EKOCA: Energy Aware Overlapping Multihop Clustering for Wireless Sensor Networks

Eman Ramadan, Moustafa A. Youssef, Magdy Abd-ElAzim Ahmed
Dept. of Computer and Systems Engineering
Alexandria University
Alexandria, Egypt
email: eman.ramadan@alexu.edu.eg, moustafa.youssef@ejust.edu.eg, magdy_aa@hotmail.com

Mohamed Nazih El-Derini
Dept. of Computer and Systems Engineering
Pharos University
Alexandria, Egypt
email: nazih.elderini@pua.edu.eg

***Abstract*—Among the algorithms developed for Wireless Sensor Networks is Overlapping Multihop Clustering. Overlapping clusters are useful in many network applications, such as intercluster routing, node localization and time synchronization protocols. In some environments, especially hostile, sensor nodes are always left unattended and there is no way to recharge them or exchange their batteries. Therefore, use of energy is a key issue in designing protocols for sensor networks in order to extend the network lifetime. In this work, we present a distributed energy aware overlapping multihop clustering algorithm based on the remaining energy of the sensors. Each node first elects itself as a cluster head according to a certain probability. During the next rounds, cluster head nodes select the new cluster head nodes based on the remaining energy of the sensor nodes in their clusters. The node with the highest remaining energy within a certain range from the cluster head node is elected to be the new cluster head node. The clustering process terminates in O(1) iterations and does not depend on the network topology or size. This algorithm is evaluated using NS2 Simulator. The proposed algorithm is intended to help in extending the life time of the network and balancing the energy consumption among different nodes by rotating the cluster head role.**

Keywords-Clustering; energy aware clustering; multihop clustering; overlapping clustering; sensor networks

I. INTRODUCTION

Wireless **S**ensor **N**etworks (WSNs) are no longer a new technology and there are lots of applications used in our daily lives, in military and civilian domains. One of the challenging topics in designing an application in this field is energy-aware protocols.

Energy is a scare resource for wireless sensor nodes. The energy consumed is divided into two categories either according to useful work or wasteful work. Energy consumed while transmitting/receiving data, querying requests and forwarding data is considered useful energy consumption. Since the communication between nodes is through a wireless medium, so there are collisions, which result in retransmissions of packets, also nodes can stay idle listening to the channel to send their data, in addition to the overhead of packet header even for small packets. These are all considered wasteful energy consumption. Energy consumption reduces network lifetime, which is defined as the time elapsed until the first node, or a certain percentage of nodes [1], uses up its energy.

Many clustering techniques have been suggested in order to reduce the energy consumption by grouping some nodes into clusters. Clustering depends on nodes classification in which some nodes work as cluster heads being responsible for collecting data from other nodes in their clusters. Then, cluster heads can forward the data to the base station or aggregate it and send it as a single packet. Thus, aggregation reduces the overhead of data packets' headers. Clustering helps in reducing useful energy consumption by improving bandwidth utilization as well as reducing wasteful energy consumption by reducing overhead.

Among the algorithms proposed for Wireless Sensor Networks is K-hop Overlapping Clustering Algorithm for Wireless Sensor Networks (KOCA) [2]. It is a novel clustering algorithm aimed at generating overlapping multihop clusters. Overlapping clusters are useful in many network applications, such as intercluster routing [3], node localization [4][5][6][7] and time synchronization protocols [8].

Since nodes in a sensor network have limited energy, extending the network lifetime and improving scalability become essential. Cluster head nodes are supposed to consume more energy than other nodes due to the management procedure for cluster formation and also data aggregation from all nodes to be sent to the base station or gateway. After a certain time, cluster head nodes lose all their energy and then their clusters will stop working.

In this work, we present a distributed energy aware overlapping multihop clustering algorithm based on the remaining energy of the sensors. Each node first elects itself as a cluster head according to a certain probability. During the next rounds, cluster head nodes select the new cluster head nodes based on the remaining energy of the sensor nodes in the cluster. The node with the highest remaining energy within a certain range from the cluster head node is elected to be the new cluster head node.

The paper is structured as follows: Section II formulates the energy aware overlapping k-hop clustering problem. In Section III, we present the details of the EKOCA heuristic algorithm for solving the problem. Section IV provides sim-

 ISBN: 978-1-61208-283-7

ulation experiments for evaluating the EKOCA algorithm. In Section V, we present related work. Finally, Section VI provides the concluding remarks.

