2008

A battery aware clustering technique

Mohamed K. Watfa
University of Wollongong in Dubai, mohamed.watfa11@gmail.com

Omar Mirza
American University of Beirut, omm07@aub.edu.lb

Jad Kawtharani
American University of Beirut, jwk00@aub.edu.lb

A Battery Aware Clustering Technique

Mohamed K.Watfa
Computer Science Department
University of Wollongong
Dubai, UAE
Mohamed.watfa11@gmail.com

Omar Mirza and Jad Kawtharani
Computer Science Department
American University of Beirut
Beirut, Lebanon
{omm07, jwk00}@aub.edu.lb

Abstract— **Clustering allows for data aggregation which reduces congestion and energy consumption. Recent study in battery technology reveals that batteries tend to discharge more power than needed and reimburse the over-discharged power if they are recovered. In this paper, we first provide an online mathematical battery model suitable for implementation in sensor networks. Using our battery model, we propose a new Battery Aware Reliable Clustering algorithm for WSNs (BARC). BARC incorporates many features which are missing in many other clustering algorithms. It rotates cluster heads according to a battery recovery scheme and it also incorporates a trust factor for selecting cluster heads thus increasing reliability. Most importantly, our proposed algorithm relaxes many of the rigid assumptions that the other algorithms impose such as the ability of the cluster head to communicate directly with the base station and having a fixed communication radius for intra-cluster communication. BARC uses Z-MAC which has several advantages over other MAC protocols. Simulation results show that using BARC prolongs the network lifetime greatly in comparison to other clustering techniques.**

Keywords- *Clustering, Sensor Networks, Battery Awareness, Hierarchical, Load Balancing*

I. INTRODUCTION

Sensor networks represent a significant improvement over traditional sensors. In WSNs, the location of sensors need not be engineered or pre-determined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Our work falls under creating such protocols for self-organizing and collecting nodes into groups or *clusters*. Clustering in the wireless networks literature is used for wireless network management. There are two design approaches for management. In the first, we maintain knowledge of the network in each node and they achieve management themselves. This requires significant communication responsibility on individual nodes. Each node must maintain routes to the rest of the nodes in the network. In large networks the number of messages needed to maintain routing tables may cause congestion in the network. The second choice used in managing wireless networks is to identify a subset of nodes within the network and vest them with the extra responsibility of being a leader in charge of a group of nodes, the cluster, in their proximity. The leader, also called the cluster head (CH), is responsible for managing communication and routing among its group. This proves to be a better design choice; however, choosing clusters and CHs must be studied carefully. A CH must be selected diligently and must exhibit specific qualifications that entitle them to be leaders

Thus, a clustering algorithm in general attempts to find natural groups of components (or data, or nodes) based on some similarity. In this paper, we introduce a novel approach to clustering in Wireless Sensor Networks (WSNs). We show that battery aware sensors can make battery-state-informed clustering decisions that translate in energy efficient clustering schemes. Our Battery Aware Reliable Clustering (BARC) algorithm results in: increased energy efficiency (by using battery awareness techniques and cluster head rotation), load balancing (by limiting the number of nodes each cluster head can support), increased reliability (by introducing a trust factor), h-level clustering hierarchy, better bandwidth reuse, and increased network lifetime. Also, the proposed algorithm relaxes many of the rigid assumptions that the other algorithms impose such as the ability of the cluster head to communicate directly with the base station and having a fixed communication radius in [8] and [9] respectively. The rest of the paper is divided as follows. In Section II, we discuss related work in WSNs while highlighting their disadvantages. In Section III, we present and analyze our clustering algorithm. Section IV shows BARC's effectiveness via simulations and compares it to other clustering techniques using ns-2 simulator. Section V concludes this paper.

