

Efficient Survivable Self-Organization for Prolonged Lifetime in Wireless Sensor Networks

Abderrahim Maizate
STIC Laboratory
Chouaib Doukkali University,
B.P: 20, El Jadida, Morocco

Najib El kamoun
STIC Laboratory
Chouaib Doukkali University,
B.P: 20, El Jadida, Morocco

ABSTRACT

Wireless sensor network is a large number of small battery powered sensors, where the failures of sensor nodes and the loss of connectivity are common phenomena. Therefore, energy consumption is an important issue to achieve a longer network lifetime. Several clustering protocols have been aimed to provide balancing of the residual energy particularly between the clusterheads and minimizing the number of clusterheads. This paper presents a novel clustering algorithm named EDED (Enhanced distributed, energy-efficient, and dual homed clustering) which provides robustness, a distributed cluster formation and reduces the number of clusters (clusterhead) in the WSN. Simulation results confirm that EDED is effective in prolonging the network lifetime and it can further efficiently relay the cluster data.

Keywords

Clustering algorithms, Cluster head, Energy consumption, CH selection, energy efficiency, sensor nodes, and Wireless sensor networks.

1. INTRODUCTION

The wireless sensor network consists of a large number of sensor nodes deployed in a remote region to sense events inside the phenomenon or very close to it. These sensor nodes are generally equipped with sensing, data processing and communicating component for gathering data from a field [1]. A wireless sensor network typically consists of a base station and a group of sensor nodes (see Figure 1). When sensors are deployed, they self organize to form a network and then start sensing the surroundings in order to transmit the gathered data to the base station. Since sensor nodes have limited energy, the self organization protocols must focus primarily on energy/power conservation of large scale WSNs. These networks are used for the systematic collection of information related to the environment such as intrusion detection, weather prediction and detection of environmental conditions.

In recent years, many kinds of efforts have been done on maximizing network lifetime as it is impractical to change or replace exhausted batteries of sensor nodes [2]. Such a constraint must be taken into account in the design of wireless sensor networks to minimize the energy consumption and allow the exchange of large amounts of data between nodes and the base station. These two competing objectives reveal the importance of an efficient self organization in WSNs. Therefore, many algorithms have been proposed for an efficient energy management in order to maximize the lifetime of wireless sensor networks.

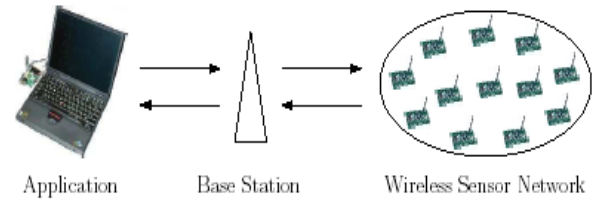


Figure 1: Architecture WSN.

Self-organization can be defined as the emergence of a global behavior from local interaction [3]. Wireless sensor networks are bandwidth and energy constrained. Self organization algorithms that minimize the number of message transmissions (and receptions) are preferable. The challenge is to organize dynamic and spontaneous nodes to form a network and prolonging the life-time, while satisfying the constraints of service quality. It is therefore necessary for self-organizing algorithms proposed to limit control packet exchanges for a minimum expenditure of energy and preserve the structure of self-organization for better stability and reliability the network.

Clustering is a key technique to improve the network lifetime, reduce the energy consumption and increase the scalability of the sensor network. It partitions the nodes into a number of small groups called clusters. Each cluster has a leader, known as a clusterhead (CH) to whom other member nodes (MNs) of the cluster communicate. Clustering consists in two-tier hierarchy, in which higher tier is formed by clusterhead and lower tier is formed by member nodes. The member nodes send their data to their immediate clusterheads. The clusterheads conduct data aggregation and send results to the base station directly or using multihop communication using other CHs. CHs often transmit data of all members over long distances, they loose more energy compared to member nodes. The network may be reclustered periodically in order select nodes with more energy to serve as CHs.

