Wireless Sensor Network Infrastructure : Construction and Evaluation

Kamal Beydoun, Violeta Felea, Hervé Guyennet

University of Franche-Comté
LIFC, 16 route de Gray
25030 Besançon
{Kamal.Beydoun, Violeta.Felea, Herve.Guyennet}@lifc.univ-fcomte.fr

Abstract— **Large area wireless sensor deployments rely on multi-hop communications. Efficient packet transmissions and virtual topologies, which structure sensor networks, are two main features for efficient energy management in wireless sensor networks. This paper aims to present a distributed and low-cost topology construction algorithm for wireless sensor networks, addressing the following issues: large-scale, random network deployment, energy efficiency and small overhead. We propose structuring nodes in zones, meant to reduce the global view of the network to a local one. This zone-based architecture is the infrastructure used by our hierarchical routing protocol. The experimental results show that the proposed algorithm has low overhead and is scalable.**

Keywords- WSN, zone partitioning, energy management, scalability

I. INTRODUCTION

Technological advances in microelectronic and wireless communications have enabled environmental monitoring using small sensor devices grouped in new types of networks called wireless sensor networks (WSNs). Space exploration, vehicular movement monitoring and critical events detection are some examples of wireless sensor network applications. Sensor nodes are equipped with both sensory devices allowing data sensing and wireless transceivers that help them communicating. When detecting a stimulus, sensor nodes (called sources) generate data packets and transmit them through the network to one or several special nodes (called sinks). Direct communications would be possible if large transmission power on the transmitter node was used. However, in largely deployed networks, high transmission power would not be enough to reach the sink and would consume lots of energy. This problem can be overcome by multi-hop communications, sensors playing thus a double role: data generator and data router.

Routing protocols providing this path construction should fill some requirements concerning energy efficiency, distributed-based algorithmics and scalability. Among different routing techniques, table-based routing (inspired from ad hoc networking) exploits communication costs (generally in terms of expended energy), which are required to reach the sink via a given neighbor. These protocols are a trade-off between path availability and the overhead due to the construction of the routing tables themselves. Moreover, in WSNs, routing protocols should be designed under the constraints of limited memory and large-scale node deployments.

In wireless, mobile and multi-hop networks, routing protocols should be able to deal with random node deployment. It means that even though sensors' positions are known (manually deployed) no particular hypotheses concerning its neighbors can be done due to the large scale (neighborhood discovery protocols need to be implemented). Therefore, sensor networks are considered as a subclass of ad hoc networks because of the absence of infrastructure. Thus, ad hoc networking may influence some routing approaches in wireless sensor networks, in respect to topologies. In hierarchical structures, topology control can be applied to minimize the set of active nodes (switching off some of them to preserve energy) or to define coordination tasks for some particular nodes. We are interested in hierarchical structures because flat architectures generally depend on the size of the network, which makes routing approaches difficultly scalable. In either approach, an important issue that needs to be addressed is the most crucial aspect: the energy efficiency.

Our contribution to the topology construction is a hybrid solution: organizing sensors in zones (virtual architectures) at a low cost. For this purpose, an inexpensive neighborhood discovery algorithm is proposed. The idea is to distribute routing roles between nodes inside a zone, avoiding cluster management (including cluster head election and rotation, cluster construction as in classical hierarchical approaches). The paper is organized as follows. In the next section, we present some of the solutions in literature concerning the construction of virtual network structures for WSNs. Next, we give our zone partitioning algorithm. Then, we compare it to some existing partitioning algorithms. We also show its evaluation through simulations. Finally, the last section concludes the paper.

