X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

COMMUNITY DETECTION APPROACH FOR CLUSTER FORMATION IN WIRELESS SENSOR NETWORKS

Leonardo N. Ferreira[1], **A.R. Pinto**[1,2], **Alneu A. Lopes**[1], **Liang Zhao**[1]
ICMC - USP - Universidade de São Paulo / São Carlos - SP[1]
DCCE - UNESP - Universidade do Estado de São Paulo, São José do Rio Preto-SP[2]
{leoferr, arpinto, alneu, zhao}@icmc.usp.br, arpinto@ibilce.unesp.br

Abstract – Wireless Sensor Networks (WSN) are composed of tiny wireless nodes that collect and process data. These networks can be composed of hundreds of nodes and be deployed in large and hazardous monitoring areas. WSN presents several hadware, software and communication constraints and a limited energy budget. Thus, one of the most challenging tasks in this technology is the achievement of a trade-off between Quality of Service and energy consumption. Most of the wireless standards are not able to deal with these large scale challenges. Therefore, when these WSN are deployed on large monitoring areas, they must be organized and divided into clusters in order to cover the monitoring area. Thus, it is necessary to develop approaches that organize the network and minimize lost messages rate. This way, community detection approaches could be an interesting alternative to organize and improve WSNs performance. In this paper, a community detection approach for cluster formation in large scale WSN is presented. Our approach was able to decrease lost messages rate when compared with a random cluster-head deployment approach.

Keywords – Complex Networks, Community Detection, Wireless Sensor Networks

1. INTRODUCTION

Wireless Sensor Network (WSN) is usually formed by a large number of sensor nodes with limited computing and radio communication capabilities, deployed to sense and control the field [1]. These nodes typically organize themselves in order to perform collaborative sensing, computation and data delivery tasks. A lot of applications can be developed with this technology [1–4].

The battery consumption is a key issue when the goal is to extend the lifetime of the network. There are many researches that use techniques as sleep-wake schemes and energy saving [4]. Nevertheless, these techniques induce lots of changes in the network topology and the network management effort is increased. Thus, in order to deal with these features, it is necessary new algorithms and protocols that enable the nodes to support topology changes.

WSN are supposed to sense signals in real world. Concepts like "data freshness" are important in its applications [1]. This way, depending on the application, data temporal validity can expire very quickly. Therefore, real-world conditions can introduce explicit or implicit time constraints. Typically, applications consume data in a periodic fashion imposing firm deadlines in the data fusion task [1]. Messages must arrive in master node before the data fusion deadline. However, energy expensive operations in nodes have to be controlled. Communication, for instance, is a very energy-consuming task. This way, a trade-off between reducing energy consumption, meeting time constraints and minimizing lost message rate is necessary. Therefore, when large WSN are considered it is often useful to organize the nodes into clusters. This approach present several advantages like: greater extensions of monitoring area that can be covered, packet collisions reduction in wireless media and energy saving.

WSN can be considered complex networks [2], mainly due to its characteristics (high fault degree, hardware and software constraints, limited battery budget, large number of nodes) and the high number of connections between the nodes that are necessary for its correct functioning. When these WSN are deployed over large monitoring areas, they must be organized and divided in clusters. Thus, community detection approaches could be an interesting alternative to organize and improve WSN's performance.

In this paper, we present a cluster formation approach based on community detection technique [5]. Furthermore, a random deployment cluster deployment is presented. Our community detection technique was able to decrease lost message rate in a large networks, what could be observed through simulations. This paper is organized as follow: a brief description of WSN and complex networks is presented in Section 2 and 3 respectively. Related works are presented in Section 4. The used model is defined in Section 5. Finally, simulation results are presented in section 6 and final remarks at Section 7.