II. RELATED WORK

Many clustering protocols have been proposed for ad-hoc and sensor networks in the last few years each targeting a different goal. In [1], the authors propose using a spanning tree (or BFS tree) to produce clusters with some desirable properties. Energy efficiency, however, is not the primary focus of this work. In [2], the authors propose passive clustering for use with on-demand routing in ad-hoc networks. In [3], the authors propose LEACH clustering algorithm which assigns clusters and CHs according to a predefined probability. This probability is computed in a manner which ensures that all nodes in the network become CHs the same number of times. This approach does not take into consideration the energy dissipation in each node (and specifically the CH node) since this dissipation does not rely only on the number of times a node becomes a CH but also on some other parameters such as the number of nodes affiliated to each CH, the initial energy in the node itself, the communication distance, data aggregation, and other parameters. The authors also assume that all nodes are time synchronized, nodes have homogenous energy levels

initially, use 1-hop clustering, and that they can communicate directly with the base station. In BARC, we use the idea of rotating cluster heads; however, we relax these assumptions. Nodes may not communicate with the base station, energy levels are heterogeneous, and most importantly we will incorporate a new probability function which relies on a new energy model and takes into consideration the residual energy of the nodes and the state of the node's battery. Moreover, the proposed algorithm uses a d-hop clustering method. In HEED [4], the authors provide a distributed clustering algorithm which selects cluster heads based on the residual energy of each node. However, they require transmitting the control packets. Also, the authors assume that that the sensor nodes are uniformly and independently dispersed in a region. Classical clustering techniques for wireless sensor networks pay much importance to reducing the per-node energy consumption, which may not always guarantee a globally efficient solution. In [5], the authors propose E/sup 2/LBC, which considers energy efficiency as a system-wide issue that focuses on improving the overall stability of operation of a wireless sensor network. However, they assumed that CHs communicate directly with the base station and no rotation of CHs is performed. In [6], the authors study the effect of different communication paradigms (single hop vs. multi-hop) on the performance of clustering protocols. Clustering can also be a side effect of other protocol operations. For example, in topology management protocols, such as ASCENT [7], nodes are classified according to their geographic location into equivalence classes. A fraction of nodes in each class participate in the routing process, while other nodes are turned off to save energy.

III. The Clustering Algorithim

In this section, we introduce the main aspects of our proposed clustering platform and present the details of the BARC algorithm.

A. Assumptions

For the development of BARC we have assumed that:

- Each node has the computational power to support different MAC protocols and performs signal processing functions for aggregation.
- Nodes can control their transmission power to vary their communication range.
- CH has additional load incurred than normal nodes and that the nodes are loosely synchronized.
- Links are symmetric, i.e., two nodes $v1$ and $v2$ can communicate using the same transmission power level.
- The network serves multiple mobile/stationary observers, which implies that energy consumption is not uniform for all nodes.
- Nodes are location-unaware, i.e. not equipped with GPS-capable antennae. This justifies why some techniques are inapplicable.
- All nodes have similar capabilities (processing/communication), and equal significance. This motivates the need for extending the lifetime of every sensor.
- Nodes are left unattended after deployment. Therefore, battery re-charge is not possible. Energy-aware sensor network protocols are thus required for energy conservation.

It is important to note that in our model, no assumptions are made about (1) the homogeneity of node dispersion in the field, (2) the network density or diameter, (3) the distribution of energy consumption among sensor nodes, (4) the proximity of querying observers, (5) the ability to communicate with the B.S, and (6) each node having a fixed communication range.

B. Battery Awareness Background

Recent study in battery technology helps us better understand the battery behavior [8] and [9]. When discharging, batteries tend to consume more power than needed, and can reimburse the over-consumed power later. The process of the reimbursement is often referred to as *battery recovery*.

In general, a battery consists of cells arranged in series, parallel, or a combination of both. Two electrodes: an anode and a cathode, separated by an electrolyte, constitute the active material of each cell. When the cell is connected to a load, a reduction-oxidation reaction transfers electrons from the anode to the cathode. To illustrate this phenomenon, In a fully charged cell, the electrode surface contains the maximum concentration of active species. Active species are consumed at the electrode surface and replenished by diffusion from the bulk of the electrolyte. However, this diffusion process cannot keep up with the consumption, and a concentration gradient builds up across the electrolyte. A higher load electrical current I results in a higher concentration gradient and thus a lower concentration of active species at the electrode surface. When this concentration falls below a certain threshold, the electrochemical reaction can no longer be sustained at the electrode surface and the charge is unavailable at the electrode surface. However, the unused charge is not physically "over-consumed," but simply unavailable due to the lag between the reaction and the diffusion rates. If the battery current I is reduced to zero or a very small value, the concentration gradient flattens out after a sufficiently long time, reaching equilibrium again. The concentration of active species near the electrode surface following this recovery period makes unused charge available again for extraction. We refer to the unused charge as *discharging loss*. Effectively recovering the battery can reduce the concentration gradient and recover discharging loss, hence prolong the lifetime. Experiments on nickel-cadmium battery and lithium-ion battery show that the discharging loss might take up to 30% of the total battery capacity. Hence, precisely modeling battery behavior is essential for optimizing system performance in cluster head selections in sensor networks. Many different types of mathematical models have been developed to study battery behaviour. These models are mainly categorized into four groups: Physical, empirical, abstract and mixed. For WSNs the most efficient way to study the battery behaviour of nodes is by using a discrete time battery model which is an abstract model. Several analytical battery discharge models have been developed in [10]. Although these battery models are