II. EKOCA: Energy Aware Overlapping Multihop Clustering

In this section, we present the system model, definitions and notations and the problem formulation.

A. System Model

Sensors will be randomly and uniformly deployed in the area of interest. Each node has a unique ID and is not equipped with GPS. All sensor nodes transmit packets at the same power so they all have the same transmission range. All communications are over a single shared wireless channel. A wireless link between two nodes is established if they are within the wireless transmission range of each other. Two nodes are immediate neighbors if they have a wireless link between each other and the two nodes are referred to as 1-hop. The communication environment is assumed to be contention-free and error-free, so no retransmission is required. Some of the nodes will be elected as Cluster Heads (CHs). The number of clusters is variable. Nodes are static which is a typical assumption for sensor networks. CH is a relaying element which sends the data received to the base station. There is a continuous reporting for the monitored event. The network is homogeneous which is also typical for sensor networks and commonly used in literature, thus all the nodes are alike (i.e., all having equal capacity in terms of computation, communication and power). CHs are picked from the deployed sensors. Multihop will be used for intracluster routing. The clustering process is distributed which means there are no base stations responsible for clusters formation and all nodes have to cooperate to make decisions.

B. Definitions and Notations

- Network size (n): The number of sensor nodes in the network. Since the deployment area is fixed so changing (n) changes the node density.
- Cluster Radius (k): The maximum distance between any node in the cluster and the cluster head node.
- Short Round Cluster Radius (m): The maximum distance between any node in the cluster and the cluster head node while choosing the new cluster head node in the short round. The cluster head chooses the node with the highest residual energy to be the next cluster head from the nodes within this range.
- Cluster head probability (p): The probability that a node can be a cluster head at the first round.
- Overlapping degree between two clusters: The number of nodes which are common between the two clusters.

C. Problem Formulation

Energy aware overlapping clustering algorithm can be formulated as finding the set of cluster head nodes satisfying the following conditions:

- Coverage condition: This means that each node is either a cluster head or within k-hops from at least one cluster head.
- Overlapping condition: This means that for each cluster there is at least one cluster that overlaps with it and the overlapping degree between them is greater than a certain threshold.
- Connectivity condition: This means that during each short round, cluster head node is changed but no cluster formation is done and the nodes should communicate with the new cluster head smoothly without any problems.

III. EKOCA Heuristic

There are three kinds of nodes in the clusters generated by EKOCA:

- Cluster Head (CH): A cluster head maintains the required information about the nodes in the cluster and the neighboring clusters and how to reach them.
- Normal Node: It is a member node in only one cluster.
- Boundary Node (BN): It is a member node in more than one cluster.

In this section, we discuss the necessary data structures maintained at each node. Then, we describe the cluster head selection process and the cluster membership.

A. Data Structures

Each node maintains the following variables:

1) Node ID (NID): A unique ID assigned to each node before deploying the network.
2) Adjacent Clusters Table (AC_table): A table maintained by CH nodes to store information about adjacent clusters. The table consists of tuples in the form (CHID, BNL, costL, oldCHID), where CHID is the CH node ID of adjacent cluster, BNL is a list of IDs of boundary nodes to reach this CH accompanied by the cost required to reach it and oldCHID is used to maintain the transition between the old cluster head and the new cluster head when no cluster reformation is required in short rounds. [see Table I]
3) Boundary Table (CH_table): A table maintained by each node to store information about the clusters known to this node. If the table contains more than one entry, this means that the node is a boundary node; otherwise, it is a normal node. The table consists of tuples in the form (CHID, HC, prev, oldCHID), where CHID is the CH node ID, HC is the number of hops leading to this cluster head, prev is the node ID of a 1-hop neighbor node that can lead to this CH node

using minimum number of hops and oldCHID is used to maintain the transition between the old cluster head and the new cluster head when no cluster reformation is required in short rounds. [see Table II]

4) Nodes Table (nodes): A table maintained by CH node to store information about the nodes in its cluster.

B. Cluster Head Selection

The main issue in the clustering process is to select a group of nodes to be cluster head nodes and these nodes then should gather other nodes to form different clusters. The cluster head node is supposed to consume more energy than the other nodes in the cluster for the management tasks it performs as well as aggregating data and sending it to the base station or the gateway. If this cluster head node lost all its energy, then the cluster would not be able to send the sensed data and this cluster will be separated from the whole network. Hence, we need to replace this cluster head node with another node so as not to let this cluster stop working. Rotating the role of CHs among nodes in the cluster can also be a means for fault-tolerance in addition to their load balancing advantage. The EKOCA clustering process is divided into three different types of rounds: First or Initial Round, Short Round and Long Round. The selection of the cluster head nodes differs in each of these rounds. We will now illustrate the differences in the selection process among these rounds.

