia, R. Kumar, "Power control and clustering in ad hoc networks", in: Proceedings of INFOCOM, 2003.
- [5] L. Reynaud and D. Meddour "An Open Architecture that Manages Quality of Service within Ad Hoc Networks", in Proc. IEEE IWVAN, 2004.
- [6] M. Chatterjee et al., "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks" Journal of Cluster Computing, V. 5/2, April, 2002.
- [7] T. Small, and Z. J. Haas., "The shared wireless infostation model: a new ad-hoc networking paradigm", In ACM Mobihoc, pp 233-244. ACM Press, 2003.
- [8] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks", IEEE Transactions on Wireless Communications 1 (4), 2002.
- [9] H. Huang, A. W. Richa, and M. Segal, Dynamic coverage in ad-hoc sensor networks, in Mobile Networks and Applications, Springer Science + Business Media, 10:9-17, 2005.
- [10] Y. T. Hou, Y. Shi, J. Pan, and S. F. Midkiff, Lifetime-optimal data routing in wireless sensor networks without flow splitting, in Workshop on Broadband Advanced Sensor Networks, San Jose, CA, 2004.
- [11] Do-Seong and Kim Yeong-Jee Chung, "Self-Organization Routing Protocol Supporting Mobile Nodes for Wireless Sensor Network" in Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences, Volume 2, PP 622 - 626, ISBN:0-7695-2581-4, 2006.
- [12] Arboleda L.M., Nasser N., "Comparison of Clustering Algorithms and Protocols for Wireless Sensor Networks", IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Ottawa, May 2006.
- [13] W. Choi, S.K. Das, "A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks". In: Proceedings of MobiQuitous Networking Conference, 2004.
- [14] M. Qin, R. Zimmermann, "An energy-efficient voting-based clustering algorithm for sensor networks", in: Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05).
- [15] Benenson Z., Gartner F., Kesdogan D., "User Authentication in Sensor Networks (Extended Abstract)", Germany, 2004.
- [16] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks", in Ad Hoc Networks, Volume 7, PP 955-972, 2009.
- [17] X. Li, "Genetic Algorithm Simulated Annealing Based Clustering Strategy in MANET", in Advances in Natural Computation, First International Conference, ICNC 2005, Changsha, China, Volume 3610, PP 1121-1131, 2005. .