2. WIRELESS SENSOR NETWORKS

Wsn can be composed of a large number of nodes and it presents three main kind of topologies: star (one hop), cluster-tree (network is divided in clusters) and mesh (a routing algorithm between nodes is considered) [4]. These networks are composed of small communicating nodes that contain a sensing unit, wireless communication module, processor, memory and a power supply, typically a battery [1, 4]. The nodes that compose these networks are able to collect data and communicate with each other. The set of nodes can be composed by same sensors or some of them may have special characteristics, like different kinds of sensors. Some WSNs consider the use of a base station that has more computational power than other nodes. The base station is responsible to collect, process and store data sent by slave nodes [4].

Resources in WSN technology, such as processor, memory and battery, are generally constrained. Some networks are deployed in hazardous or inaccessible places where change of battery replacement is prohibitive [2]. Thus, there are many research effort aiming to increase the system lifetime by adopting approaches that minimize the duration of communication tasks and also minimize context switches. Due to battery depletion, faults in wireless communication and hardware nodes, the network topology becomes very dynamic [3].

Some approaches consider a large number of nodes (a dense network), which are deployed near the phenomenon that needs to be monitored. Sometimes, when the network is deployed quickly and their nodes are deployed by random, the position of the nodes can not be predetermined [2]. The strategy behind the deployment of a large number of non-reliable nodes has several advantages: (i) better fault tolerance through distributed operation; (ii) uniform covering of the monitored environment; (iii) easy deployment; (iv) reduced energy consumption; and (v) longer network lifetime.

Data fusion approaches are used to increase sensor reading dependability, in order to make a more accurate estimation of the monitored environment and to achieve longer network lifetime [3]. In these approaches, collected data are sent to a base station that fuses the data to extract useful information from a set of readings. Even in the presence of faulty sensors, dependable information may be generated. This issue is one of the most important one that outcomes from data fusion approaches: it is no longer necessary to rely just upon one sensor reading, when supporting dependable applications.

Large WSNs present several advantages, however self-management characteristics are required in order to deal with a large number of nodes. Self-management techniques are part of autonomic-computing methodologies, that can also be used to manage WSN with conflicting targets (energy efficiency, self organizing, time constraints and fault tolerance). The main goal of self-management is the development of a computing system that does not need the human intervention. Thus, computing systems are able to self-organize and self-optimize themselves, once they follow global goals indicated by a system administrator .

3. COMMUNITY DETECTION IN COMPLEX NETWORK

The increasing interest in studying and understanding real networks is motivated by many science fields. Example of these networks is social network [6], where people are represented by nodes and their friendship represented by connections. The World Wide Web is another example, where each webpage is represented by nodes and its hyperlinks denoted by edges [7]. Other examples are energy transmission [8], neural networks [9] and protein iteration [10].

An interesting property observed in many real networks is the modular structure [11]. In these networks, there are a lot of edges between nodes of the same subgroups and few edges between different nodes subgroups. These groups of vertices, also defined as *communities* [11], are illustrated in Figure 1. The World Wide Web is a real network that has many hyperlinks between related web pages and a few hyperlinks between unrelated web pages. The understanding of community structure present in complex networks is interesting to many research fields because it may reveal important information about the dealing problem.

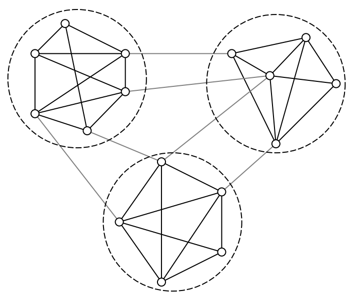


Figure 1: Network with three communities represented by dotted line circles.

In order to understand and extract information of real networks, many algorithms of community detection was developed. These algorithms are classified as agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms start considering that every node is a community and merge them forming larger communities. On the other hand, divisive algorithms detect communities by considering at the beginning that the whole network as a community and divide it in smaller communities.