The paper is organized as follows: Section 2 presents related work. The proposed protocol is detailed in Section 3. Simulation results are presented and discussed in Section 4. Finally, Section 5 concludes the paper and announces our future work.

2. PREVIOUS WORK

Prolonging Sensor Network Lifetime and power conservation techniques for wireless sensor networks have been the focus of numerous studies [4], [5], [6], [7], [8], [9]. However, we focus on three papers most relevant to our work, LEACH [10], SOS [11] and DED [12]. LEACH [10] operations are

divided into set up phase (cluster organization, CH role rotation) and steady state phase (data collection, data aggregation and data forwarding). During the set up phase, each node locally determines whether it will become a CH or not. The CH selection is based on the residual energy of the node. Those with higher residual energy have the advantage during the CH competition. Clusters are created by non-CH nodes choosing to join a CH based on the signal strength of advertisements received from CHs. It has been shown that LEACH reduces energy consumption and improves network lifetime compared to fixed CH schemes.

Self Organizing Sensor Networks Using Intelligent Clustering (SOS) [11] provides a Self Organizing Sensor (SOS) network based on an intelligent clustering algorithm which forms the smallest number of clusters that makes all nodes of network to belong to one cluster. Minimizing the number of cluster heads provide an efficient cover of the whole network and minimizes the cluster overlaps. This reduces the amount of channel contention between clusters, and also improves the efficiency of algorithms that executes at the level of the cluster-heads. The clusters are created with the node, which has the maximum number of followers. Clustered nodes will be included into a clustered node set G. The node which can communicate with a node in G and has the maximum number of followers will be the next head node, and we make a cluster with it. If there exist an unclustered node or nodes we apply the same principle.

In [12], DED (distributed, energy-efficient, and dual-homed clustering) provides robustness for WSNs without relying on the redundancy of dedicated sensors and node density. DED uses the information already gathered during the clustering process to determine backup routes from sources to observers, thus incurring low message overhead. It does not make any assumptions about network dimension, node capacity, or location awareness and terminates in a constant number of iterations. The algorithm is divided into three phases for clarity. Phase I initializes the parameters and variables. At the start of clustering, the set of neighbours is needed to compute cost and PCH. The minimum value for PCH is set to a predetermined threshold P_{min} to ensure a constant time complexity of the algorithm. In Phase II, each node waits for the announcements of CH candidates that are advertised based on PCH. In Phase III, the backup is determined locally from the information that a sensor collected at the end of previous phase. The preferable backup of an MN is another CH within cluster range, if any. Otherwise, the node randomly chooses one of its neighbors whose CH is different from its own CH. If no such neighbor is found, the node picks up a neighbour whose CH is the same as its own CH, but whose backup destination is not the same as its own CH.

3. PROPOSED PROTOCOL

This section presents the network model used, Problem statement, and detailed description of the proposed algorithm.

3.1 Network Model

Consider a sensor field consisting of set of randomly deployed sensors in a rectangular field. In this network, the sensing tasks and data reporting are periodic. The algorithm assumes the following properties about the sensor network model:

- Sensor nodes are mobile.
- Sensor nodes are densely deployed.
- Sensor nodes have similar capabilities for sensing, processing and communication.

- Sensor nodes transmit data to its immediate cluster head in the allotted time slots or to the backup.
- All nodes are energy constrained and perform similar task.

3.2 Problem statement and EDED mechanism

A description of clustering problem can be defined as follows. Suppose that the nodes are randomly dispersed in a field. At the end of clustering process, several clusters are generated. Each cluster consists of a single cluster head and a bunch of followers. Each node belongs to a single cluster and be able to communicate with the cluster head directly via a single hop. The challenge of clustering algorithm is to form the smallest number of clusters that makes all nodes of network to belong to a cluster and rotating the role the cluster head between nodes with more energy. Minimizing the number of cluster heads would not only provide an efficient cover of the whole network but also reduces the number of failure in the network and the cluster overlaps. Another important point is the energy minimization consumed by each node, which is obtained by reducing the amount of communications among the sensor nodes and providing alternate routes from sources to the base station.

