

A Clustering Topology for Wireless Sensor Networks

New Semantics over Network Topology

Ionel Tudor Calistru, Paul Cotofrei and Kilian Stoffel

Information Management Institute, University of Neuchatel, Neuchatel, Switzerland

Keywords: Sensor Networks, Semantic Sensor Web, Network Topology, Energy-aware Routing, Routing protocols, Data Mining, Clustering, DBSCAN.

Abstract: Sensor networks are a primary source of massive amounts of data about the real world that surrounds us, measuring a wide range of physical parameters in real time. Given the hardware limitations and physical environment in which the sensors must operate, along with frequent changes of network topology, algorithms and protocols must be designed to provide a robust and energy efficient communications mechanism. With a view to addressing these constraints, this paper proposes a routing technique that is based on density based spatial clustering of applications with noise (DBSCAN) algorithm. This technique reveals several network topology semantics, enables the splitting of sensors responsibilities (communication/routing and sensing/monitoring), reduces the level of energy wasted on sending messages through the network by data aggregation only in cluster-head nodes and last but not the least, brings along very good results prolonging the network lifetime.

1 INTRODUCTION

Wireless sensor networking is an emerging technology that has a wide range of potential applications (Sheth et al., 2008; Calbimonte et al., 2011) including environment monitoring (e.g. meteorology, civic planning, traffic management, calamities detection), medical systems (e.g. health monitoring), smart spaces, home automation or homeland security. Such networks will consist of a large number of heterogeneous sensor nodes, sensors that organize themselves into a multihop wireless network. Each node has one or more embedded processors, low-power radios and is normally battery operated. Typically, these nodes are very densely deployed and have sensing, communicating and data processing capabilities. They can gather different type of information such as pressure, humidity, temperature, speed, location and they must coordinate to perform a common task.

Although many protocols and algorithms have been proposed for traditional wireless ad-hoc networks, they are not well suited to the unique features and application requirements of sensor networks. Given the hardware and computational limitations (small size, low power, limited computational and memory capacities), the constraints of the physical environment in which the sensors must operate, along with frequent changes of network topology,

optimal routing algorithms and protocols (Akyildiz et al., 2002; Stojmenovic, 2005) must be designed to provide a robust and energy efficient communications mechanism.

2 MOTIVATION

Prolonged network lifetime, topology awareness, scalability and load balancing are important requirements for many sensor network applications. We have also to take into consideration that the most applications usually deploy a larger number of sensors than the optimum necessary (Akkaya and Younis, 2005). Also many of the sensors that are used to discover alternative networks paths to the Sink (central node) waste their energy without contributing neither to routing or sensing phase (Cardei et al., 2002; Cardei and Wu, 2006). To satisfy these requirements, several solutions have been proposed (Akkaya and Younis, 2005; Abbasi and Younis, 2007) (Younis and Fahmy, 2003; Shin et al., 2006) (Cardei and Wu, 2006) that exploit the trade of among energy, accuracy, and latency. Different topologies were proposed with hierarchical topologies as the most popular one. In this paper we address a cluster based topology with a clear semantics which enables us to better achieve the previously mentioned goals. Accord-

ing to the general classification of clustering scheme done in (Abbasi and Younis, 2007), different clustering techniques have been addressed in function of the network model, clustering objective or the taxonomy of the clustering attributes.

Among the large number of clustering algorithms for sensor networks proposed in the literature, such as LCA (Baker and Ephremides, 1981), CLUBS (Nagpal and Coore, 1998), EEHC (Bandyopadhyay and Coyle, 2003) - all having a linear convergence rate - or LEACH (Heinzelman et al., 2002), HEED (Younis and Fahmy, 2004), EECPL (Bajaber and Awan, 2010), N-LEACH (Tripathi et al., 2012) - with a constant convergence time - our proposed cluster topology is based on the DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm (Sander et al., 1998), a simple and widely used density-based clustering algorithm.