This clustering algorithm is a distributed clustering which means that all nodes should cooperate to collectively specify the cluster heads and form overlapping clusters. There is no gateway or base station which has all information to decide how clusters will be formed.

1) First Round: This is the initial round that happens only once at the beginning of the clustering process. During this period each node generates a random number between zero and one and based on a certain probability, which is a parameter to the algorithm, this node will either be elected as a cluster head node or not. Then cluster head nodes will advertise themselves through an advertisement message (CH_AD) to the sensors within their ranges. This message is then forwarded to all sensors that are no more than k-hops away from the cluster head which represents the cluster radius. This forwarding limitation is used to limit the flood of advertisement messages. A sensor that receives such advertisements joins the cluster even if it already belongs to another cluster. Since forwarding of the advertisement message is limited to k-hops, if a node did not receive any advertisement message within a reasonable time duration, it deduces it is not in the range of any cluster head and hence elects itself as a cluster head node and starts sending advertisement message. After this round, clusters are formed and nodes start sensing data and send these data to the cluster head. The cluster head then sends the data to the base station. After a certain time, the energy of the cluster head node is consumed more than other nodes so we need to change the cluster head node through the short round.

Example for AC_Table for CH node (n9) of cluster G:

TABLE I: EXAMPLE OF AC_TABLE

CHID	oldCHID	BNL
n4	n7	(n1,5), (n2,4)
n6	n8	(n5,2), (n3,1)

Example for CH_Table for node n3:

TABLE II: EXAMPLE OF CH_TABLE

CHID	oldCHID	HC	prev
n9	n12	3	n10
n6	n8	1	n11

2) Short Round: In this round, each cluster head node sends an energy request (ENREQ) message to the nodes in the cluster within m-hops where m represents the short round cluster radius and it is less than or equal to k (the cluster radius). When a node receives this ENREQ message, it sends its current energy through an energy reply (ENREP) message to the cluster head node and forwards the ENREQ message to all sensors that are no more than m-hops. The current cluster head node compares the energies received from different nodes and selects the node with the highest residual energy to be the next cluster head node of the cluster. Thus, an election (ELECSHRTRND) message is sent from the current cluster head node to the new elected node and this message contains the information maintained about the cluster to be stored in the new cluster head node. Upon receiving the election message, the new elected node updates its data structures and hence, sends a new advertisement (NewCH_AD) message to all the nodes in the cluster in order to inform them that it is now the new cluster head node to which the sensed data shall be sent. This NewCH_AD message is forwarded to all sensors that are no more than (k + roundNum * m) hops, as the selected node is now deviated from the center and k-hops are not enough to reach all the nodes in the cluster. During this round, no cluster reformation is performed, only the new cluster head maintains the cluster information from the old cluster head and the nodes update their data structure according to the new cluster head node. There can be one or more short rounds before the long round.

3) Long Round: In this round, the current cluster head node sends an energy request (ENREQ) message to all the nodes in the cluster. When a node receives this ENREQ message, it sends its current energy through an energy reply (ENREP) message to the cluster head node and forwards the ENREQ message to all sensors that are no more than (k + maxNumberofShortRounds * m) hops. The current cluster head node compares the energies received from different nodes and selects the node with the highest residual energy

 ISBN: 978-1-61208-283-7

to be the next cluster head node of the cluster. An election (ELECLNGRND) message is then sent from the current cluster head node to the new elected node. Upon receiving the election message, clustering reformation is required and the new elected node starts advertising itself through an advertisement message (CH_AD) to the sensors within its range. This message is then forwarded to all sensors that are no more than k-hops away from the cluster head which represents the cluster radius. There can be one or more short rounds before the long round. A cluster cycle is defined as one or more short rounds followed by a long round.