Many techniques have been developed for community detection. A divisive technique calculates minimum path between all nodes, counts the number that every edge was used (edge betweeness) and removes the most used ones [12]. Another method to find communities uses the concept of a Brownian motion developed in [13] and later extended in [14]. A Brownian particle measures the distance between two nodes and it is used to calculate a dissimilarity index. According to this index, network is decomposed into communities. An agglomerative algorithm uses a measure called modularity and merges edges that cause the highest increase of this measure [15]. Artificial integrate-and-fire neurons was used to represent nodes and detect communities by synchronizing their fire [16]. Another algorithm places the nodes in a circle and move them until the nodes form groups along the circle representing the communities [17].

The agglomerative algorithm proposed by [5] uses the modularity Q that measures the quality of some network division. This algorithm maintain three data structures:

1. A matrix containing ΔQ_{ij} for each pair i e j of communities that has at least one edge between them.
2. A max-heap H with the largest elements of each row of the matrix ΔQ_{ij}
3. An array with elements a_i with the fraction of ends of edges that are attached to vertices in community i.

These structures (ΔQ_{ij} and a array) are initially set according to 1 and 2.

$$\Delta Q_{ij} = \begin{cases} \frac{1}{2m} - \frac{kikj}{(2m)^2} & \text{if } i, j \text{ are connected,} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

and

$$a_i = \frac{k_i}{2m} \tag{2}$$

where m is the number of edges in the graph and k_i is the number of edges incident upon a node i. After setting the data structures, the Algortihm 1 describs the process of community detection.

Algorithm 1: Community Detection

begin
  Calculate the initial values of ΔQ_{ij} and a_i according to 1 and 2 respectively;
  Put the largest element of each row of the matrix ΔQ_{ij} in max-heap;
  repeat
    Select the largest ΔQ_{ij} from H;
    Join communities i and j;
    Update matrix ΔQ, max-heap H and a_i;
    $Q \leftarrow Q + \Delta Q_{ij}$;
  until *until only one community remains*;
end

When communities i and j are merged, we label this new community as j, update every k element of jth row and column, and remove the ith row and column. To update the matrix ΔQ_{ij} is considered three cases. If community k is connected to both i and j, then

$$\Delta Q'_{jk} = \Delta Q_{ik} + \Delta Q_{jk} \tag{3}$$

If k is connected to i but not j, then

$$\Delta Q'_{jk} = \Delta Q_{ik} - 2a_j a_k \tag{4}$$

If k is connected to j but not i, then

$$\Delta Q'_{jk} = \Delta Q_{jk} - 2a_i a_k \tag{5}$$

Finally update array $a'_j = a_j + a_i$ and $a_i = 0$.

4. RELATED WORKS

In this section we discuss the state of the art of complex networks approaches used in WSN self-organizing techniques. In [18] a small-world model is used to build an efficient data dissemination in a WSN. Authors had proposed and evaluated two small-world models that were used to build a data dissemination approach in WSN, their main goal was the trade-off between energy and latency. Watts and Strogatz [19] proposed the small-world model where most of the vertices of a network is reachable from other ones through a small number of edges. This characteristic is found, for example, in social networks, where everyone can be reached through a short chain of social acquaintances [20]. Small-world principles are also presented in [21], in this work the authors took advantage of data correlations in order to avoid transmitting redundancy. Moreover, a simple randomized algorithm for routing information on a sensors grid that satisfies the collision time condition is presented. The investigation of hybrid WSN (mix of wired and wireless connection) is presented in [22]. Small-world approach was used to prove that some wired connections can save energy and decrease average hop in a WSN.

A topology control algorithm for fault tolerance in large scale WSN, named AWSF, was presented in [23]. AWSF is based on scale-free characteristics of some complex networks. Scale-free networks was introduced by Barabási and Albert [24] and, in this model, a few nodes are highly connected while a lot of them have few connections [20]. The AWSF aims to minimize transmission delay and increase robustness. The transmission power is locally adjusted and neighborhood are determined. .