EDED algorithm has been designed with the following principles in mind:

- Cluster formation is initiated and maintained in distributed manner, through self organization of sensor nodes.
- Minimum numbers of cluster are created throughout the lifetime of the network.
- Each node has a backup path to the base station.
- Network coverage must be done with a minimum number of clusterheads.
- CHs should be evenly distributed throughout the network.
- Clustering process should terminate within a finite interval.
- Energy consumption should be well distributed among sensor nodes by rotating the role of the CH.

Each node should have a primary Clusterhead and a secondary backup which may be either another Clusterhead or an ordinary node. Data is forwarded to the secondary destination in case the primary Clusterhead fails. Figure 2 show how sensors may be dual homed. Here, nodes 8 and 11 are backed up through the directly reachable CH of a different cluster. Since no such CH is found for node 3, it is backed up by an ordinary node of a neighboring cluster. In the third case, as for node 5, the backup is a neighbor 3 who has the same primary Clusterhead (thus using the neighbor's backup). Otherwise, which may occur when the CH is the only neighbor of an ordinary node, an ordinary node itself may become a CH after its designated CH fails.

We note that there are two approaches to protect the structure of clusters and avoid triggering of reclustering. The first approach is to protect the cluster as a whole through a CH backup which involves exchanges between clustrhead and the backup clusterhead and packages to notify members. Therefore, an additional consumption of energy by different clusters. The second approach provides protection for each sensor node independently which performs fault management of a distributed and decentralized manner and enables fault tolerance especially when multiple failures.

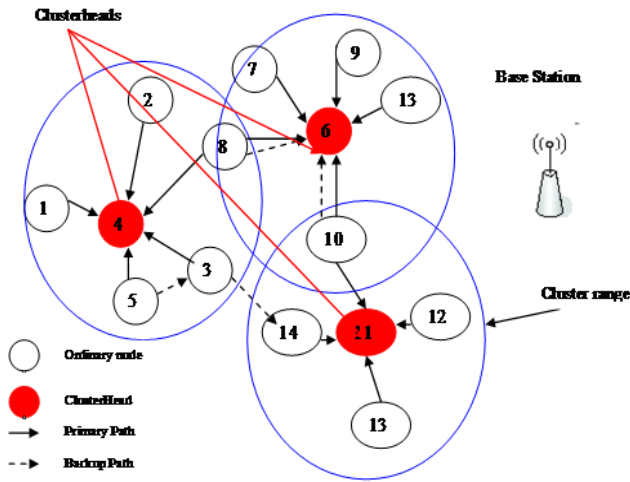


Figure 2: Cluster Structure with EDED.

3.3 EDED: Enhanced distributed, energy-efficient, and dual homed clustering

The CH selection process in EDED is performed in two steps: first a few nodes advertise themselves as tentative CHs according to value of P_{CH} and the number of neighbors. Then ordinary nodes select from these advertised nodes as final CHs.

The algorithm starts by initializing the parameters and variables. At the start of clustering, the set of neighbours is needed to compute P_{CH} and the number of neighbours for each node. The minimum value for P_{CH} is set to a predetermined threshold P_{min} to ensure a constant time complexity of the algorithm. The equation of P_{CH} is:

$$P_{CH} = \max \left\{ \frac{E_{rem}}{T \times E_{rate}}, P_{min} \right\},$$

where T is a constant that indicates the upper bound of the duration for which a node can continuously work as a CH. The value of T should be chosen such that the maximum energy of a node in the WSN is not greater than $T \times E_{rate}$.

Table 1 presents a pseudo-code that shows the operation details of EDED.

Table 1: EDED clustering algorithm.