In the context of sensor networks, DBSCAN has already been applied with different purposes; for example, Apiletti (Apiletti et al., 2011) used it to detect sensor correlations whereas Almuzaini (Almuzaini and Gulliver, 2011) applies it to a range-based sensor nodes localization algorithm. However, to the best of our knowledge, DBSCAN has not been used before as a in-network clustering strategy of sensor nodes. Its definition of a cluster is based on the notion of density reachability and it can find arbitrarily shaped clusters, which makes it suitable in the context of randomly deployed wireless sensor networks. Moreover, the algorithm does not require to know the number of clusters a priori and it has a good efficiency on very large datasets.

Taking all this facts into account, we propose a DBSCAN based communication schema, where sensor nodes are organized into clusters. The cluster-head nodes are dedicated solely to transmit (route) messages between them in order to reach the Sink, and the border-nodes are responsible with the sensing (monitoring) activity. Our objective is to prove that this communication strategy fits well sensor networks requirements and improves considerably the network lifetime as well as balances the energy consumption. Moreover, several network topology semantics can be revealed by analyzing the results of a such technique.

The rest of the paper is structured as follows. The next section will present a detailed description of DBSCAN algorithm and its application to our routing technique for wireless sensor networks. Several simulation results will be presented in Section 4, followed by different discussions and improvements. Finally, the last section summarizes our work and proposes some promising future research directions.

3 DBSCAN

One of the major data mining methods is the clustering, defined as the unsupervised learning task of grouping the objects from a dataset into meaningful sub-classes. There has been a lot of research on clustering algorithms for decades but their application to sensor networks rise the following new requirements:

- i. Discovery of clusters with arbitrary shape, as the clusters' shape in sensor networks may be non-convex, spherical, linear, elongated etc.
- ii. Good efficiency on very large networks, with significantly more than just a few thousand objects.

Introduced in (Ester et al., 1996), the clustering algorithm **DBSCAN** relies on a density-based notion of clusters. For each point p of a cluster, the *density* in its ϵ -neighborhood (the number of points situated at a distance less than ϵ from p) has to exceed some threshold $MinP$. **DBSCAN** requires two input parameters (ϵ and $MinP$) and supports the user in determining an appropriate value for them. Designed to discover clusters of arbitrary shape as well as to distinguish noise, **DBSCAN** is also efficient for large spatial datasets.

In (Sander et al., 1998) it is shown that we can use any binary predicate which is symmetric and reflexive in the definition of a neighborhood relation. For example, when clustering polygons, the neighborhood may be defined by the *intersect* predicate. Furthermore, instead of simply counting the objects in the neighborhood of an object, we can use other measures to define the "cardinality" of that neighborhood. Thus, the generalized **GDBSCAN** algorithm (Sander et al., 1998) can cluster point objects as well as spatially extended objects according to both spatial and non-spatial attributes. In the following subsection we present the notion of density-connected sets and in second subsection we give a detailed description of **GDBSCAN** algorithm.

3.1 Density-connected Sets

The concept of *density-connected sets* is a generalization of the concept of *density-based clusters* (Ester et al., 1996; Sander et al., 1998). The generalization concerns the neighborhood relation $N_h(p, p')$ - the distance is replaced by any binary predicate, symmetric and reflexive - and the density measure - the ordinal cardinality is replaced by any function $C_w : 2^D \rightarrow R^+$. In the following, we assume D to be a finite set of objects characterized by spatial and non-spatial attributes.

The definition of a cluster in (Ester et al., 1996) is restricted to the special case of a distance based

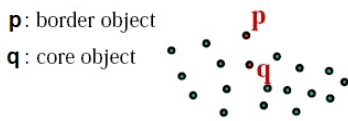


Figure 1: Core Objects and Border Objects (Source: (Sander et al., 1998)).

neighborhood, $N_\epsilon(p) = \{p' \in D : |p - p'| \leq \epsilon\}$. A distance based neighborhood is a natural notion of a neighborhood for point objects, but if clustering spatially extended objects such as a set of polygons of largely differing sizes it may be more appropriate to use neighborhood predicates like *intersects* or *meets* for finding clusters of polygons.