5. MODEL DESCRIPTION

The used communication model considers one *master node* (base station), N_{ch} *Cluster-head* nodes and N_s *slave* nodes. Data collected by slaves is sent to their respective cluster head nodes that perform the data fusion. All the slave nodes reach the cluster head using just one hop. After a sensing slot time window, cluster-head nodes send their messages to the master node. Slave and cluster-head nodes use the same frequency for wireless communication. We consider that cluster heads communicate with master node in a different radio frequency. This difference in radio frequence avoid interference between slave-CH and CH-master communications.

First, the network is built by deploying slaves in a random position inside the monitoring area. The slaves covered by the cluster heads' antenna range are connected, forming clusters of slaves. Slaves inside of more than one CHs is connected to all of them. The Figure 2 shows an example of network. For each simulation iteration (time), slave nodes generate message with λ probability, store the message on its buffer and tries to send it to its cluster heads. The wireless transmission medium is shared with all nodes inside a specific cluster. Thus, every cluster head can receive just one slave message by time and the other ones store their messages on buffer. This behavior was reproduced by choosing a random slave of a cluster that has not a empty buffer and decreasing its buffer. The same behavior was reproduced with cluster heads and masters. If a slave generates a message and is buffer is full, then this message is considered lost. The massage also is lost when a CH's buffer is full and a slave send information to it.

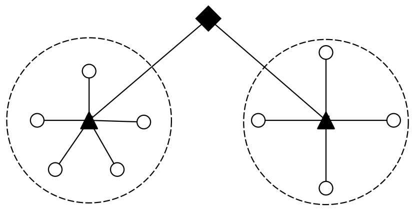


Figure 2: An example of WSN where slaves are represented by circles, cluster heads by triangles and the master by a diamond. The circles with dotted line show the clusters.

The following simulation parameters were considered in our model:

Number of slaves: A specific number of slave nodes are deployed in the monitoring area.

Number of cluster heads: A number of CH are deployed in the monitoring area and connected to slave accordind to its antenna range.

Deployment Area: Simulation area size to deploy nodes.

Slave buffer size: Slave's message storage capacity.

Cluster head buffer size: CH's message storage capacity.

Simulation time: Number of iteration used as simulation stop criteria.

Probability λ of message generation: A probability of message generation.

Cluster head antenna range: Cluster head coverage. Slaves inside this coverage are connected to this CH.

Slave sensor range: Slave sensoring coverage. If another slave is inside this converage, an artificial edge is created in order to detect community.

5.1. CLUSTER HEADS RANDOM DEPLOYMENT

We have considered that N_s slave nodes were deployed by random in a defined monitoring area. After the slave nodes deployment, N_{ch} cluster head nodes are also randomly deployed. Slaves inside CH's antenna range are linked forming clusters. A common effect observed is that several slave nodes can be positioned in an area where there is no cluster head node. Some cluster head nodes can also be deployed close from each other, covering the same area. After the deployment, the monitoring phase starts. Slave nodes begin to generate messages, cluster head nodes receive them and send them to master node. The monitoring phase is discribed in Algorithm 2.

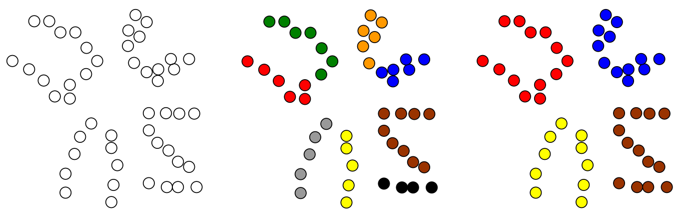
Algorithm 2: Monitoring phase

```
begin
    while simulation time did not finish do
        forall the slaves do
            if generated a message with probability λ then
                if slave node's buffer is not full then
                    store it in slave node's buffer;
                else
                    Lost messages ← lost messages + 1;
                end if
            end if
            if slave buffer is not empty then
                tries to send the message to its cluster-head;
                if message is successfully sent to cluster-head node then
                    message is removed from the buffer;
                end if
            end if
        end forall
        After a sensing time slot there is a slot time used for the cluster head nodes to send their messages to master node;
        Simulation time ← simulation time + 1;
    end while
end
```