II. RELATED WORK

Energy consumption is one of the main challenges in wireless sensor networks. Energy saving assures a long lifetime for the system. Another main goal is reducing the size of the stored data (e.g. routing table) in each node of the network. Clustering is an important technique for prolonging the system lifetime and reducing the size of the stored data. In clustering, nodes are gathered in several groups, generally disjoint, which are named clusters. Each cluster has a cluster

head (CH). The nodes collect data and send it to the CH that forwards this data to the final user or Base Station (BS). CHs can communicate with the Base Station directly or via other CHs. There are many existing clustering protocols. LEACH [1] is a distributed clustering-based protocol that uses randomized rotation of the CHs to evenly distribute the energy load among the sensors in the network. LEACH assumes that the fixed sink is located far from the sensors and that all sensors in the network are homogeneous and battery-constrained. Lin's protocol [2] is a distributed clustering technique for large multi-hop mobile wireless networks. The cluster structure is controlled by the hop distance. In each cluster, one of the nodes in the cluster is designed as cluster head. Other nodes join a cluster if they are within a predetermined maximum number of hops from the cluster head. HEED [3] is a distributed clustering protocol that periodically selects cluster heads according to a hybrid function between their residual energy and a secondary parameter, such as node proximity to its neighbors or node degree. In CES distributed protocol [4], each sensor computes its weight based on the k-density, the residual energy and the mobility features. Then it broadcasts the weight to its 2-hop neighborhood. The sensor node having the greatest weight in its 2-hop neighborhood becomes the cluster head and its neighboring sensors will join its cluster. SPAN [5] is a distributed, randomized protocol in which nodes make local decisions on whether to sleep, or to join a coordinator that rotates at times. Each node makes its decision depending on the amount of available energy on the node and on its degree (the number of its neighbors when the node is active). SPAN is a protocol that operates under the routing layer and above the MAC and physical layers. The routing layer uses information SPAN provides, and SPAN leverages any power saving features of the underlying MAC layer [5]. The centralized PEGASIS protocol [6] constructs chains instead of clusters. Each node delivers the sensed data to the nearest neighbor node. One sensor node on the chain is assigned as the cluster head node that delivers sensed data to the base station. The head node is selected by turns; this technique allows to consume energy evenly in wireless sensor networks. However, the PEGASIS protocol causes redundant data transmissions since one of the nodes on the chain is selected as the head node regardless of the base station's location. In [7], authors propose the enhanced PEGASIS protocol based on the « concentric clustering » scheme to solve this problem. It means that clusters have the shape of concentric circles. Similar to PEGASIS, the SHORT protocol [8] adopts centralized approaches and requires powerful BS to take the responsibility of managing the network topology and to calculate the routing path and time schedule for data collection.

Most topologies based on clusters assume that cluster heads are high-energy nodes and their transmission power can be adapted in order to reach the base station at far distances and to communicate directly to other cluster heads. Another assumption is that nodes within a cluster can directly communicate to the cluster head. In SHORT, HEED, CES, PEGASIS and Enhanced PEGASIS all nodes are supposed to have the ability to modify the transmission power in order to control topology. The LEACH radio model [1] is used for these protocols. Requirement of adaptive and dynamic transmission power modification can be prohibitive, especially for sensors not equipped with transmission amplifier. SPAN uses the radio model of the Cabletron Roamabout 802.1 card that has fixed transmission range and does not support power control. Lin's protocol does not mention the radio model used for simulations.

The transmission defines the set of neighbors for a sensor node, those able to receive the transmitted signals. Because variation of the transmission range consumes more resources, virtual topologies should be proposed for sensor networks that are made of sensors with fixed transmission power. The challenge addressed in this paper presents an approach of virtual structuring of networks without using topology control technique. Our contribution to topology construction addresses two main issues in WSNs: distributed approaches, and energy efficiency. Moreover, our approach is independent of the embedded sensor technology (being able to vary the transmission power or not); the only parameter considered is the current node's transmission range. The algorithm is executed simultaneously with the neighborhood discovery protocol for random sensor node deployments. In the next section, we will detail our work for structuring wireless sensor networks into zones, which is not a real clustering algorithm like the cited related work. Therefore, no cluster heads exist in our topology; no other information on the network (e.g. geographic position) is required.