A natural, but naive approach to define a *density-connected set* $S \subseteq D$ as a generalization of a density-based cluster is to require, for each object $p \in S$, that the cardinality measure C_w of the $N_h(p)$ -neighborhood to be less than a given threshold $MinC$. However, this approach fails because a density-connected set contains two types of objects: "inside" the set (core object) and "on the border" of the set (border objects). In general, an N_h -neighborhood of a border object has a significantly lower C_w value than an N_h -neighborhood of a core object (see Fig. 1). Therefore, the value $MinC$ must be set to a relatively low value in order to include all objects belonging to the same density-connected set. This value, however, will not be characteristic for the respective density-connected set - particularly in the presence of noise objects.

The definition of a density-connected set must precise how to decide if a given object p belongs to a given set S by defining the form of the relation "connecting" two objects from the same set. The binary relations *density-reachable* and *density-connected* introduced in (Sander et al., 1998) are used to define a density-connected set with respect to N_h -neighborhood, C_w cardinality function and threshold $MinC$.

Definition 1 (directly density-reachable). *An object p is directly density-reachable from an object q if $p \in N_h(q)$ and $C_w(N_h(q)) \geq MinC$.*

Obviously, directly density-reachable is symmetric for pairs of core objects. In general, however, it is not symmetric if one core object and one border object are involved (see Fig. 2).

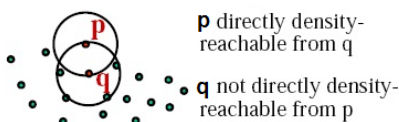


Figure 2: Directly density-reachable relation (Source: (Sander et al., 1998)).

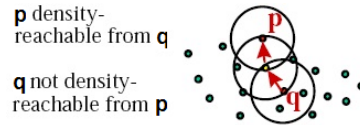


Figure 3: Density-reachability relation (Source: (Sander et al., 1998)).

Definition 2 (density-reachable). *An object p is density-reachable from an object q if there is a chain of objects $p_1 = q, p_2, \dots, p_n = p$, such that $\forall i = 1, \dots, n : p_{i+1}$ is directly density-reachable from p_i .*

Density-reachability is a canonical extension of direct density-reachability. This relation is transitive, but it is not symmetric in general (see Fig 3), except for core objects (a chain from q to p can be reversed if p is also a core object).

Two border objects p and q of the same density-connected set S may be not density reachable from each other. However, two objects belonging to the same density-connected set must be "connected" by a symmetric binary relation. This relation, denoted *density-connectivity*, requires a core object in S from which both objects of S are density-reachable.

Definition 3 (density-connected). *An object p is density-connected to an object q if there is an object o such that both p and q are density-reachable from o .*

Density-connectivity is a symmetric relation. For density reachable objects, the relation of density-connectivity is also reflexive (see Fig 4).

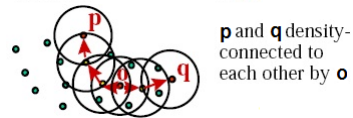


Figure 4: Density-connectivity (Source: (Sander et al., 1998)).

Definition 4 (density-connected set). *The set $S \subseteq D$ is density-connected if satisfies the following conditions:*

- i. *Maximality: $\forall p, q \in D$: if $p \in S$ and q is density-reachable from p then $q \in S$.*
- ii. *Connectivity: $\forall p, q \in S$. p is density-connected to q .*

Note that a density-connected set S contains at least one core object: since S contains at least one object p , this one must be density-connected to itself via some object o (which may be equal to p). Thus, at least o has to satisfy the core condition, $C_w(N_h(o)) \geq MinC$.

3.2 GDBSCAN

The clustering algorithm GDBSCAN (Generalized

Density Based Spatial Clustering of Applications with Noise), introduced by Sander (Sander et al., 1998), was designed to discover density-connected sets in a spatial, possible noise, dataset.

To find a density-connected set, GDBSCAN starts with an arbitrary object p and retrieves all objects density-reachable from p with respect to N_h , C_w and $MinC$. If p is a core object, this procedure yields a density-connected set. If p is not a core object, no objects are density-reachable from p and p is assigned to *NOISE*. This procedure is iteratively applied to each object which has not yet been classified.

```
GDBSCAN (SetOfObjects, Nh, Cw, MinC)
//SetOfObjects is UNCLASSIFIED
ClusterId := nextId(NOISE)
FOR i FROM 1 TO SetOfObjects.size DO
  Object := SetOfObjects.get(i);
  IF Object.ClId = UNCLASSIFIED THEN
    IF ExpandCluster
      (SetOfObjects, Object, ClusterId, Nh, Cw, MinC)
    THEN ClusterId := nextId(ClusterId)
  END IF
END IF
END FOR
END; // GDBSCAN
```