5.2. CLUSTER HEADS STRATEGIC DEPLOYMENT

This strategic approach considers three steps: network generation, community detection and CH deployment. The network generation consists on deploying by random the slave nodes in a certain monitoring area. Then, artificial edges are created between slaves inside other sensor range. It means that sensors that are sensing the same area will have an artificial connection. The Second step consists of detecting communities in this slave networks using algorithm proposed by [5] described in Section 3. In the third step, a cluster head node is deployed in the midpoint of each community set of points. The three steps of the strategic deployment are illustrated in Figure 4. Finally the monitoring phase described in Algorithm 2 starts.

We considered the creation of a network becouse graph-based algorithms may find clusters of different forms. Non-graph-based algorithms use some similarity function and may consider that nodes that are far from each other do not belong to the same cluster. This situation is ilustrated in Figure 3.



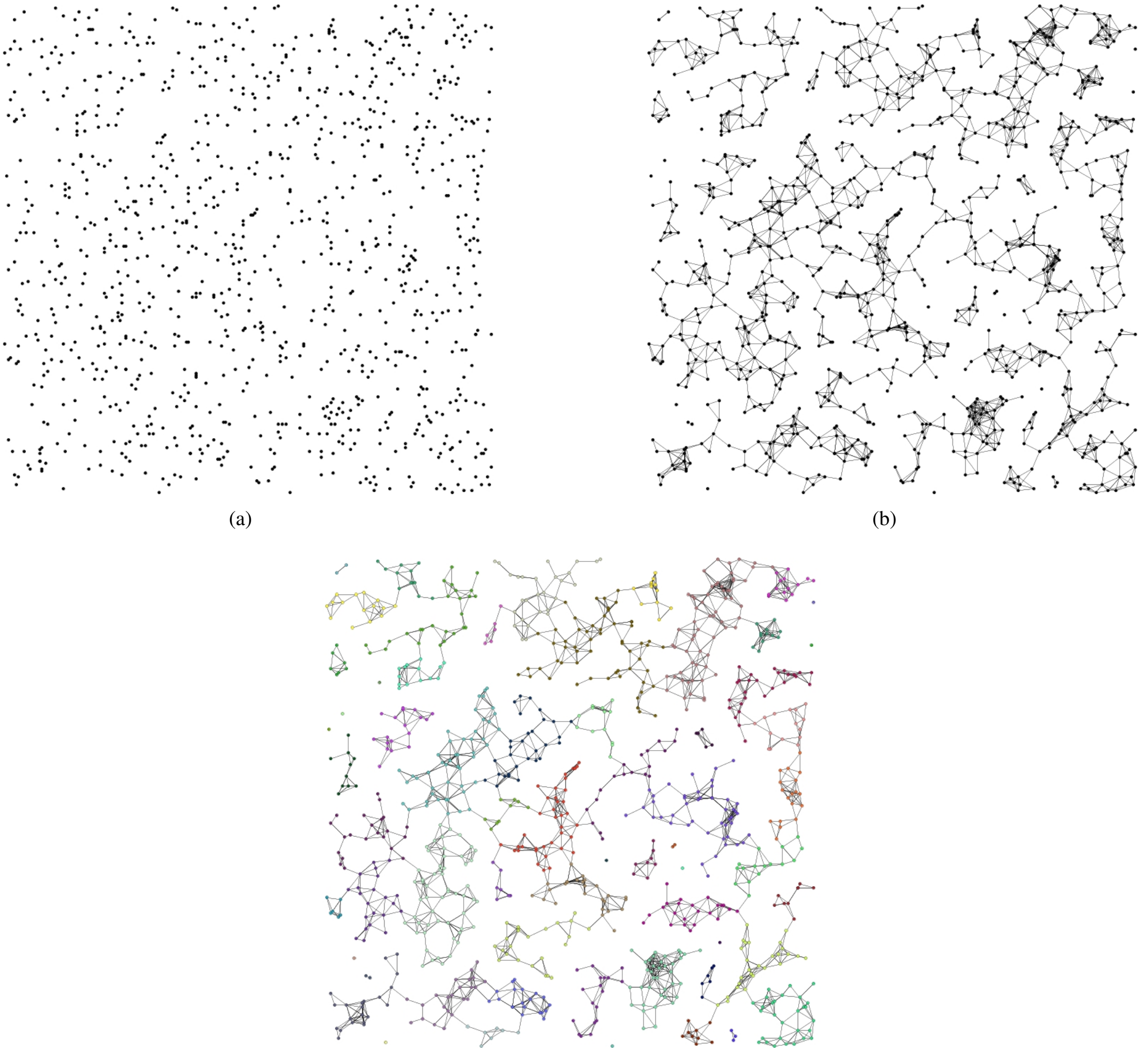
(a) (b) (c)