C. Cluster Membership

Each node has a CH_table which stores information about the cluster(s) it belongs to. Upon receiving a CH_AD message, a node adds an entry to its CH_table. The CH_AD message's header includes SID, CHID, and HC, where SID is the sender node ID, CHID is cluster head ID, and HC is the number of hops leading to the CH node. The SID field is used to update the CH_table.prev field for each node to know the path to the cluster head. The HC field is used to limit the flood of CH_AD message to k-hops. Upon receiving a new CH_AD message, a node will add an entry to its CH_table. The SID field is used to update the CH_Table.prev field. If another CH_AD message was received from the same CHID, the HC is then compared to that stored in the table and if it was less than the old one, the HC and SID field are updated for the new shorter path to the CHID. Usually shorter path messages arrive faster but there may be a delay in the MAC layer. If a node has more than one entry in its CH_Table, this means that this node is a boundary node. After the propagation of CH_AD messages, each node then sends a join request (JREQ) message to each cluster it belongs to so that it becomes a member of this cluster. The JREQ message's header includes CHID, SID, RID, RSID, nc and (CHID, cost)$_{1..nc}$, where SID is the sender node ID, CHID is cluster head ID to join, RID is the receiver node ID in order to send this message to the CHID, its value is obtained from the CH_Table.prev field of the entry of this CHID, RSID is the node ID which originally sends this JREQ to the CHID, nc is the number of clusters this node can hear from, which is equal to the size of the CH_Table, and (CHID, cost)$_{1..nc}$ represents the cluster(s) this node can hear from and the cost to reach the CHID of these clusters.

Each cluster head node maintains a list of all the member nodes in the cluster and also another list of all adjacent clusters. These adjacent clusters are known from the JREQ (CHID, cost)$_{1..nc}$ field and store all the boundary nodes between the adjacent clusters and the hop count to reach the adjacent cluster head node. There can be more than one boundary node between overlapping clusters. Also the same node can be a boundary node for more than two overlapping clusters. Each CHID sets a timeout for receiving the JREQ messages which is enough for a message to be forwarded along k-hops.

EKOCA terminates in O(k + c m) steps. Typically, k, m and c are constants, so the clustering process terminates in a constant number of iterations regardless of the network size for all the different rounds.

IV. Performance Evaluation

In this section, we describe the network settings used to evaluate the performance of EKOCA protocol using NS2 Simulator [9]. First, we define the simulation environment then the performance metrics used. This is followed by the experimental results and comparison with KOCA algorithm.

A. Simulation Environment

Sensor nodes are deployed randomly over a flat square area of dimensions $100 * 100\ m^2$. Simulation parameters are shown in Table III.

The experiments consist of an initial round, followed by a 3 short rounds and then a long round. After adjusting the clusters for each round, data sensing and forwarding processes starts till the next coming round.

B. Performance Metrics

We use the following performance metrics:

1) *Percentage of covered nodes*: defined in the first and long rounds as the percentage of nodes which became cluster heads or heard an advertisement message from a cluster head within k-hops in the first wave of cluster head advertisements.
2) *Communication overhead*: defined as the total number of messages transmitted.
3) *Time until the first node dies*: defined as the time elapsed until the first node in the network dies.
4) *Average Remaining Energy*.
5) *Number of nodes alive per certain time in seconds*.

C. Results

Fig. 1 shows the effect of changing the cluster head probability *p* and the cluster radius *k* on EKOCA. Fig. 2 compares the performance of EKOCA vs. KOCA.

TABLE III: SIMULATION PARAMETERS

Parameters	**Value**
Cluster head probability (p)	0.05, 0.1, 0.15 and 0.2
Number of nodes	200
Deployment Area	100*100 m^2
Cluster Radius (k)	2, 3, and 4
Short Round Cluster Radius (m)	1
Transmission range of each node	10m
Number of short and long Rounds	4
MAC Layer	TDMA
Propagation Model	Free Space model

1) Effects of Parameters on Node Coverage: Fig. 1a shows the percentage of nodes covered increases by increasing *p* as the number of cluster heads increases so there is more chance to cover all nodes. We can also notice that by increasing *k* the percentage of nodes covered increases as more nodes are reached and the nodes per cluster increases.

2) Effects of Parameters on Communication Overhead: Fig. 1b shows the communication overhead increases by increasing the cluster head probability *p* which leads to increasing the number of clusters which leads to the increase of the number of communication messages sent or forwarded. It also increases with increasing the cluster radius *k* due to increasing the number of nodes per cluster which leads to more communication overhead messages.

3) Effects of Parameters on First Node Dies: Fig. 1c shows the time when the first node dies decreases by increasing the cluster head probability *p* which leads to increasing the number of clusters and since they are overlapping clusters so the boundary nodes will be responsible for sending and forwarding more communication messages. This time also decreases with increasing the cluster radius *k* due to increasing the number of nodes per cluster which leads to more communication overhead messages.