SetOfObjects is either the whole database or a discovered cluster from a previous run. N_h and $MinC$ are the global density parameters and C_w is a pointer to a function $C_w(\text{Objects})$ that returns the cardinality of the set Objects. ClusterIds are values from an ordered and countable datatype (e.g. implemented by Integers) satisfying

$$\text{UNCLASSIFIED} < \text{NOISE} < \text{otherIds}.$$

Each object is marked with a clusterId Object.ClId . The function $\text{nextId}(\text{ClusterId})$ returns the successor of ClusterId in the ordering of the datatype (implemented as $\text{Id} := \text{Id}+1$), while the function $\text{SetOfObjects.get}(i)$ returns the i^{th} element of SetOfObjects. Also the call to function ExpandCluster is constructing a density-connected set for a core object Object.

Obviously the efficiency of the above algorithm depends on the efficiency of the neighborhood query because such a query is performed exactly once for each object in SetOfObjects satisfying the selection condition. If an indexing structure is used allowing the execution of a neighborhood query in $O(\log n)$, an overall runtime complexity of $O(n \log n)$ is obtained. Therefore, the basic algorithm DBSCAN (Ester et al., 1996), seen as a specialization of the algorithm GDBSCAN for the parameters $N_h = N_e(\cdot)$, $C_w = \text{ordinal cardinality}$ and $MinC = MinP$, has an overall runtime complexity of $O(n \log n)$.

3.3 DBSCAN based Routing Technique

The main common task of a routing protocol for WSN is to provide a robust and energy efficient communication mechanism that enables the collaboration between a large number of wireless sensors, randomly distributed in a region called Sensor Field, in order to send the collected information to a central processing node, called Sink. To achieve this, one of the well known strategy is to group sensor nodes into clusters. We are introducing a new routing technique for wireless sensor networks (WSN), technique that, to the best of our knowledge, is the first approach based on the DBSCAN clustering algorithm.

The idea of our approach is to form density-based clusters of sensor nodes so that we can enable the splitting of sensors responsibilities: make the border-nodes responsible of sensing/monitoring the events and the cluster-head nodes (core nodes) aware of aggregating/fusion and communication/routing the messages to the Sink node.

As we have already mentioned, our approach is following very closely DBSCAN's definition of a cluster that is based on the notion of density reachability. The notion of directly density-reachable is achieved by the design of WSN, where each sensor node is aware only of its neighbors with whom it can interact (by sending or collecting information) using only local communication strategies. As soon as a sensor node detects a sufficient number ($MinP$) of neighbors, it becomes a cluster-head node, that is dedicated - in our approach - solely to transmit (route) or to aggregate (fusion) information messages. The border-nodes (sensor nodes that make part of the cluster but are not cluster-heads) are only responsible with sensing (monitoring) tasks. When a new event is detected, they just have to transmit the message (describing the event) to the nearest cluster-head node.

The density-reachable notion is assured by a communication between cluster-head nodes (from cluster-head to cluster-head). To directly communicate to each other, two cluster-head nodes must be neighbors, so part of the same cluster. In order to reach the Sink, every cluster-head node must have established its routing paths. This procedure is executed after the clustering phase, when cluster-head nodes are elected. In a first phase, the Sink is broadcasting a special (recognition) message to its cluster head neighbors. In a second phase, as soon as a cluster-head node receives this kind of message (signed by the Sink), it updates its routing table and then forwards it to all its cluster-head neighbors. After a limited number of steps all the cluster-head nodes, that are density-reachable from Sink, have their routing

tables established and are ready to route the events detected by the border-nodes.

One of the key advantages of DBSCAN, in the context of randomly deployed sensor networks, is that it can find arbitrarily shaped clusters. It can even find clusters completely surrounded by (but not connected to) a different cluster. Due to the *MinP* parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced. Also, DBSCAN has the notion of noise, which is very helpful for identifying the sensors that should not waste their energy trying to monitor different events, and could may be helpful for other tasks.