Initialization

```

 $S_{Neighbor} \leftarrow$  nodes lie inside my radio range
myCH  $\leftarrow$  NULL
myBackup  $\leftarrow$  NULL
n := number of my neighbors

```

c := myID

Begin

```

// for unclustered node
while (myState is Initial)
    Compute probability  $P_{CH}$ 
    Broadcast a clustering initiation message

```

```

message := wait_for_a_message()
if (message.command is "survey")
    uf := calculate_number_of_followers(myID)
    send(("report",uf,  $P_{CH}$ , myID),message.node_id)
if (message.command is "recruit")
    myCH := message.node_id
    myState := Ordinary node
if (message.command is "notify" and message.node_id is myID)
    myState := Cluster_Head
    broadcast(("recruit", ,myID))

// for ordinary node
while (myState is Ordinary)
    message := wait_for_a_message()
    Neighbours := NONE // array for follower nodes
    if (message.command is "survey")
        uf := update_my_followers(myID)
        if(Neighbours is not NONE)
            send(("survey", uf,  $P_{CH}$ , myID), Neighbours)
            msgQueue := wait_for_followers_reports()
            nodeBest := SelectCH(msgQueue)
            nodeBackup := SelectBackup(msgQueue)
            message :=
            (message.command,message.data,myID)
            send(nodeBest,myCH)
            purge(msgQueue)
        else
            send(("report",NONE,NONE),myCH)
            terminate()
    if (message.command is "notify" and message.node_id is myID)
        myState := Gateway
        nextCH := nodeBest.node_id
        send(message,nodeBest.node_id)

// for cluster head
while (myState is a Cluster_Head)
    message := wait_for_a_message()
    if (message.command is "survey")
        broadcast_to_followers("survey", ,myID)
        msgQueue := wait_for_followers_reports()
        clusterBest := SelectCH(msgQueue)

```

```

send(clusterBest,message.node_id)
if (clusterBest.node_id is NONE) terminate()
    purge(msgQueue)
    if(message.command is "notify" and message.node_id is
clusterBest.node_id)
        broadcast_to_followers(message)
// for Gateway
while (myState is a Gateway)
    message := wait_for_a_message()
    if (message.command is "survey")
        message.node_id = myID
        send(message, nextCH)
    if (message.command is "notify")
        send(message, nextCH)
    if (message.command is "report")
        send(message, myCH)
    if(message.node_id is NONE)
        teminate()
End

```

The clustering process is initiated by the base station. Every node, whose status is unclustered, waits for a message. The base station broadcasts 'survey' message to its neighbors. Every node, which receives 'survey' message from its cluster head (include base station), investigates that how many unclustered nodes exist within the area of its communication range. If there are no existing nodes that can communicate with, then it reports it to their head and terminates its algorithm. If some nodes exist, it send 'survey' message to every follower and waits for its 'report' messages. Every clustered node, which receive 'notify' message, compares 'notify. node_id' with saved id and if it is same, then it changes its status as 'Gateway' and set its next clusterhead as saved node id, and sends a 'notify' message to its next clusterhead. If clusterhead received a 'notify' message, then it compares 'notify.node_id' with stored ID and if it is same then it broadcasts otherwise just drop it. If unclustered node received 'notify' message then it changes its status as cluster-head and broadcasts a 'recruit' message to its followers to make a cluster with it.

In the procedure for selecting the CH (see Table 2), each node waits for the neighbor's announcements to know CH candidates that are advertised on the basis of P_{CH} and the number of neighbors. A CH-broadcast message contains three parameters: the node ID, the number of neighbors, and the status of the CH. The status can be either final or tentative. The status is tentative if $P_{CH} < 1$; otherwise, it is final. A tentative CH may switch to ordinary node, if it finds a neighboring CH with lower cost. The messages received from the neighbors are not rebroadcast. Finally, S_{CH} becomes the set of CHs that are in the final status, and the node with the lowest cost and more neighbors is chosen as the designated CH for the ordinary node. If S_{CH} is empty, the node broadcasts a message advertising itself as a CH with final status.