computational approaches and independent of battery chemistry, they are not quite suitable for implementation in sensor networks protocols. The main drawback is that they are off-line models with high computational complexity, and the battery parameters have to be pre-computed.

Next, we introduce our online battery model that will play an important role in the cluster head selection. Intuitively, a good clustering algorithm would use *well recovered* nodes in the network to prolong the lifetime of the network.

C. An Online Battery Model

The problem encountered is to find an efficient on-line mathematical discrete battery model that captures the battery behaviour and can return the battery energy level at any time. Such a model should take into consideration the fact that sensor nodes have limited computational powers and memory. Therefore, it should reduce the computation complexity and the memory needed as much as possible.

The battery model proposed is based on the Rakhmatov model [10]. First, we will define the variables used in this model.

T = Battery time to failure = Lifetime.

i(t) = Discharge current.

β = A battery dependent constant.

α_T = Battery charge capacity before the battery started to discharge given by the equation

$$\alpha_T = \int_0^T \left[1 + 2\sum_{m=1}^{\infty} e^{-\beta^2 m^2 (T-\tau)}\right] i(\tau)d\tau$$

α_t = Battery charge capacity after duration *t* given by the equation

$$\alpha_t = \int_0^t \left[1 + 2\sum_{m=1}^{\infty} e^{-\beta^2 m^2 (t-\tau)}\right] i(\tau)d\tau$$

If *i(t)* is expressed as a set of n-step functions then

$$\alpha_t = \int_0^t \left[\Delta_k + 2\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (t-t_k-\Delta_k)} - e^{-\beta^2 m^2 (t-t_k)}}{\beta^2 m^2}\right] i(\tau)d\tau$$

$$\alpha_t = \sum_{k=0}^{n-1} I_k \left[\Delta_k + 2\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (t-t_k-\Delta_k)} - e^{-\beta^2 m^2 (t-t_k)}}{\beta^2 m^2}\right]$$

For each battery we should have its β, T and the discharge current. For each battery, we can know how much of its charge capacity it has lost. Actually, we evaluate the battery according to how much charge capacity it has lost. Let $r = \frac{\alpha_t}{\alpha_T}$ be the ratio of the charge capacity after duration *t* to the total battery charge capacity. Therefore, we can deduce the percentage of the used battery capacity and compare different batteries accordingly. The battery with smaller *r* means that it has been used more and hence requires more time for recovery. Moreover, we can deduce the recovery time as follows:

$\alpha_t =$ Charge capacity after time *t*

$\alpha_{t+\Delta} =$ Charge capacity after time t and recovery time Δ

Since the battery is recovering for time Δ, this implies that $\alpha_{t+\Delta} > \alpha_t$ or $\frac{\alpha_{t+\Delta}}{\alpha_t} > 1$. We can approximate this inequality as follows: $\alpha_{t+\Delta} = k.\alpha_t$ with $k > 1$. However, $\alpha_{t+\Delta} < \alpha_T$. Hence, $\alpha_{t+\Delta} = k.\alpha_t < \alpha_T \Rightarrow k.\alpha_t < \alpha_T \Rightarrow k < \frac{\alpha_T}{\alpha_t}$. So, now we have $k > 1$ and $< \frac{\alpha_T}{\alpha_t}$; we can approximate k as $k = \left\lfloor \frac{\alpha_T}{\alpha_t} \right\rfloor$. Note that this approximation is not accurate however it is very simple. Using k and $\alpha_{t+\Delta} = k.\alpha_t$, we can deduce the recovery time needed Δ from $\alpha_{t+\Delta} = k.\alpha_t$ with Δ being the only unknown.