In the next section we will provide some sets of results obtained after simulating the activity of a sensor network under different scenarios. We will see how our approach reveals several network topology semantics, reduces the level of energy wasted on sending messages through the network by data aggregation only in cluster-head nodes and last, but not the least, brings along very good results prolonging the network lifetime.

4 EXPERIMENTS

Our experiments consist of different simulations that are based on the following general **scenario**: A large number of wireless sensors distributed in a region (Sensor Field) will collaborate for a common application such as environmental monitoring. They should collect and send information about different events detected, to a central processing node (Sink).

To probe our routing algorithm we have used Repast Suite¹ that is a family of advanced, free and open source agent-based modeling and simulation platforms. It can be found in two main editions: Repast Symphony (North et al., 2013) and Repast for High Performance Computing (HPC) (Collier and North, 2012). The output of our simulation was collected in a HDF5² file, that is a format for flexible and efficient I/O when dealing with high volume and complex data.

The battery consumption process have been implemented as a credit point system, where each activity of the sensor node has assigned an amount of points. Each and every sensors have an initial maximum battery capacity. Activities such as sleep mode, send/receive messages and sensing events are defined. During simulation, the battery charge is decreased gradually according to the sensor activities. Also several global counters like number of messages sent

¹<http://repast.sourceforge.net/>

²<http://hdfgroup.org/HDF5/>

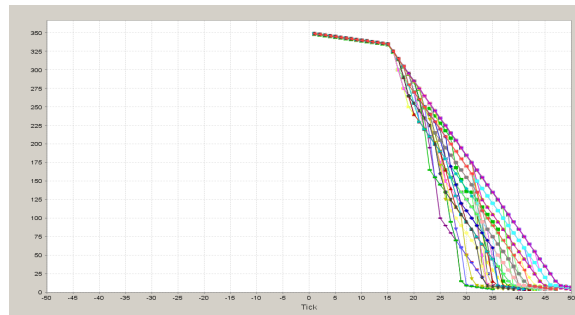


Figure 5: Sensors' energy evolution (Directed Diffusion).

through the network or number of unique messages sent through the network are continuously updated.

To underline the improvements of the technique that we propose, we consider as a benchmark another routing algorithm, an adaptation of *Directed Diffusion* approach (Akkaya and Younis, 2005; Intanagonwiwat et al., 2003; Estrin et al., 1999). The basic idea consists of sending messages of recognition (interest) between the sensor nodes. Based on these messages (known also as gradient fields), each sensor node builds up its own routing list to reach the central node, Sink. The criterion taken into account is *Minimum Energy Path (MEP)*, defining the path that consumes less energy for sending packets between source and sink node. For simplicity, energy required to transmit data is the same between any two nodes that can communicate.

Figure 5 presents the evolution of sensors energy when the routing list is constructed based on directed diffusion approach. As all nodes are tacking part to all network activities, the behavior of all sensors is pretty much the same. After only 50 ticks (simulation ticks) all the sensors finished already their power. We can remark also that first sensor node that dies finishes its energy after around 28 ticks. The total amount of energy spent on routing messages through the network by the alive sensors riches 235 (see Fig. 6).

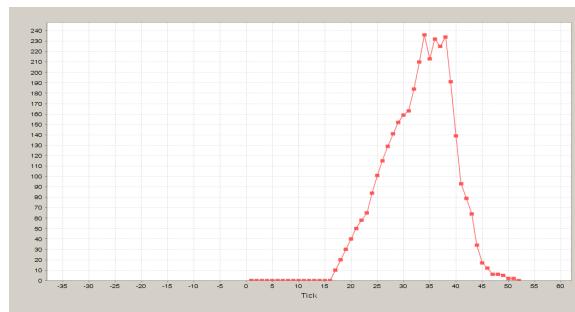


Figure 6: Energy spent by alive sensors to route messages.

4.1 DBSCAN over Sensor Networks

In this first scenario, all nodes were taking part to the network activity, even though many of them were useless. Keeping in mind that our main goal is to maximize network lifetime and to reduce the level of energy consumption, much better results can be achieved by splitting the sensor responsibilities in two tasks: communication/routing and sensing/monitoring. And this is what our DBSCAN approach does.