Table 2: Pseudo-code describes the procedure for selecting the ClusterHead.

```

Procedure SelectCH( $S_{CH}$ ) {
    while ( $P_{CH} < 1$ )
        if ( $S_{CH}$  is not empty)
            if (myID = Max-Nbr-Neifghbors( $S_{CH}$ ))
                send(("report",uf,  $P_{CH}$ ,
myID),message.node_id,TENTATIVE)
            else if (random(0, 1)  $\leq$  PCH)
                send(("report",uf,  $P_{CH}$ , myID),message.node_id,
TENTATIVE)
             $P_{CH} \leftarrow P_{CH} \times 2$ 
        endwhile
    Remove those CHs from  $S_{CH}$  which have not still declared
themselves in the FINAL status
    if ( $S_{CH}$  is empty)
        myCH  $\leftarrow$  myID
        send(("report",uf,  $P_{CH}$ , myID),message.node_id,
FINAL)
    else myCH  $\leftarrow$  Max-Nbr-Neifghbors ( $S_{CH}$ )
        joinCluster (myID, myCH)
    }

```

The backup is determined from the information that a sensor collected at the end of selection phase of the cluster head (see table 3). The preferable backup of an ordinary node is another CH within cluster range in case of gateway node, if any. Otherwise, the node chooses one of its neighbors with more energy remaining and whose CH is different from its own CH. If no such neighbor is found, the node picks up a neighbor with more energy remaining whose CH is the same as its own CH, but whose backup destination is not the same as its own CH. If no neighbors meet these conditions, the node represents itself as the backup. A node can be either a CH or the backup for itself but not both. At the end of the algorithm, each node broadcasts a "FINISH" message informing its CH and the backup to other nodes.

Table 3: Pseudo-code describes the procedure for selecting the backup node.

```

Procedure SelectBackup ( $S_{CH}$ )
    if (myCH  $\neq$  myID)
        if ( $|S_{CH}| > 1$ )
            myBackup  $\leftarrow$  2nd Max-Nbr-Neifghbors ( $S_{CH}$ )

```

```

else myBackup ← randomly select a neighbor which is
in the FINISH state and whose cluster head is not equal to
myCH

if (myBackup = NULL)

    my Backup ← randomly select a neighbor which is
in the FINISH state and whose backup is not equal to myCH

if (myBackup = NULL)

    myBackup ← myID

send(("report",uf, PCH,
myID,myBackup),message.node_id)

```

4. SIMULATION

In this section two self-organization protocols are compared DED [12] and the proposed algorithm EDED. The simulated results are obtained to evaluate the performance of EDED using a C/C++ language-based event-driven simulator [13, 14]. We use the same simulation model as in [12] for comparison purposes. We simulated the energy consumption, number of clusters and resulting lifetime of the network.

The results obtain in terms of three metric: energy consumption, number of clusters and life time of WSN. The performance metrics used for these protocols are:

- Network Lifetime: There are multiple definitions available for network lifetime [15, 16]. It is the time interval from the start of the process until the first and last node dies;
- Stability Period: this is the time interval from the start of the process until the death of the first alive node;
- Instability Period: this is the time interval from the death of the first alive node until the death of the last alive node
- Number of alive nodes per round: this is the real time measure reflects that the total number of alive nodes per round that have not used all of their energy.

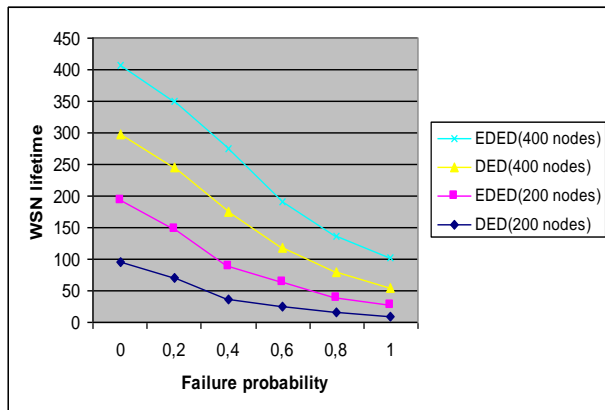


Figure 3: Wireless sensor network lifetime according to failure probability

Figure [3] shows that the lifetime of WSN is prolonged significantly with EDED, as high 50% of the lifetime in DED. In DED, the selection CHs is based on energy remaining witch geanerated more clusters. Consequently, the clustering process in DED is more expensive. However, EDED select the CHs with more neighbors. Thus, the efficiency of EDED clustering protocol helps it to perform better.