D. Cluster Head Selection

The BARC algorithm is initiated every round. Each round consists of two stages initialization/setup and steady state. The round lasts for T seconds while the initialization/setup stage lasts for t seconds. We will elaborate on each stage in detail in the following sections. BARC allows the formation of a cluster in a WSN by electing a set of CHs, according to the battery recovery model, where each CH is responsible for servicing a set of nodes of a specific cluster. Each node requests to join a CH according to certain criteria, mainly, by evaluating which CH suits the exact needs of this node. Thus, a node selects a CH according to the degree of *trust* pertaining to this CH and depending on the CH's proximity to this node. By *proximity* we imply the cost incurred for a node to join this CH. On the other hand, a CH itself has the prerogative to choose which node to service and thus can reject a node that is requesting to join it. This technique enforces *load balancing* on each CH. After this initialization phase where CHs are elected and clusters are formed; a CH assigns a Z-MAC schedule for intra-cluster communication. Inter-cluster communication is carried on using CSMA/CA and data is routed via a geographic routing paradigm. After this initialization/setup phase a steady phase starts consisting of inter/intra-cluster data transmissions. After a time T a new round of BARC starts in order to rotate the CH and thus giving the recovered nodes a chance to take their part as CH in an effort to maximize the network lifetime and optimize battery usage.

To prolong the lifetime of the network, our clustering protocol should drain the battery at each node in the pulsed fashion. Thus, as discussed above, a WSN lifetime is divided into pre-determined, equal sized periods (T). Each period is divided into two phases, the cluster setup phase (t) and the data transfer phase. The cluster setup phase is itself divided

into two sub-phases, the CH election phase and the cluster formation phase. The former generates the set of well-recovered CHs as follows:

1. Every node includes in its periodic HELLO packets (at the beginning of time T), its ratio r and recovery time Δ as discussed in Section 3.3.
2. The battery with smaller r means that it has been used more and hence requires more time for recovery. Hence, each node elects itself as a CH if it has the smallest r (i.e. highest theoretical capacity) among its d-hop neighbors (optimal d is computed by analysis prior to WSN deployment). Ties are broken by using the recovery time Δ. A new candidate node is only called upon after its recovery time has elapsed in order to guarantee that it had enough time to recover.
3. If a CH runs out of energy during a given round then it broadcasts that it needs to be replaced by another node so that it could enter a recovery state. The next best recovered node in its neighborhood takes it place and announces that it has become the new CH. This cluster head rotation allows nodes to recover while other nodes are taking the responsibility of the cluster head and thus prolonging the network life time.

A node i receiving a CH announcement message, checks whether the CH is *trustworthy* or not (we define trust in the next section). Node i sends a join request message (JR) to the most trustworthy CH it has received an announcement from. Based on the CH's reply the node either joins this CH or seeks affiliation with the second best trustworthy CH it has also received an announcement from.

E. Other Features of BARC

The trust among nodes is represented by an opinion or confidence measure which is dynamically and frequently updated. The concept of trust has been previously mentioned in a number of research works. We used the concept of trust in a different manner in this paper. If one node performs normal communications, the opinion of others nodes towards this node is increased; otherwise, if a node misbehaves, it will be eventually denied by the whole network. It is assumed that the sensor network is equipped with some monitoring mechanisms or intrusion detection units so that one node can observe the behaviors of its d-hop neighbors. The evidences of this model are collected through the successful or failed state when nodes perform communications with other nodes. In WSN, a node may be uncertain about another node's trustworthiness because of the lack of enough collected evidence. Thus, we need to include this uncertainty in our trust model. An opinion therefore consists of a belief, disbelief and uncertainty.