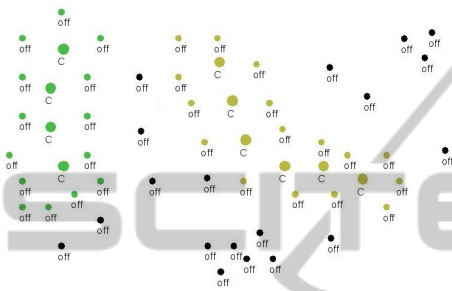


Figure 7: DBSCAN Clusters.

Figure 7 shows a sample of DBSCAN output over a sensor network. Sensor nodes are organized into two main clusters (colored nodes). The black nodes represent the "noise", i.e. the sensors that should not participate to any action and that will be kept in a sleepy state. The so-called "core nodes" or cluster-heads are marked in the figure with "C" and their only responsibility is to route the received data to the Sink. The "border nodes" (clustered nodes that are not cluster-heads) are aware of monitoring different events that may occur. They also transmit a message directly to their cluster-head, informing about the events already detected. In the clustering phase, their state is "off".

The performance criterion used to compare DBSCAN clustering schema against the Directed Diffusion Routing Algorithm is related to the total energy spent by alive sensors for sending messages through the network (Fig. 8). The latest results show us that

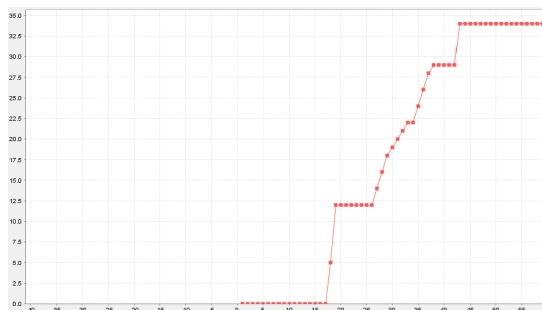


Figure 8: Energy spent by alive sensors to route messages (DBSCAN).

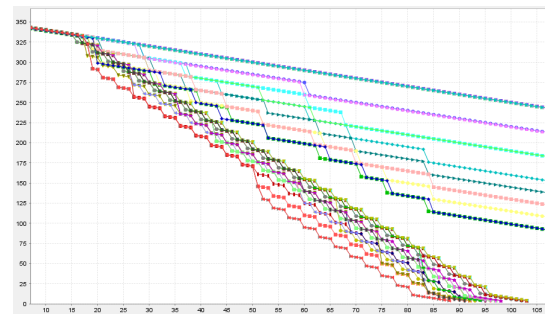


Figure 9: Sensors' energy evolution (DBSCAN).

the network lifetime is considerably prolonged; the first node dies only after **80** simulation ticks (Fig. 9).

5 DISCUSSION

As mentioned in the section before, our first results are related to the Directed Diffusion Routing Algorithm. We remind here that the number of alive sensors starts to decrease around tick **28** (Fig. 5) and that the energy spent with sending messages is about **235** (Fig. 6). In the following we will see how current results can be improved by several approaches.

Data Fusion (Akyildiz et al., 2002) is a technique that can be used to aggregate and to avoid sending duplicates messages. One sensor node will not broadcast a message about the same event more than once. Consequently, the number of messages sent in the network is reduced, saving also energy consumption, especially of those sensor nodes that are frequently used in routing data. By implementing Data Fusion into our first scenario, the simulation results show that we can have now all sensors alive for a longer period (about **37** ticks), the energy consumption being more uniform. Of course, the major advantage is the reducing of the energy used for sending messages between nodes, that is about **100 (57% less)**.

Another important aspect is that, in our previous examples, all sensor nodes were active but not all of them were needed to contribute. As we have motivated, the technique that we propose, based on DBSCAN approach, has the notion of noise. This is very helpful for identifying the sensors that should not waste their energy trying to monitor different events, and that may be helpful for further tasks. In our scenario, about **20 nodes out of 50** may remain in a sleepy state conserving their energy by not participating to routing or monitoring activities.