III. VIRTUAL WSN STRUCTURING: ZONE PARTITIONING ALGORITHM

Virtual topology construction for wireless sensor networks should deal with the following challenges: large-scale and random network deployment, energy efficiency and small overhead. These are the issues addressed by our algorithm of WSN virtual structuring, achieved through node grouping in multiple zones. As illustrated in Figure 1 the proposed algorithm tends to divide a wireless sensor network in several zones and to save useful information for routing. This zone construction enables intra-zone view for every node, which reduces complexity compared to a global network management.
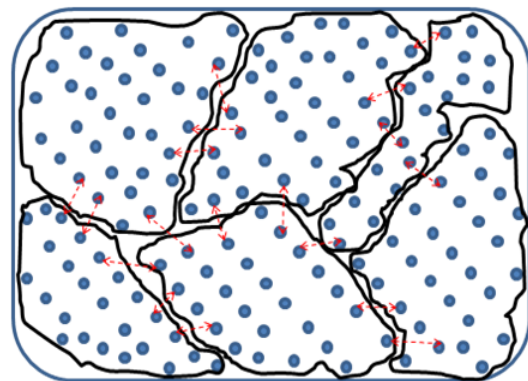

Figure 1. Zones in a Wireless Sensor Network

The nodes in the same zone can communicate within one or more hops. After structuring the network, the nodes at the border of each zone route data between these zones (dashed rows in Figure 1).

We consider the general case of randomly-deployed nodes. In our approach, there are some network nodes named *inviting nodes* that launch the partitioning algorithm. These nodes are designed randomly because we assumed that nodes in the network have no information either about its neighbors (e.g. geographic positions) or about the network in order to elect the *inviting node* in a distributed way. In addition, election protocols in large networks are expensive in terms of number of control packets and energy consumption. Note that the *inviting nodes* do not have any additional role like the cluster heads in the clustering protocols, so there is no need to execute an expensive protocol to elect the *inviting nodes*. The ideal solution is to choose the *inviting nodes* such that the distance between them is maximal in order to avoid closely placed *inviting nodes*. Therefore, the problem that could intervene in random selection is to have geographically close *inviting nodes*. In order to justify our assumption, we have calculated the probability of finding an *inviting node* in the communication range of another one. For that, we have simulated a network of sensor nodes that are distributed randomly. These nodes include several special nodes (*inviting nodes*). We varied the network size from 400, 500 nodes (low-density network), to 600, 700 and 800 (high-density network) nodes. The simulations were performed on a large area of 1500m*1500m. The transmission range T_r is set to 300 meters[1]. Each simulation result in Figure 2 is an average of 1000 simulation results. We varied the number of the *inviting nodes* from 5 to 25 nodes. Figure 2 shows that the probability of finding an *inviting node* in the communication range of another one does not exceed $1.2*10^{-3}$ when considering a deployment of 800 nodes and 25 *inviting nodes*. We conclude from the negligible probability that the random distribution of the *inviting nodes* can be fairly used to partition sensor networks; moreover, this method has the advantage of being less expensive than the execution of a distributed election protocol.

A. The algorithm parameters

In our approach, we suppose that each node in the network has a global *ID, NodeId*. The partitioning algorithm depends on three parameters: **R**, **zN**, and **N**. **R** is the radius of a zone. It is the maximum number of hops between the *inviting node* and the invited nodes that will possibly join the zone. **zN** is the required number of zones. **N** is the number of nodes in the network. Each node specifies, during the algorithm execution, the following attributes: **ZoneId**, **NodeType**, and **BorderTable**. **ZoneId** defines the zone to which the node belongs. **NodeType** is the node type. It may have two possible values *NORMAL* or *BORDER*. All nodes have the *NORMAL* type at the beginning. Node may change its type to *BORDER* if it is located between two or more zones. **BorderTable** is the table in which a border node (of

---

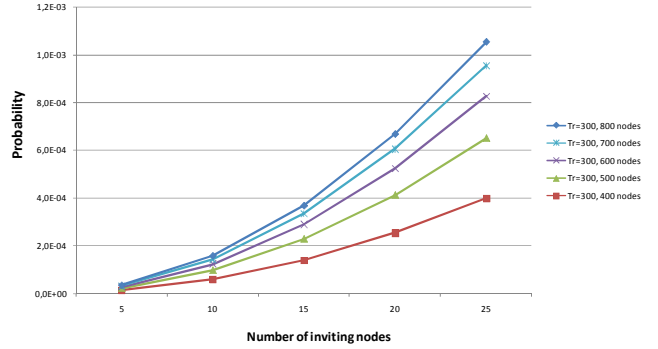*BORDER* type) saves information about the other border nodes in the neighboring zones.