In this model, each node maintains a three-dimensional opinion metric defined as follows:

$$W_{i,j} = (B_{i,j}, D_{i,j}, U_{i,j}) \quad (3)$$

where *B, C* and *U* correspond to the belief (the probability of a node *j* can be trusted by a node *i*), disbelief (the probability of a node *j* cannot be trusted by a node *i*), and uncertainty (fills the void in the absence of both belief and disbelief) respectively. These three elements satisfy:

$$B_{i,j} + D_{i,j} + U_{i,j} = 1 \quad (4)$$

In a WSN, each node will continuously collect all the positive and negative events about its neighboring nodes' trustworthiness. With these events, the opinion value is derived as follows:

$$B_{i,j} = \frac{p}{p+n+2};\ D_{i,j} = \frac{n}{p+n+2};\ U_{i,j} = \frac{2}{p+n+2} \quad (5)$$

where p and n are positive and negative events collected by node i about node j trustworthiness. In this paper, the positive events are the successful communication times between two nodes, and the negative events are the failed ones. Each time a node i performs a successful communication with another node j (i.e. forwarding requests or replies normally); j's successful events will be increased by one. On the other hand, j's failed events will be increased by one in case of a failed communication. Then the opinion value will be recalculated using equation (5). First, each node will initialize its opinion vector to (0, 0, 1) for all its d neighboring nodes. When a node receives a CH announcement message, it checks its opinion towards this CH to decide whether to affiliate to it or not as follows:

1. If the component belief is larger than 0.5, i trusts j and might choose j as its CH.
2. If the component disbelief is larger than 0.5, i does not trust j and will not choose j as its CH.
3. If the component uncertainty is larger than 0.5, i will request j's digital signature or any other mean to authenticate j or it might ask its neighbors about their opinion about node j and updates its trust value using any trust combination
4. If all the components are smaller than 0.5, step 3 applies.
5. Otherwise, node i waits for another CH announcement message.

Since prolonging the lifetime of the WSN is one of our main concerns, our approach to the energy/battery aware cluster head rotation is paralleled by a load balancing technique that reduces hot spots in the sensor network. We propose a technique that allows a CH to adequately choose the number of nodes it can appropriately service based on the average load a CH can handle and the expected incurred load that a node might apply in the next round of BARC. Our technique for load balancing mandates that each elected CH computes an average load, L, it can handle for the next round. L is an expectation of the work load a CH can sustain at a given round without depleting its energy. It is the average number of

transmissions received, processed, and sent that a CH expects to handle. This average, L, can be computed by incorporating the CH's residual energy level, the expected life time of the network, the energy model of the network, and the number of neighboring nodes. On the other hand, each node sending a join request (JR) to a CH will append its expected load to the JR. This load is computed based on the transmissions the node has sent in the last round of BARC. The CH will assess, while receiving the join requests, the node's anticipated load and will decide if it can handle this load. The CH either accepts the join request or rejects it by sending an acknowledgement (ACK) message or a rejection (REJ) message to the node requesting a join respectively.

In our approach, we utilize a multi level clustering hierarchy. We achieve this by running the BARC algorithm iteratively on the CHs we had computed. The algorithm works in a bottom-up fashion. BARC first elects the level-1 clusterheads, then level-2 clusterheads, and so on.

IV. SIMULATION RESULTS

In this section, we simulate BARC using ns2 simulator to show its effectiveness. We compare BARC to LEACH. LEACH clustering proved to be 4× to 8× more effective in prolonging the network lifetime than direct communication or minimum energy transfer. The simulation parameters are the similar to those used in LEACH to make the comparisons more effective as depicted in Table I.

Experiment 1 (*Network Lifetime*)
In the first experiment, we highlight the main strength of BARC compared to LEACH in terms of extending the network life time by 2x and 3x compared to LEACH depending on the number of deployed nodes. Also, a trusted version of BARC was also simulated to show the effects of the tradeoff between adding trust to BARC and the resulting computational and communication overhead. Fig. 1 compares the network lifetime using BARC with LEACH, where network lifetime is the time until the 10% of the nodes die (after that, the network would not be connected). BARC clustering clearly improves the network lifetime over LEACH clustering for all cost types. LEACH randomly selects cluster heads (and hence cluster sizes), which may result in faster death of some nodes. LEACH does not take into consideration the battery state of each node when selecting the cluster head. Also, the cluster head rotation established in BARC allows nodes to enter the recovery state while other nodes will play the role of the cluster head and thus increase the network lifetime. The load balancing technique reduces hot spots in the sensor network and increases the energy lifetime of the sensor network. Comparing the trusted version of BARC with LEACH, we notice that even though, extra overhead will be evident; the network lifetime is still extended. For large deployments, the trusted version of BARC will perform worse considering that more packets would be exchanged and monitored in order to evaluate the trust level of each node.