We must underline that implementing Data Fusion on each sensor node might be too expensive. A remarkable improvement of our DBSCAN based schema is that the messages are routed to Sink "from

Table 1: Repast HPC Simulation Results.

#	Directed Diffusion(DD)			DD with Data Fusion			DBSCAN			DBSCAN Data Fusion		
	Msg	Evts(Unique)	Power	Msg	Evts(Unique)	Power	Msg	Evts(Unique)	Power	Msg	Evts(Unique)	Power
1	0	0(0)	90%	0	0(0)	90%	0	0(0)	92.1 %	0	0(0)	92.1 %
2	121	8(3)	74.1%	113	3(3)	74.5%	77	3(2)	80.5 %	70	2(2)	80.6 %
3	153	11(4)	62.6%	134	4(4)	63.5%	90	4(2)	72.1 %	84	2(2)	72.5 %
4	214	13(5)	49.6%	160	5(5)	52.2%	113	6(4)	63.2 %	102	4(4)	63.8 %
5	363	16(8)	33%	304	9(9)	35.2%	216	12(6)	50.9 %	189	6(6)	52.2 %
6	390	16(8)	23%	333	9(9)	25.8%	231	13(7)	43.1 %	197	7(7)	44.2 %
7	400	16(8)	17%	343	9(9)	28.2%	240	15(8)	36.1 %	203	8(8)	37.7 %

cluster-head to cluster-head", implying a much less number of sensors than for Directed Diffusion approach. By limiting the number of steps in the routing paths, the information arrive much more quickly to the central node. This brings us a great advantage, and we can see that the energy consumed by broadcasting messages between sensors, in this last scenario, is less than **35** (Fig. 8), which represents **35%** of the best result for Directed Diffusion with Data Fusion (around **100**). Figure 9 confirms our DBSCAN results: network lifetime is considerably prolonged by having all sensors alive for about **80** ticks, also better than the best result for Directed Diffusion with Data Fusion (around **37** ticks).

However, the hardware limitations did not permit us to evaluate these techniques with a large scale network. In order to have a deeper analysis of our routing strategy, we have considered to develop a Repast HPC Simulation, by enabling our previous solution to work in a parallel distributed environment. Taking the advantage of a computer cluster network, we were able to work with an increased number of sensors randomly deployed. The summary of our results, including the first seven simulation steps for a network with eight hundred nodes (simulated sensors), is presented in Table 1.

Directed Diffusion and DBSCAN strategies were both simulated with and without Data Fusion. All the four cases, presented in Table 1, were running in parallel. The columns *Msg* indicate the number of messages that have been sent so far between sensor nodes. The table summarizes also the number of events detected by the network and sent to central Sink node, how many of them are unique (have not been detected before) as well as the remaining network power.

As we may remark, a bigger number of events are detected in earlier steps in the case of Directed Diffusion strategy. The explanation for this fact is that the events are generated by a uniform distributed random process. In the earlier steps, when events occur very close to Sink, it is better to send the information directly to the Sink and not to a cluster-head. On the other hand, when events occur far away from Sink, the clustering can speed up the events routing, fewer hops being needed to reach the central node.

The results listed in Table 1 show us also that the communication level between sensor nodes (*Msg*) is considerably reduced when using DBSCAN, and this is reflected also in the surplus of network energy. However, several interesting results may be obtained by adjusting DBSCAN parameters like *MinP* (number of neighbors a sensor node must have in its neighborhood to act as a cluster-head) and ϵ (the sensor radius within sensors may communicate). At this point we can say that a DBSCAN based communication strategy improves considerably the network lifetime as well as balances the energy consumption.

6 CONCLUSIONS AND FUTURE WORK

To provide a robust and energy efficient communications mechanism for wireless sensor network we proposed a routing technique derived from the density based spatial clustering of applications with noise (DBSCAN) algorithm. Based on a series of simulation-based experiments, we could conclude that the implementation of DBSCAN clustering technique to WSN reveals several network topology semantics, by enabling the routing from cluster-head to cluster-head and letting the sensing responsibilities to the border-nodes. Moreover, the DBSCAN strategy reduces the level of energy wasted on sending messages through the network by data aggregation only in cluster-head nodes and brings along very good results prolonging the network lifetime.