Figure 2. Probability of finding an *inviting node* in the communication range of another one (T_r =300 m)

The fields of a packet exchanged during the execution of the algorithm are shown in TABLE I. Three possible values for the packet subject are used in the algorithm: *INVITATION, DISAGREEMENT, BORDER* and *NEW_NODE*. Next, we describe the use of each packet subject in the algorithm.

TABLE I.   Fields of a control packet

SrcId	Global *ID* of the node sending the packet
DestId	Global *ID* of the destination node
ZoneId	Zone *ID* of the node sending the packet
Subject	Subject of the packet
NodeType	Type of the node sending the packet
TTL	Time To Live of the packet

B. Zone partitioning algorithm description

In the proposed algorithm, information concerning nodes (such as energy, link states, geographic positions, etc...) is **not** required. The details of the algorithm executed by each network node are shown in Figure 3.

Initially (Figure 3, part **A**), the zones are formed of the *inviting nodes* (one node per zone) which broadcast an *INVITATION* packet only once to its neighbors in order to join its zone (the number of the *inviting nodes* in the network equals the number of zones **zN**). This *INVITATION* packet has an initial *TTL* value that equals the zone radius **R**. In Figure 3, part **B**: the node that receives an *INVITATION* packet treats it according to its state (assigned or not). Nodes also exchange packets having other subjects (*DISAGREEMENT* or *BORDER*) in order to specify the node type (*NORMAL* or *BORDER*) and to complete the table **BorderTable** of the border nodes. When a new node joins the network, it has to broadcast a *NEW_NODE* packet in order to be assigned to a zone. Nodes that receive this packet answer by an *INVITATION* packet to its zone. As output, the node attributes (**BoderTable**, **ZoneId** and **NodeType**) are defined. The constructed zones are disjoint: every node belongs to only one zone (the first from which it receives an *INVITATION* packet).

---

[1] twice the transmission range used by MICA2 sensors

```
INPUT : R
   PART A :
   IF (n is an inviting node, it sends the INVITATION packet P only once)
       broadcast INVITATION packet P(n.nodeId, ANY, n.ZoneId, INVITATION, NORMAL, R)
   ENDIF

   PART B : when node n receives a packet P
   IF (P.Subject = INVITATION)
       IF (n.ZoneId undefined)
           join the zone (n.ZoneId := P.ZoneId)
           IF (P.TTL = 0)
               n.NodeType := BORDER
               broadcast a BORDER packet P'( n.NodeId, ANY, n.ZoneId, BORDER, n.NodeType, 1 )
               RETURN
           ELSE
               broadcast an INVITATION packet P'( n.NodeId, ANY, n.ZoneId, INVITATION, n.NodeType, P.TTL - 1 )
               RETURN
           ENDIF
       ELSE
           IF (n.ZoneId = P.ZoneId)  RETURN
           ELSE
               n.NodeType := BORDER
               broadcast a DISAGREEMENT packet P'( n.NodeId, ANY, n.ZoneId, DISAGREEMENT, n.NodeType, 1 )
               n.BorderTable.add (P.ZoneId, P.SrcId)
               RETURN
           ENDIF
       ENDIF
   ENDIF
   IF (P.Subject = BORDER OR P.Subject = DISAGREEMENT)
       IF (n.ZoneId = P.ZoneId)  RETURN
       ELSE
           n.NodeType := BORDER
           broadcast a BORDER packet P'( n.NodeId, ANY, n.ZoneId, BORDER, n.NodeType, 1 )
           n.BorderTable.add (P.ZoneId, P.SrcId)
           RETURN
       ENDIF
   ENDIF
   IF (P.Subject = NEW_NODE)
       send an INVITATION packet P'(n.NodeId, P.SrcId, n.ZoneId, INVITATION, n.NodeType, 1 ) RETURN
   ENDIF
OUTPUT : NodeType, ZoneId, BorderTable at node n
```

Figure 3. Zone partitioning algorithm

C. Topology maintenance: node join, leave and mobility

Once the initial topology is built (either in clusters, chains or even zones as in our algorithm), a main problem needing solution is topology maintenance. Maintaining consistent view of a virtual network structure generally needs information update on neighbor nodes when nodes join and leave the network (due to failure), or/and when node mobility is involved.