Experiment 2 (*Varying the Sink Location*)
In experiment 2, we changed the location of the sink and studied the effect of this change on the network lifetime. The distance is computed from the sink to the closest point to it on the network. The number of nodes was fixed at 1000. Fig. 2 shows that BARC prolongs network lifetime, compared to LEACH and to direct communication due to the embedded features in it as discussed in the previous simulation. Network lifetime severely deteriorates when using direct communication as the distance increases, which emphasizes the advantages of network clustering. Direct communication to long distances also results in severe interference problems, especially in dense networks. Using direct communication may be tolerable only in when the sink is very close to the data source in the network (which is not the case in most applications), to avoid clustering overhead.

Experiment 3 (*Battery Awareness Parameters*)
Our goal in this experiment is to study the effects of varying the sampling rate and/or the BARC round time.

a) *Varying the sampling rate with constant round time (T=10 secs)*: Let *delta* = 1/*f*, where *f* is frequency at which we sample the load of the battery at a time *t*. Fig. 3 simulates our algorithm for a varying the sampling rate (2, 5, and 10) respectively. Simulation show that the less the sampling rate, the longer the *network lifetime* (the ration of the number of nodes alive to the total number of nodes in the network). This is due to the more accurate sampling of the battery load leading to a more pulsed load as opposed to a constant load.

b) *Varying the round time with constant sampling rate (delta=2)*: Fig. 4 simulates our algorithm for a constant *delta* and a varying round time *T* (10, 20, and 30 secs). Experiments show that there is an optimal value to be used for the round time that is neither too small nor too large. A large round will exhaust a CH's energy and a small round will cause overhead in cluster formation.

TABLE I. SIMULATION PARAMTERS

Type	Parameter	Value
Battery	Lithium Ion battery	α=2.71.47, β=10.39
	Initial Energy	2J/node
	Dead Nodes	<0.1J
Radio	E_processing	50 nJ/bit
	E_communication	10 pJ/bit/m^2
	Free Space Model	Proportional to d^2
Network	Grid	(0,0) to (100,100)
	Sink	(50,175)
	# of Deployed Nodes	1000
Application	Data Packet Size	100 bytes
	Packet Header	25 bytes

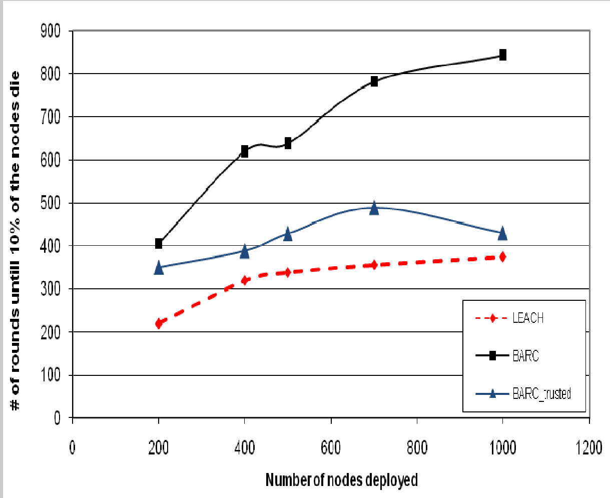

Fig. 1. Lifetime plots as the number of nodes increase.

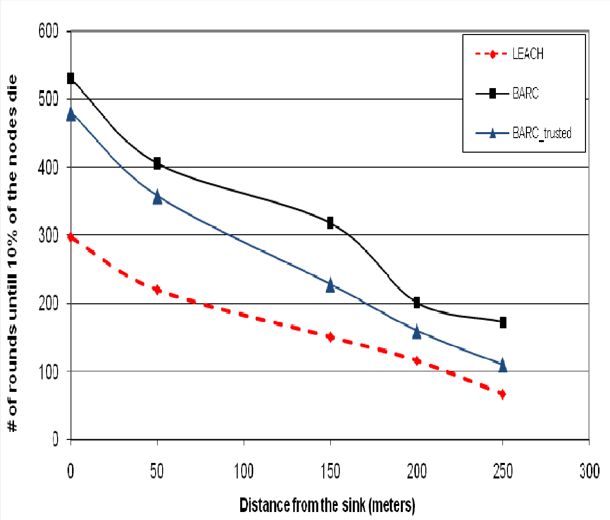

Fig. 2. Lifetime plots as the sink location changes.