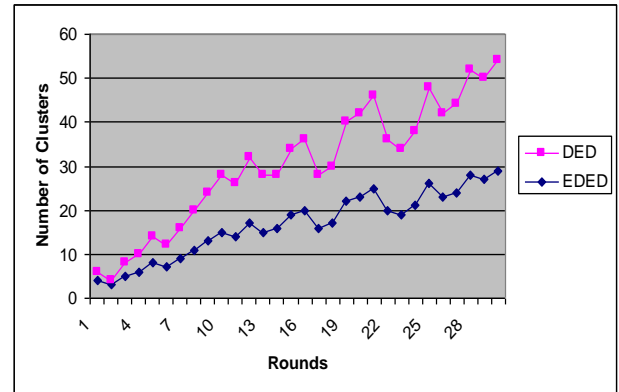


Figure 4: Number of Clusters

Fig [4] shows the distribution of the number of clusters in EDED and DED for 30 rounds. It shows that the number of clusters in EDED is much fewer than DED, total number of CHs for every round decreases significantly from that of DED.

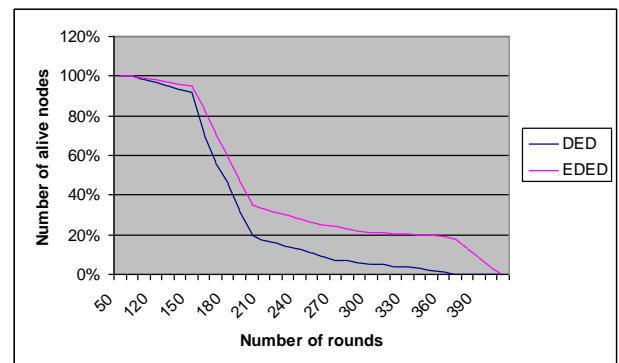


Figure 5: Number of nodes alive nodes during rounds.

Figure [5] shows rate of nodes in network which are alive with number of rounds. In these results, we see that EDED out performs DED in instability period, the number of alive nodes of EDED is higher the 8% in comparison with DED. However, There is very little difference between stability period of DED and EDED. From our simulations, we clearly see that EDED has largest network lifetime witch is more 10% more then DED.

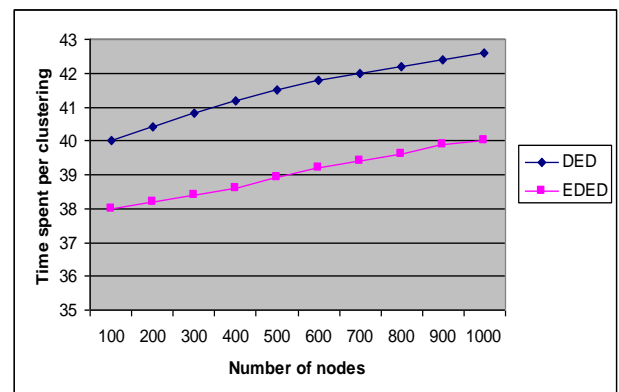


Figure 6: Time spent per clustering

Figure [6] show that the EDED algorithm reduces the time spent by clustering a significant manner which is 12%. DED spends much higher time than EDED due to the need for higher number of CHs nodes, thus a higher number of message communications. The reason for this behavior is that the EDED algorithm is executed in an efficient manner which generates a small number of clusterhead and subsequently clusters. Thus, EDED increases the lifetime of the network.