In our partitioning algorithm, if a new node joins the network, it must broadcast a *NEW_NODE* packet in order to be assigned to a zone. Concerning node failure, only border nodes send regularly neighbor discovery packets (e.g. *HELLO* packet) to update their **BorderTable** tables. A normal node failure does not have any effect in the present network topology. It will be treated in our future proposed routing protocol based on this hierarchical structure.

Node mobility can be treated as a node **leave** and its **join** in a different geographical environment. This change can be reflected in our zone-based network topology in the following manner: a node moving to another location resets its attributes (**BorderTable** and **ZoneId**) and broadcasts a *NEW_NODE* packet. Nodes receiving this packet answer by

an *INVITATION* packet. The mobile node, when receiving an *INVITATION* packet, treats it according to the algorithm presented in Figure 3.

IV. DISCUSSION

In this section, we compare the proposed algorithm in the paper with some similar existing work. The aim of our zone partitioning algorithm is to organize sensor networks in local view structures in order to reduce complexity of routing protocols. The proposed zone infrastructure may be compared with other topology constructions, in particular with clustering techniques. We show in the remainder of this part several differences between the two approaches, which justify considering our zone partitioning different from clustering.

One critical aspect concerning cluster construction and maintenance is the overhead. Control messages exchanged in order to choose the cluster heads, to define the cluster to which a node belongs and to maintain consistent view of cluster may generate important overhead.

In LEACH, sensors elect themselves to be cluster-heads at any given time with a certain probability. These cluster head nodes broadcast their status to the other sensors in the

network. Each sensor node determines to which cluster it wants to belong choosing the cluster head that requires the minimum communication energy (based on the received signal strength) [1]. The clustering phase in LEACH needs several communications between different sensors in the network. In Lin's protocol [2], each node, at the beginning, is a cluster head. Depending on the distance between the cluster head nodes and on their IDs, clusters merge up to a point where there is no more fusion possible. This technique needs important communications and computation. HEED [3], CES [4] elect and re-elect the cluster heads according to nodes information (residual energy and position). Exchanging this information needs more communications between nodes, which means more energy consumption. SPAN [5] uses routing information and geographical position of nodes in order to construct its own topology. SPAN adaptively elects « coordinators » from all nodes in the network. SPAN coordinators perform multi-hop packet routing within the network. SPAN also rotates the coordinators. PEGASIS [6] and enhanced PEGASIS [7] assume that all nodes have the knowledge of the network, which requires communications between nodes. In comparison with the previous protocols, our approach diminishes the number of communications (sent and received packets) between the network nodes. Decisions are less cooperative and there is no need of any network information (geographic position, energy, routing tables, etc...) like in the distributed algorithms LEACH, Lin's, HEED, SPAN, enhanced PEGASIS and CES. In addition, these algorithms, except for Lin's, require that the node have ability to modify the transmission power in order to control topology, which is not the case in our approach. PEGASIS and SHORT, two centralized approaches, design the BS, considered more resourced, to control the network, whereas our approach is completely distributed.

V. ZONE PARTITIONING: EXPERIMENTATIONS AND RESULTS

Experiments were built upon the J-Sim simulator [9] dedicated to WSN simulations. It is a compositional, component-based simulation environment. It is built upon the concept of autonomous component programming model. J-Sim is developed entirely in Java. The signal attenuation due to **obstacles** or other factors (e.g. use of unidirectional antennas) is simulated in J-Sim. Therefore, the vicinity of a node in terms of transmission range is not necessarily **spherical**.

In order to evaluate the efficiency of the zone partitioning algorithm, we considered three metrics: the error ratio, the overhead and the scalability. Zone construction virtually affects sensor nodes to particular zones. Ideally, every network node should be member of a zone. Depending on the parameters of the algorithm and on the interconnection topology of the network, we evaluate the efficiency of our algorithm in terms of percentage of unassigned nodes (isolated nodes), defined as the error ratio. One particular evaluation metric for infrastructure construction is the generated overhead. We estimate it by both the number of sent and received packets (exchanged during the execution

of the zone partitioning algorithm) and the dissipated energy. The previous two evaluation metrics depend on the network size. Scalable zone partitioning is of main interest for large-area sensor deployments. For this reason, we also evaluate the behavior of the proposed algorithm when varying the network size.