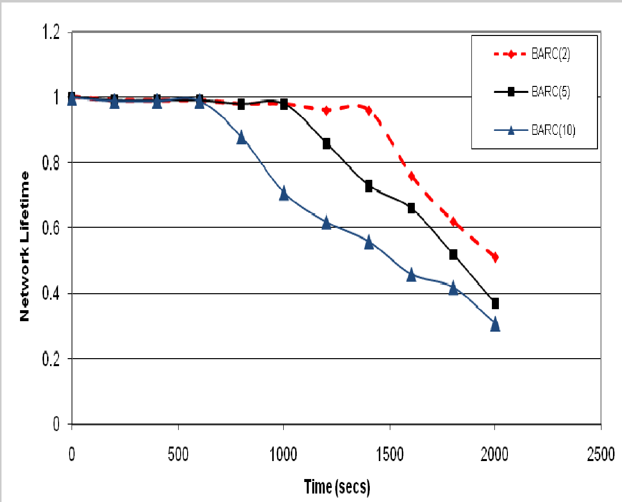

Fig. 3. Lifetime plots as we vary the sampling rate.

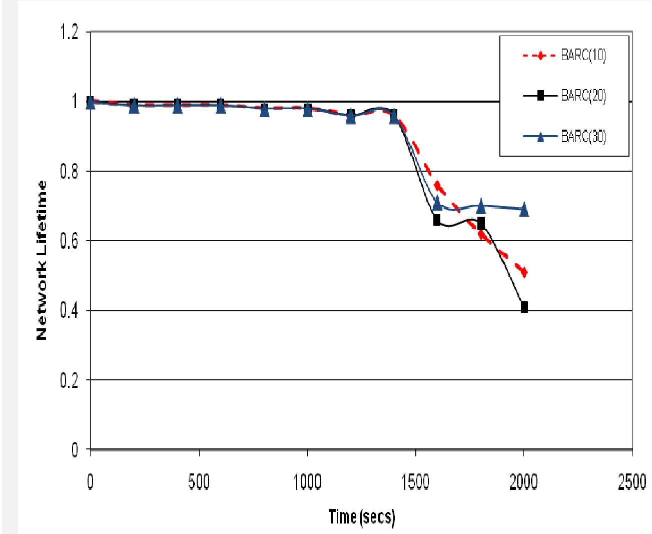

Fig 4. Lifetime plots as we vary the round time.

V. CONCLUSIONS

We have proposed a new clustering algorithm that combines several features that are not present in one existing clustering algorithm such as battery awareness, CH rotation, load balancing, hierarchal clustering and trust reliability. We validated the usefulness of the extra features added to BARC showing that it also increases the network lifetime.

REFERENCES

[1] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," in *Proceedings of IEEE INFOCOM*, April 2001.

[2] M. Gerla, T. J. Kwon, and G. Pei, "On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering," *WCNC* 2000.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transactions on Wireless Communications, Vol. 1, No. 4, October 2002, pp.660-670.

[4] 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, 2004, pp. 366-379.

[5] L. Jayashree, S. Arumugam, and N. Rajathi, "E/sup 2/LBC: An Energy Efficient Load Balanced Clustering Technique for Heterogeneous Wireless Sensor Networks", Wireless and Optical Communications Networks, 2006 IFIP International Conference on, Vol., Iss., 11-13 April 2006, pp.7-14.

[6] Mhatre, V., Rosenberg, C.: Design guidelines for wireless sensor networks: communication, clustering and aggregation. Elsevier Ad Hoc Networks (2003), pp. 45–63.

[7] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," in *Proceedings of IEEE INFOCOM*, NewYork, NY, June 2002.

[8] Chiasserini, C. F., & Rao, R. R. (2001, July). Improving battery performance by using traffic shaping techniques. IEEE JSAC Wireless Series, 19(7), 1385–1394.

[9] Chiasserini, C. F., & Rao, R. R. (2001, July). Energy efficient battery management. IEEE JSAC Wireless Series, 19(7), 1235–1245.

[10] Rakhmatov, D. N., & Vrudhula, S. B. K. (2001). An analytical high-level battery model for use in energy management of portable electronic systems. Proceedings 2001 IEEE/ACM Int"l Conf. Computer-Aided Design pp. 488–493.