4) EKOCA vs. KOCA First Node Dies: Fig. 2a shows that due to the rotation of the CH role, the time when the first node dies in EKOCA is more than that of the KOCA algorithm. We can also notice that this time decreases by increasing the cluster radius *k* due to increasing the cluster size which accordingly causes the increase of communication overhead.

5) EKOCA vs. KOCA Average Remaining Energy: Fig. 2b shows that the average remaining energy in EKOCA is higher than that of KOCA algorithm due to energy balance. We can also notice that the average remaining energy decreases by increasing the cluster radius *k* due to the increase in the cluster size which accordingly causes the increase of communication overhead.

6) EKOCA vs. KOCA Number of Nodes Alive: Fig. 2c shows that the number of nodes die in KOCA is higher than that of EKOCA and this gap increases as time passes due to the CH rotation and the balance in the energy consumption. Thus, we can notice that the nodes die slower in case of EKOCA algorithm. This extends the network lifetime more than KOCA.

7) EKOCA vs. KOCA Communication Overhead: The communication overhead of the EKOCA algorithm is higher than that of the KOCA algorithm and this is due to the energy request and reply messages sent or forwarded to determine the node with the highest remaining energy to be the new cluster head. This difference increases by increasing the cluster radius (k) due to increasing the cluster size.

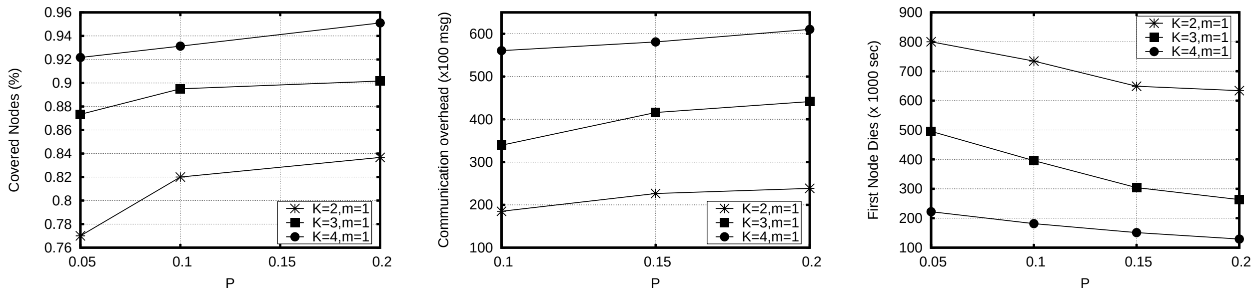
V. Related Work

There are many algorithms proposed for energy efficient clustering for wireless sensor networks either single hop or multihop for nonoverlapping clustering. Among these algorithms are [10][11][12][13]. (a) Energy Efficient Hierarchical Clustering (EEHC) [10]: Bandyopadhyay and Coyle proposed a distributed randomized clustering algorithm for WSNs with the objective of maximizing the network lifetime. Their technique depends on two stages: initial and extended. In the initial stage nodes form single hop clusters and in the extended stage they build h levels of clustering hierarchy which is recursively repeated to form additional tier. (b) Low Energy Adaptive Clustering Hierarchy (LEACH) [11]: Heinzelman et al proposed a distributed algorithm where some nodes elect themselves as CHs based on probability p and broadcast it for other nodes to join their clusters based on minimum communication energy between these nodes and CH. The role of being a CH is rotated periodically among the nodes of the cluster in order to balance the load. (c) Hybrid Energy-Efficient Distributed Clustering (HEED) [12]: is a distributed clustering algorithm where nodes are selected to be cluster heads based on energy and communication cost and other nodes go through several iterations until they find the CH they can communicate with using the least transmission power. (d) Distributed Weight-Based Energy-Efficient Hierarchical Clustering (DWEHC) [13]: Ding et al. proposed a distributed algorithm which differs from HEED in generating balanced cluster sizes and optimizing the intra-cluster topology. Nodes are selected to be CH based on their energy and proximity of their neighbors and other nodes try to join minimum cost reachable CH. Up to our knowledge KOCA was the first overlapping multihop clustering algorithm in which nodes can be members of more than one cluster at the same time, but it was not based on residual energy of sensor nodes while forming clusters.