The simulations were performed on a large area of 1500m*1500m. The node's maximum transmission range is set to 150m.
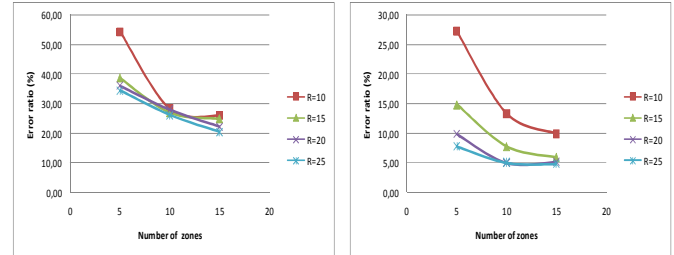

Figure 4. Error ratio for 300 nodes (left) and 400 nodes (right)

In our experiments, we varied the network size from 300, 400 nodes (low-density network), to 600 and 800 (high-density network). Each simulation result in our graphs is an average of five simulation results with the same algorithm parameters.

A. Error ratio evaluation

The error ratio is defined as the percentage of nodes unassigned to a zone (isolated nodes). It is evaluated depending on the two algorithm parameters: the zone radius (**R**) and the number of zones (**zN**).

The left side of Figure 4 shows that the error ratio of our algorithm in a low-density network (300 nodes) is quite high. If we increase the number of nodes (400 nodes), the error decreases (see Figure 4, right side). In the previous two cases, this error decreases when we increase the zone radius and the number of zones. A high error ratio, in a low-density network, is due to two main causes:

- the node connectivity - nodes do not experience enough transmission range to reach neighboring nodes in the network or there is an obstacle between them obstructing communication,
- the radius of the zone **R** - it is not large enough. The propagation of invitation packets stops before reaching all nodes in the network.
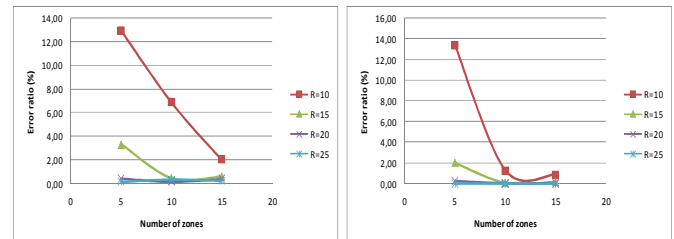

Figure 5. Error ratio for 600 nodes (left) and 800 nodes (right)

The radius of the zone serves to reach more nodes that are within long hop-distances from the *inviting nodes*. The number of zones serves to increase the probability for a node

to be assigned to a zone. There is the same effect when we increase the zone radius.
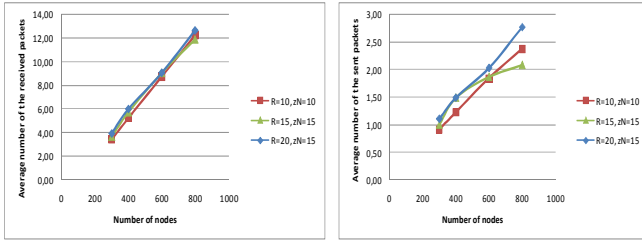

Figure 6. Average number of the received (left) and sent (right) packets

In a dense network (600 nodes), the error ratio goes down to 4% (left side of Figure 5) and close to zero in a network of 800 nodes (right side of Figure 5). That shows the importance of node connectivity which assures total communication coverage (every node is reachable).

B. Overhead evaluation

The overhead, in our experiments, was evaluated by counting the control packets and by estimating the battery level. The control packets are the sent and the received packets exchanged to partition the network into zones.

Figure 6 (left side) shows the average number of the received packets for several simulations (different values of **R** and **zN**). The average does not exceed 13 packets per node in large networks (800 nodes). In Figure 6 (right side), the average number of the sent control packets is shown. It does not exceed 3 packets per node in the network made of 800 nodes.