5. CONCLUSION

This paper presented a novel algorithm called EDED that solves the problem of energy and cluster-head having low energy using the clustering approach which assumes the role of the clusterhead in the wireless sensor network. The proposed algorithm further enhances the network lifetime by distributing the cluster-heads according to the energy remaining and number of neighbors. It uniformly divides area of cluster area for the WSN and save the energy dissipation of sensor nodes in the cluster. Computer simulation results showed that the proposed approach allows much longer lifetime of wireless sensor network than the existing schemes particularly DED, it is more important to balance the energy consumption of the sensor nodes and provide an efficient coverage of network than to minimize network-wide total energy consumption of the sensor nodes. Most recent works on self organization problem are still limited to theoretical study. In future, it is planned to study the impact of sensor coverage, connectivity and energy on QoS applications with WSNs.

6. REFERENCES

- [1] J. N. Al-Karaki and A. E. Kamal, Routing techniques in wireless sensor networks: a survey, In *IEEE Wireless Communications*, Volume 11, pp. 6-28, 2004.
- [2] I. Demirkol, C. Ersoy, and F. Alagoz, MAC Protocols for Wireless Sensor Networks: A Survey, *IEEE Communication Magazine*, vol. 44, Issue 4, pp.115-121, 2006.
- [3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energyefficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd Annual Hawaii International Conference on System Sciences (HICSS'00)*, Hawaii, USA, Jan. 2000, pp. 3005–3014.
- [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(4), pp. 366–379, Oct- Dec. 2004.
- [5] M. H. Tolou and J. Chitizadeh, "Lifetime prolonging of wireless sensor networks via a recursive clustering algorithm," in *Proc. Of the third IEEE International Conference in Central Asia on internet the Next generation of mobile,wireless and optical communications networks (IEEE/IFIP ICI'07)*, Tashkent, Sep. 2007, pp. 1–6.
- [6] M. Ye, C. Li, G. Chen, and J. Wu, "Eecs: An energy efficient clustering scheme in wireless sensor networks," in *Proc. of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC'05)*, Phoenix, Arizona, USA, Apr. 2005, pp. 535–540.
- [7] J. Kamimura, N. Wakamiya, and M. Murata, "Distributed clustering method for energy -efficient data gathering in sensor networks," *International Journal on Wireless and Mobile Computing*, vol. 1, no. 2, pp. 113–120, 2006.
- [8] J. Yu, W. Liu, J. Song, and B. Cao, "Eemr: An energy-efficient multi-hop routing protocol for wireless sensor networks," in *Proc. Of the International Conference on Computer Systems and Application (IEEE/ACS AICCSA'08)*, Doha, Qatar, Mar. 2008, pp. 291–298.
- [9] Md. Mamun-or-Rashid, M. Mahbub Alam and C. Seon Hong, "Energy Conserving Passive Clustering for Efficient Routing in Wireless Sensor Network,"
- [10] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, 2002.
- [11] K. Shin, A. Abraham and S. Yong Han "Self Organizing Sensor Networks Using Intelligent Clustering," School of Computer Science and Engineering, Chung-Ang University 221, Heukseok-dong, Korea.
- [12] Mohammad M. Hasan and Jason P. Jue, "Survivable Self-Organization for Prolonged Lifetime in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, Volume 2011, pp. 1-11, 2011.
- [13] D. Curren, "A survey of simulation in sensor networks," 2006, <http://www.cs.binghamton.edu/~kang/teaching/cs580s/>.
- [14] G. Chen, J. Branch, M. Pflug, L. Zhu, and B. Szymanski, "SENSE: a wireless sensor network simulator," in *Advances in Pervasive Computing and Networking*, chapter 13, Kluwer Academic, Boston,Mass, USA, 2004.
- [15] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, 2006.
- [16] Q. Wang, H. Hassanein, and G. Takahara, "Stochastic modelling of distributed, dynamic, randomized clustering protocols for wireless sensor networks," in *Proceedings of the International Conference on Parallel Processing Workshops (ICPP '04)*, pp. 456–463, August 2004.