Analysis of energy efficient Self-organization scheduling Scheme for Ad Hoc WSN

Birendra Kumar*, Dr. K P Yadav*

*PhD Scholar, Department of CSE, CMJ University, Shillong
*Director, SIET, Ghaziabad

Abstract—Networks considered in this paper consist of tiny energy constrained commodity sensors massively deployed in an area of interest. In a wireless sensor network, energy conservation is the primary design goal. This paper describes an Energy Efficient self Organization Scheduling Scheme (EEOSS), a simple event-driven and leader-based self-organization. It provides and maintains a communication structure for wireless ad hoc sensor networks with low energy consumption. To achieve low power operation, EEOSS employs a quick join procedure with low message cost for newly arrived sensor nodes. EEOSS guarantees a low-cardinality dominating node set selection in the network. The resulting communication structure can adapt itself with the growth of the network service area. EEOSS is a completely localized algorithm. Through analysis, it can be shown here that EEOSS provides a way to extra energy savings in compared to other self-organization schemes when using the same distributed MAC scheduling Networks considered in this paper consist of tiny energy constrained commodity sensors massively deployed in an area of interest. In a wireless sensor network, energy conservation is the primary design goal. Here a method is proposed to manage the spontaneous organization of sensor activity in ad hoc wireless sensor network systems. The small wireless sensors exchange messages to coordinate responses to requests for sensing data, and to control the fraction of sensors which are active. This method can be used to manage a variety of sensor activities. In this way, one can use it for reducing the power consumption by battery operated devices only when low resolution sensing is required. This in turn increases the operation lifetimes.

Keywords — Network, WSN, Lifetime, EEOSS, Self organisable, Leader, Member.

I. INTRODUCTION

Unlike their wired networks, most types of wireless networks are rapidly deployable, scale well, and are cost-effective. Recent advances in nanotechnology has made it technologically feasible and economically viable to develop a large variety of Micro Electrical-Mechanical Systems (MEMS) — miniaturized low-power devices, referred to as sensor nodes, that integrate sensing, special-purpose computing and wireless communications capabilities. Individual sensor nodes have a non-renewable power supply and, once deployed, must work unattended.

Wireless ad hoc sensor networks are spontaneous networks formed by sensor nodes with limited information processing capabilities and very constrained energy resources. The fundamental goal of a sensor network is to produce, over an extended period of time, global information from local data obtained by individual sensor nodes. Wireless sensor networks process data gathered by multiple sensors to monitor events in an area of interest.

As the surrounding environment changes, sensor nodes must adjust their operations to keep the network connected and operational. Self-organization schemes help nodes to deal with these spontaneous events in a collaborative fashion. Self organization aims to structure the network, taking into account nodes' intrinsic properties as well as local interactions among nodes, with the goal of providing efficient communication [1], [7], [14].

In order to provide low energy schemes, self-organization should be coupled with time scheduling. More precisely, they are randomly placed in space and randomly activated in time. Self-organization should deal with this random deployment. A novel self-organization protocol for wireless sensor networks is presented here. To meet the above-mentioned goals, we propose EEOSS, an event-driven self organization scheme. It can work with an independent, distributed MAC scheduling mechanism, and outperforms some existing self-organization schemes in terms of energy savings under same conditions.

II. SENSOR NODES AND NETWORK MODEL

Wireless ad hoc sensor networks are different from traditional ad hoc networks in terms of individual node capability, energy resource, and the amount of nodes in the network and deployment method. Some properties on the sensor nodes are assumed here. All sensor nodes have similar capabilities (processing and communication). This eliminates the centralized approach. No node could permanently play an important role. Role is neither in the communication structure nor in the information processing. All nodes have similar energy storage and once the sensor nodes are switched on, the batteries will not be changed frequently or ever. Energy efficiency is therefore the primary requirement in the scheme design. Each sensor initially is in sleep mode most of the time, waking up at random points in time for short intervals under the control of a watchdog timer. Each sensor...
has a modest transmission range; perhaps a few meters. The sensor nodes are location unaware. Limited by cost, energy and processing capabilities, nodes do not have GPS. Current signal strength aware distance measurement technologies are not applicable in large scale networks because of interference. Therefore, location based techniques are excluded in our consideration. Some properties on the network model are assumed here. The communication links are symmetric. Although radio environment may cause asymmetric links, the symmetry of links is the common assumption for wireless network protocol design. There is no pre-defined strategy for the deployment of a wireless ad hoc sensor network. The nodes are randomly distributed in the given field. The sensor nodes are randomly switched on.