Based on the successful results of this preliminary research, several possibilities for future work may be identified:

- The implementation of a technique that is considering, periodically or dynamically (in specific conditions), re-clustering of sensor nodes (since the network topology changes in time, due to loss of several sensors).
- The extension of the routing technique to a dynamic self-reorganization technique that may learn how to adjust the DBSCAN algorithm parameters (*MinP*, ϵ) in order to optimize a specific

streaming data mining task executed on a specific network topology.

ACKNOWLEDGEMENTS

We are thankful to Prof. Dr. Cornelius Croitoru ("Alexandru Ioan Cuza" University, Iasi, Romania) for the contribution and supervision of Ionel Tudor Calistru previous work, bachelor and master thesis, related to wireless sensor networks domain.

REFERENCES

- Abbasi, A. and Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.*, 30(14):2826–2841.
- Akkaya, K. and Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349.
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40:102–114.
- Almuzaini, K. and Gulliver, T. (2011). Range-based localization in wireless networks using the dbSCAN clustering algorithm. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–7.
- Apiletti, D., Baralis, E., and Cerquitelli, T. (2011). Energy-saving models for wireless sensor networks. *Knowl. Inf. Syst.*, 28:615–644.
- Bajaber, F. and Awan, I. (2010). Energy efficient clustering protocol to enhance lifetime of wireless sensor network. *Journal of Ambient Intelligence and Humanized Computing*, 1:239–248.
- Baker, D. and Ephremides, A. (1981). The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Trans. on Communications*, 29(11):1694–1701.
- Bandyopadhyay, S. and Coyle, E. (2003). An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proc. of the 22nd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1713–1723.
- Calbimonte, J.-P., Jeung, H., Corcho, O., and Aberer, K. (2011). Semantic sensor data search in a large-scale federated sensor network. *4th International Workshop on Semantic Sensor Networks 2011*, 11:23–38.
- Cardei, M., MacCallum, D., Cheng, M. X., Min, M., Jia, X., Li, D., and Du, D.-Z. (2002). Wireless sensor networks with energy efficient organization. *Journal of Interconnection Networks*, 3:213–229.
- Cardei, M. and Wu, J. (2006). Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Computer Communications*, 29:413–420.
- Collier, N. and North, M. (2012). Parallel agent-based simulation with repast for high performance computing. *Simulation: Transactions of the Society for Modeling and Simulation International*.
- Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231.
- Estrin, D., Govindan, R., Heidemann, J., and Kumar, S. (1999). Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270.
- Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. on Wireless Communications*, 1(4):660–670.
- Intanagonwivat, C., Govindan, R., Estrin, D., Heidemann, J., and Silva, F. (2003). Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking (TON)*, 11:2–16.
- Nagpal, R. and Coore, D. (1998). An algorithm for group formation in an amorphous computer. In *In Proc. of the 10th Int. Conf. on Parallel and Distributed Computing Systems (PDCS)*.
- North, M., Collier, N., Ozik, J., Tataru, E., Altaweel, M., Macal, C., Bragen, M., and Sydelko, P. (2013). Complex adaptive systems modeling with repast symphony. *Complex Adaptive Systems Modeling*, 1(3):1–26.
- Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: the algorithm gSCAN and its applications. *Data Mining and Knowledge Discovery*, 2:169–194.
- Sheth, A., Henson, C., and Sahoo, S. (2008). Semantic sensor web. *IEEE Internet Computing*, 12:78–83.
- Shin, K., Abraham, A., and Han, S. Y. (2006). Self organizing sensors by minimization of cluster heads using intelligent clustering. *Journal of Digital Information Management*, 4:87–923.
- Stojmenovic, I. (2005). *Handbook of Sensor Networks: Algorithms and Architectures*. John Wiley & Sons.
- Tripathi, R., Singh, Y., and Verma, N. (2012). N-leach, a balanced cost cluster-heads selection algorithm for wireless sensor network. In *In National Conference on Communications (NCC)*.
- Younis, O. and Fahmy, S. (2003). Distributed clustering for scalable, long-lived sensor networks. Technical report, Purdue University.
- Younis, O. and Fahmy, S. (2004). Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans. on Mobile Computing*, 3(4):366–379.