Figure 3: (a) Slaves random deployment. (b) Example of clustering process using some traditional non-graph-based algorithm. (c) Example of clustering using some graph-based algorithm.

6. SIMULATION RESULTS

The first simulation considered the random approach for some different rate of CH and deployment area. Four rates of CH were considered: 10%, 20%, 30% and 40% and 1000 slaves. This percentage refers to the number of slaves. The results of this simulation were described in Figure 5. It is possible to notice that when the WSN is deployed in larger areas the lost message rate is increased. However, this simulation shows an interesting behavior (20%, 30% and 40% CH): when a $500 \times 500m$ area is considered, lost message rate is 10% bigger than a $1000 \times 1000m$ area this happens due to the large number of collisions in wireless medium. Areas bigger than $1500m$ present larger lost messages rate.

The second simulation is a comparison between random deployment and community detection approach, presented in Figure 6. It is possible to notice (Fig. 6(a)) that in some cases, community detection approach did not showed a representative gain. This behavior is due to the fact that the monitoring area considered is just $300 \times 300m$. Considering the CH antenna range ($100m$), it is



(a) (b) (c)

Figure 4: (a) Random deployment of slaves. (b) Artificial connection between slaves inside other sensor ranges. (c) Community Detection. After this process, the cluster head nodes are deployed in the midpoint of each community set of points.

not necessary to divide the WSN into clusters (a star topology would be sufficient), because the random deployment of a few CH will cover the whole area and, therefore will present the same result of the strategic approach. The second comparison between random deployment and community detection approach is showed in Figure 6(b). When the area was increased to $700 \times 700m$ the community detection approach presented a gain of almost 15% (CH level of 10%). This behavior proves that the community detection approach is just useful in a more sparse WSN, due to the fact that in small areas it is not necessary to divide WNS into clusters. The third comparison is showed in Figure 6(c), this simulation considered an area of $1000 \times 1000m$. The same way, the WSN became very sparse. It is possible to notice that considering a 10% CH level (100 CHs) the community detection gain is almost 53%.

7. FINAL REMARKS

In this paper we have presented two approaches to form clusters in Wireless Sensor Networks. First, we considered the random deployment approach of some cluster heads in the monitoring area. The random deployment approach presented an interesting behavior when more CH nodes were deployed, causing the decrease of lost message rate reaching the global minimum and increasing again. This way, we have detected a relation between the number of slave nodes and cluster head level. On the other hand, the community detection approach based on Newman's technique is more suitable for larger areas. This is due to the fact that in smaller areas it is not necessary to detect and form clusters. The simulations that considered areas of $700 \times 700m$ and $1000 \times 1000m$ showed that the community detection approach presented a reduction in lost message rate of almost 15% and 53% respectively. As future works we recommend an extensive set of simulations considering more parameters in order to understand the influence of these parameters on lost message rate.