III. SELF ORGANISATION

Here we are interested in providing a simple, low energy cost and scalable self-organization which allows the sensor nodes to form a network structure and self-adapt to spontaneous topology changes. It should meet the requirements given here. The virtual structure emerges only from decisions taken in a local context by nodes in the network. In order to provide energy savings, periodical Hello message broadcast should be avoided. The self-organization should be efficient in terms of processing and message complexity when reacting to changes in the environment. No assumption is made about the homogeneity distribution of nodes in the area. All nodes randomly arrive in the service area on random locations. Sensor nodes execute EEOSS independently. The organization do not execute by round. No inter-node synchronization is required

IV. EEOSS PROTOCOL MODEL

EEOSS is an event-driven self-organization scheme. It is different from the schemes in which each node periodically broadcasts a Hello message and every node must also continuously monitor its neighbourhood. It is very costly in terms of energy consumption. In order to eradicate this periodical control traffic and continuous monitoring of nodes, EEOSS selects only a set of nodes to provide neighbouring information. These nodes play ‘active’ roles (similar to dominators) in the network, while the others are ‘passive’ nodes taking no action as long as no event is reported to them. One major difference between EEOSS and other cluster based schemes is that the organization need not be renewed in each round. The maintenance of the communication structure is event-driven. Three roles are defined in EEOSS: Leader, Gateway and Member. A Member is a ‘passive’ node in the organization which interacts only with its Leader in most cases. It is a one-hop neighbour of its Leader. A Leader is in charge of all the communications of its Members. No Leaders are connected together directly in the network. A Gateway interconnects the Leaders which are separated with two-hop distance. Leaders and Gateways form a virtual backbone (see Fig. 2) in the network. Only Leaders send periodical Leader Broadcast Message to let others know that it is in charge of the region.

![Figure 2: The virtual structure generated with EEOSS (EEOSS Snapshot and Virtual Backbone)](image)

The events related to changes in the environment fall into the following categories:
1) Join Events
2) Leave Events
3) Partition management
4) Local re-organization

We believe that the behaviour of the EEOSS after the occurrence of each event has direct impact on the quality of the organization. The design of EEOSS tries to provide optimal procedures to deal with these events. It is also worth noting that EEOSS is a completely localized scheme: each node independently decides its state using only local information.

We consider a fixed-size 2D space where nodes are randomly deployed in space and randomly activated in time. From the moment when the radio transceiver of a node is activated, it is considered as appearing in the network. All nodes have identical radius for both broadcast transmission and unicast transmission. Almost all self-organization schemes are designed for running on connected networks only, while in our consideration the network is not necessarily connected at all times. It is due to the random nature of node arrivals. More than one isolated partition is possible depending on the locality of nodes. That is the reason that EEOSS adopts partition management procedures in addition to join and leave procedures. Another appropriate mechanism is local re-organization, which allows active nodes (Leaders and Gateways) to switch to passive nodes (Members) in order to consume less energy. It can also bring an optimization to the virtual structure in the organization.

1) Join procedures: A newly arrived node will take at most three stages to integrate into the network organization (Fig. 3). It waits for a Leader Broadcast Msg. By hearing the message, it engages in a one-hop attachment procedure directly with the Leader using unicast messages. It becomes a Member. If no Leader is detected, it sends a one-hop broadcast Member Solicitation Msg. The latter is used to look for a configured node in its radio vicinity. Upon receiving a Member Advertisement Message, it engages in a two-hop attachment procedure with one of its neighbouring Member and the Leader of this Member. After the attachment, the newly arrived node becomes a new Leader. The neighbouring Member changes itself into a Gateway in order to connect the new Leader to the existing Leader. The two-hop attachment leads to an expansion of the communication structure in the target area. If no neighbour is detected, it assigns itself as a First Leader and generates a random partition identifier. Hence, a new partition appears in the service area.

![Figure 3: Joining procedure illustration](image)

The design of the joining procedure in EEOSS shows the full use of existing organization for newly arrived nodes. EEOSS lets each
newly arrived node discover the network structure within its neighbourhood (i.e. the presence of one hop Leaders and Members) first. Then it integrates itself into the structure by using only unicast messages.

2) Leave procedures: Since nodes with different roles have different impacts on the communication structure, there are obviously three distinct leave events. Basic EEOSS does not provide detection of an abrupt departure of a node Member. This is not harmful to the network because a Member has minimal influence on the network organization. Only the messages sent to the disappeared node are lost, which in any case cannot be avoided. However, its Leader may launch an active scan after realizing that several messages sent to this Member were lost. And it finally detects the Member’s departure. If a Gateway node leaves in a normal condition, it informs each Leader connected to it by unicast message. However, if it abruptly leaves the network without any notification, the departure can be detected by a ‘Leader-Gateway’ scan mechanism. EEOSS forces each Gateway to reply to the Leader Broadcast Message using unicast message. A Leader sends a solicitation to find the Gateway if it does not receive the Gateway reply. It finally concludes that the Gateway is gone if no alert is received. Each Leader stocks the address of its 2-hop Leaders connected by its Gateway. They will ask all their Members to enter sensing mode. A Member which receives a Leader Broadcast Message from the Leader with the lowest address will become a Gateway candidate. (See Fig. 4).

A Leader leave event occurs on the Members and Gateways which have not received Leader broadcast Message for k times. k is an adjustable parameter of EEOSS. If k is small, then a departure will be quickly detected. But some false detection might be alerted due to bad reception. If k increases, the false detection occurs less, but the latency of the detection increases. Therefore the value of k depends on the requirement of running applications. After detecting the Leader departure, Gateways originally connected to the lost Leader becomes Members, while all Members detecting a Leader departure enter a re-organization phase. They finish by either attaching themselves to another Leader or becoming a new Leader (see Fig. 5).