VI. Conclusion and Future Work

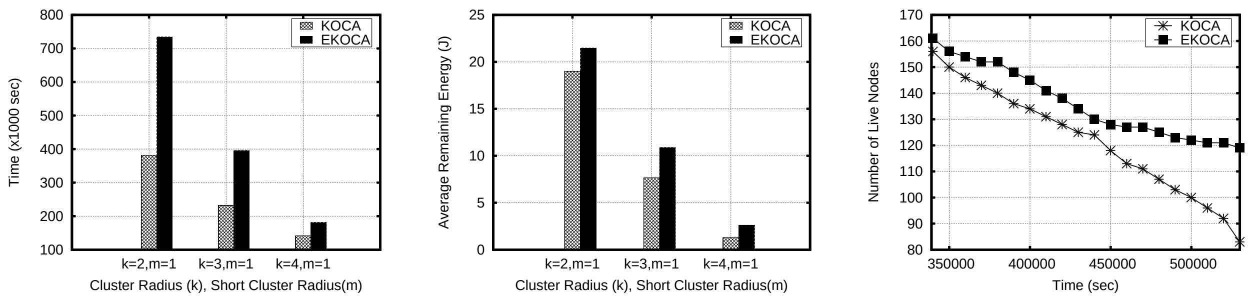
In this paper, we presented EKOCA algorithm as an energy-aware overlapping multihop protocol and explained the behavior of the algorithm in short and long rounds. The performance of EKOCA was simulated using NS2 simulator. We studied the effect of different parameters on EKOCA and compared it with KOCA algorithm. The results show that EKOCA helps in extending the life time of the network and balancing the energy consumption among nodes by rotating the cluster head role. It also increases the time before the first node dies and as it may not be a CH, it will not prevent the cluster from working. This comes with an increase in total communication overhead due to control messages for re-clustering. This work can be extended by analyzing the protocol at different error rates and proposing a schedule for the time to turn different nodes ON and OFF.

 ISBN: 978-1-61208-283-7



(a) Node Coverage (b) Communication Overhead (c) First Node Dies

Figure 1: Effect of changing the parameters on the performance of EKOCA.



(a) First Node Dies (b) Average Remaining Energy (c) Number of Nodes Alive

Figure 2: Comparing the performance of EKOCA and KOCA

REFERENCES

[1] D. M. Blough and P. Santi, “Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks,” in Proceedings of the 8th annual international conference on Mobile computing and networking. ACM, 2002, pp. 183–192.

[2] M. Youssef, A. Youssef, and M. Younis, “Overlapping multihop clustering for wireless sensor networks,” Parallel and Distributed Systems, IEEE Transactions on, vol. 20, no. 12, Dec. 2009, pp. 1844–1856.

[3] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, “A cluster-based approach for routing in dynamic networks,” ACM SIGCOMM Computer Communication Review, vol. 27, no. 2, 1997, pp. 49–64.

[4] A. Youssef, “Salam: A scalable anchor-free localization algorithm for wireless sensor networks,” PhD dissertation, Computer Science Dept., Univ. of Maryland, 2006.

[5] A. Youssef, A. Agrawala, and M. Younis, “Accurate anchor-free localization in wireless sensor networks,” in Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International. IEEE, 2005, pp. 465–470.

[6] Y. Shang and W. Ruml, “Improved mds-based localization,” in INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4. IEEE, Mar. 2004, pp. 2640–2651.

[7] X. Ji and H. Zha, “Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling,” in INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4. IEEE, 2004, pp. 2652–2661.

[8] M. Mamun-Or-Rashid, C. Hong, and C.-H. In, “Passive cluster based clock synchronization in sensor network,” in Proc. Advanced Industrial Conf. Telecomm./Service Assurance with Partial and Intermittent Resources Conf./E-Learning on Telecomm. Workshop (AICT/SAPIR/ELETE ’05), Jul. 2005, pp. 340–345.

[9] “The network simulator (ns2) http://www.isi.edu/nsnam/ns/,” [accessed 17.06.2013].

[10] S. Bandyopadhyay and E. J. Coyle, “An energy effcient hierarchical clustering algorithm for wireless sensor networks,” in INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol. 3. IEEE, Apr. 2003, pp. 1713–1723.

[11] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, Jan. 2000, pp. 10–19.

[12] O. Younis and S. Fahmy, “Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” vol. 3, no. 4, 2004, pp. 366–379.

[13] P. Ding, J. Holliday, and A. Celik, “Distributed energy-efficient hierarchical clustering for wireless sensor networks,” in Distributed Computing in Sensor Systems. Springer, Jun. 2005, pp. 322–339.