Acknowledgments: This paper was supported by Conselho Nacional de Desenvolvimento Cientifico e Tecnológico (CNPq) of Brazil.

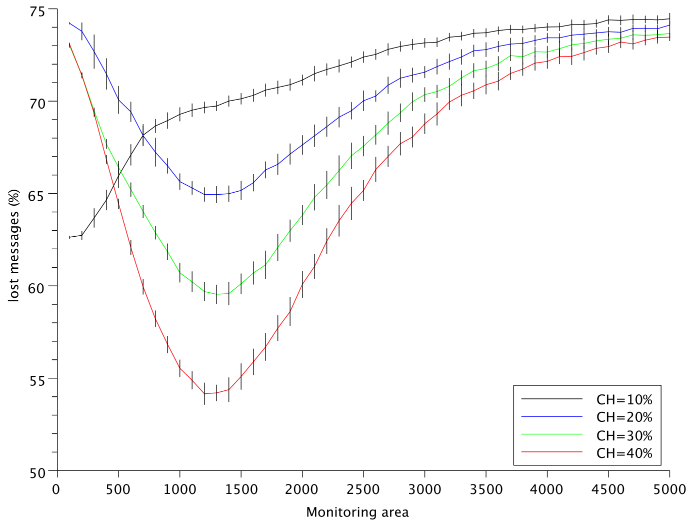
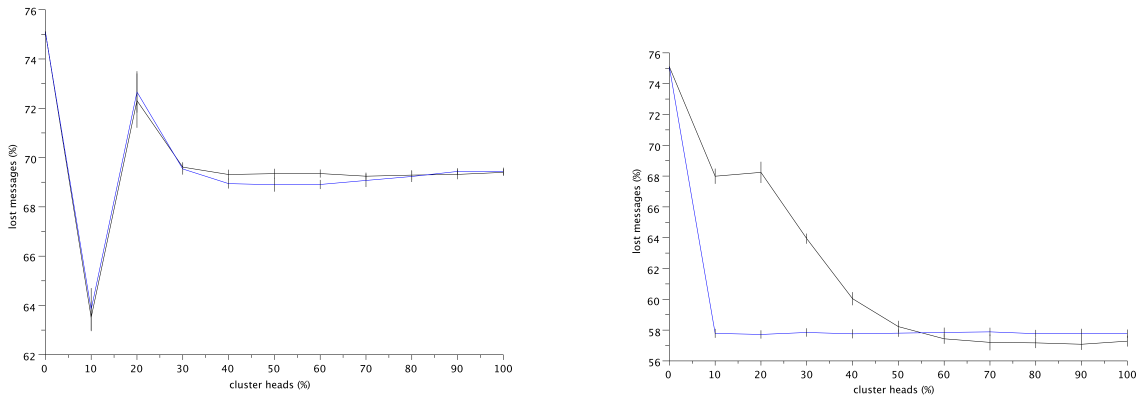
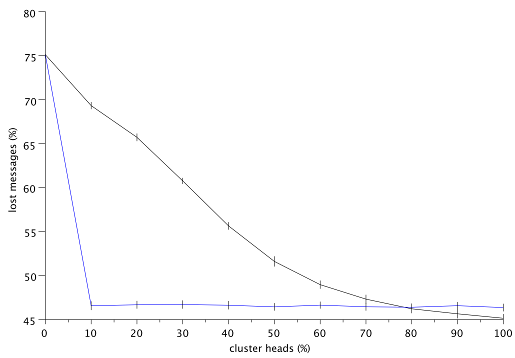


Figure 5: Random approach, 1000 slave nodes, area variable from $500 \times 500m$ to $5000 \times 5000m$, 4 levels of CH were considered 100, 200, 300 and 400, $\lambda = 20\%$, slave node buffer capacity 100 messages, cluster head node capacity 500 messages. Each point represents an average of 30 simulations and its respective standard deviation.