To estimate the energy consumption of the battery on each node, we have used the energy consumption model of the MICA2 sensor [10]. The size of the exchanged control packet equals 64 bytes. The CPU energy consumption has been ignored because the time of packet processing is generally very small compared to the time of transmission and/or reception.
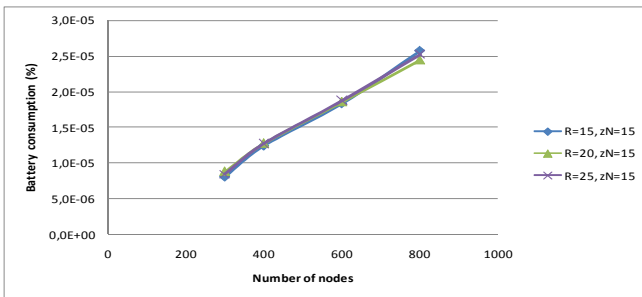

Figure 7. Average percentage of the battery consumption

Figure 7 gives the average percentage of the battery consumption on one network node, based on the energy model of the MICA2 sensors. It shows that our algorithm is energy efficient because of the very low energy consumption during zone construction. The consumed energy does not exceed $3*10^{-5}$ % of the battery's initial energy level.

C. Scalability of the zone partitioning algorithm

The scalability refers to the ability of the algorithm to deliver pertinent results when increasing the number of nodes in the network. As shown by the previous results, our algorithm adapts well when the number of nodes is increased. For example, in a random distribution of 800 nodes with the radius of the zone set to 10 and the number of zones set to 10, few nodes have not been assigned to any zone (isolated nodes). The error ratio is 0.375% of the total number of nodes.

VI. CONCLUSION AND FUTURE WORK

Energy efficiency and data storage are two main features in wireless sensor networks. Network partitioning into many zones tends to save energy and minimize the size of the routing information saved on the sensor. In this paper, we propose a new distributed zone partitioning algorithm for sensor networks based on the number of hops. Network information (e.g. node's energy level, geographical position) is not required to accomplish the partitioning. We have evaluated our approach by estimating the error ratio (percentage of unassigned nodes to any zone), the overhead (number of control packets) and the scalability. In our undergoing work, we propose a routing protocol for wireless sensor networks based on this virtual topology.

REFERENCES

[1] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan. Energy-efficient Communication Protocol for Wireless Microsensor Networks. Proceedings of the IEEE Hawaii International Conference on System Sciences. Vol. 2, p. 10. (2000)

[2] H. Lin, Y. Chu. A clustering technique for large multihop mobile wireless networks. Vehicular Technology Conference Proceedings,Tokyo, Japan. Vol. 2, pp. 1545-1549. (2000)

[3] O. Younis, S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing. Vol.3, Issue 4, pp. 366–379. (2004)

[4] M. Lehsaini, H. Guyennet and M. Feham. An Efficient Cluster-based Self-organization Algorithm for Wireless Sensor Networks. International Journal of Sensor Networks, Inderscience Publishers. Vol. 6, Issue 4 (2009)

[5] B. Chen, K. Jamieson, H. Balakrishnan, and Robert Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. Springer. Wireless Networks. Vol. 8, pp. 481-494(14). (2002)

[6] S. Lindsey, C.S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. IEEE Aerospace Conference Proceedings. Vol. 3, pp. 1125-1130. (2002)

[7] S. Jung, Y. Han, T. Chung. The Concentric Clustering Scheme for Efficient Energy Consumption in the PEGASIS. The 9th International Conference on Advanced Communication Technology. pp. 260-265. (2007)

[8] Y. Yang, W. Hui-Hai, C. Hsiao-Hwa. SHORT: Shortest Hop Routing Tree for Wireless Sensor Networks. IEEE International Conference on Communication. Vol. 2, pp. 3450 - 3454. (2006)

[9] J-Sim Home page. http://www.j-sim.org/.

[10] Crossbow. MICA2 Data sheet, http://www.xbow.com/products/ Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf, access date May 6, 2009.