3) Partition management: Unlike other self-organization schemes, EEOSS allows for the existence of multiple partitions in the network. Therefore the management of these partitions is a key issue. The topology changes conduct to partition splitting, partition detection and partition merge. Partition splitting: The key point in partition splitting is to maintain a virtual structure for communication in each partition. In EEOSS, Leaders and Gateways are the ‘active’ nodes, while all Members are ‘passive’ nodes. A Member takes no action as long as it is still covered by its Leader. If a Member loses its Leader, it will take the same actions as a newly arrived node and re-integrate into a new partition.

The nodes which were Leaders before partition splitting remain Leaders in each partition. Some local re-organizations might occur in each partition to keep a connected virtual structure. After a partition splitting, if a Leader loses any of its Gateways, it should generate a new Partition ID and inform every Leader in the new partition via the virtual backbone. This may bring about message overheads during partition splitting, but it will ensure the detection of other partitions for merging issues. Partition detection: At the beginning of network deployment, many partitions might be generated in the service area due to the random location of nodes. Each partition grows along with the joining of new nodes (especially when it should act as a Leader). The partitions will finally meet each other. EEOSS lets a newly arrived node detect different partitions around it during its joining procedure. A partition is identified by its Partition ID which is generated by the First Leader using a random number generator. The number of partitions in a service area is not comparable to the number of nodes in terms of cardinality. Therefore, if the number of bits reserved for the Partition ID is large enough, the probability of Partition ID conflict is small enough to be negligible. This assumption is made by others [16] as well. The partition ID is contained in Leader Broadcast Msg. A node which receives multiple Leader Broadcast Message can identify if there are several partitions in its neighbourhood. Partition Merge: The node detecting several partitions will inform the Leaders of each partition in its neighborhood. It will integrate itself into the partition containing the most Leaders (say Big partition: the number of Leaders is necessarily included in the Leader Broadcast Msg), acting as a new Gateway to interconnect different partitions. It also informs both partitions to update partitionId and the number of Leaders, on every Leader.

4) Local re-organization: In order to achieve a long network lifespan, a node in the network does not conserve its role for its whole life. In EEOSS, Leaders and Gateways use a local metric to judge if they should let some other Members take on their roles. This technique is known as active scheduling introduced earlier in [13] for broadcast mechanism. It helps to distribute the communication energy cost more equally among all nodes. The active scheduling will surely result in a local re-organization.

V. ANALYSIS

We have implemented EEOSS as a network layer protocol. Each node uses the same radio technology with identical radio
transmission power. The positions of the nodes are generated by a random generator. Nodes switch themselves on in random order an independent distributed MAC layer scheduling mechanism, is also implemented in order to analyze the energy gain using EEOSS as compared to other organizations (e.g., LMST and CDS). We evaluate the pure protocol cost bring by EEOSS, therefore no data traffic is generated in the network the ratio of nodes in the virtual structure of EEOSS decreases when the number of nodes in the network increases. This shows that the number of Leaders and Members increase slower than a O(n). In EEOSS, the nodes which arrive later in the network have a high probability of becoming a Member. The nodes that arrive later in the network bring about fewer and fewer changes on the existing virtual backbone. We are also interested in local structures around each Leader. Since the number of Leaders in the network grows very slowly, we observe that the maximal and average number of Members per Leader increase in a O(n) fashion. This observation is also related to the characteristic of the cardinality of Leaders in the network. We do not assume any synchronization in the network. We consider that the organization begins with the arrival of the first node and ends with the departure of the last nodes. Therefore, we can only evaluate the latency of each procedure. The average latency decreases when the number of nodes increases. This is due to the fact that nodes that arrive later are most likely to become a Member, which requires a shorter configuration time than for a new Leader. The unicast message cost per node decreases very slowly and tends to stabilize with the increase in cardinality of the network. It coincides with our analysis on message complexity of EEOSS which is a O(1). The same observation is made on the average number of messages per Member. Furthermore, the message cost is stable when the number of nodes increases, since the number of Leaders is stable in the network. Low-energy consumption is EEOSS’ most important feature; it is also the initial objective of our protocol design. As explained previously in Section I, MAC scheduling should be integrated with self-organization to provide energy savings. We adopt BMAC [9], a CSMA/CA protocol with distributed sleep scheduling for each self-organization. We are not interested in the energy savings brought by BMAC, but the additional energy savings when using EEOSS compared to other schemes. This illustrates that the percentage of energy savings of EEOSS is density invariant.

VI. CONCLUSIONS
In this paper, we propose an event-driven, Hello message free, low-energy self-organization scheme (EEOSS) for wireless ad hoc networks. It effectively performs self-organization, using local decisions to deal with random deployment and changes in the environment on a large scale. We show that it may provide an extra energy savings compared to the schemes using periodical Hello messages. This result puts forward the weakness of the traditional approach to self-organization and the need for simple, low-energy self-organization schemes.

REFERENCES