(a) (b)



(c)

Figure 6: Lost messages comparison between random (black line) and strategic (blue line) deployment of CH. The simulation parameters was 1000 slave nodes, slave buffer size and CH buffer size of 100 and 500 respectively, CH antenna range of $100m$, slave sensing range of $10m$ and $\lambda = 20\%$. Figure (a) deployed nodes in simulation area of $300 \times 300m$. (b) simulation area of $700 \times 700m$ and (c) $1000 \times 1000m$. Each point represents an average of 30 simulations and its respective standard deviation.

REFERENCES

[1] I. Akyildiz. "Wireless sensor networks: a survey". *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[2] A. Koubaa, R. Severino, M. Alves and E. Tovar. "Improving Quality-of-Service in Wireless Sensor Networks by Mitigating "Hidden-Node Collisions"". *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 3, pp. 299 –313, aug. 2009.

[3] R. Niu, P. K. Varshney and Q. Cheng. "Distributed detection in a large wireless sensor network". *Inf. Fusion*, vol. 7, pp. 380–394, December 2006.

[4] J. Stankovic, T. Abdelzaher, C. Lu, L. Sha and J. Hou. "Real-time communication and coordination in embedded sensor networks". *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1002 – 1022, july 2003.

[5] A. Clauset, M. E. J. Newman and C. Moore. "Finding community structure in very large networks". August 2004.

[6] J. P. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, January 2000.

[7] B. A. Huberman. *The Laws of the Web: Patterns in the Ecology of Information*. The MIT Press, October 2001.

[8] L. A. N. Amaral, A. Scala, M. Barthélémy and H. E. Stanley. "Classes of small-world networks". *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 21, pp. 11149–11152, October 2000.

[9] O. Sporns. "Networks analysis, complexity, and brain function". *Complex.*, vol. 8, pp. 56–60, September 2002.

[10] A.-L. Barabási and Z. N. Oltvai. "Network biology: understanding the cell's functional organization". *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, February 2004.

[11] M. Girvan and M. E. J. Newman. "Community structure in social and biological networks". *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, June 2002.

[12] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks". Aug 2003.

[13] H. Zhou. "Network landscape from a Brownian particle's perspective". *Phys. Rev. E*, vol. 67, no. 4, pp. 041908, Apr 2003.

[14] H. Zhou. "Distance, dissimilarity index, and network community structure". *Physical Review E*, vol. 67, no. 6, pp. 061901+, June 2003.

[15] M. E. J. Newman. "Fast algorithm for detecting community structure in networks". *Physical Review E*, vol. 69, no. 6, pp. 066133+, Jun 2004.

[16] M. G. Quiles, L. Zhao, F. A. Breve and R. A. F. Romero. "Detecção de comunidades em redes complexas: um modelo de correlação oscilatória". *VII Encontro Nacional de Inteligência Artificial (ENIA)*, pp. 889–898, 2009.

[17] T. de Oliveira, L. Zhao, K. Faceli and A. de Carvalho. "Data clustering based on complex network community detection". In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pp. 2121 –2126, june 2008.

[18] D. Guidoni, R. Mini and A. Loureiro. "Creating small-world models in wireless sensor networks". In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pp. 1 –6, sept. 2008.

[19] D. J. Watts and S. H. Strogatz. "Collective dynamics of 'small-world' networks". *Nature*, vol. 393, no. 6684, pp. 440–442, June 1998.

[20] Luciano, F. A. Rodrigues, G. Travieso and V. P. R. Boas. "Characterization of complex networks: A survey of measurements". *Advances in Physics*, vol. 56, no. 1, pp. 167–242, August 2006.

[21] D. C. Marinescu, C. Yu and G. M. Marinescu. "Self-organization of Very Large Sensor Networks Based on Small-worlds Principles". In *Proceedings of the 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, SASO '09, pp. 115–125, Washington, DC, USA, 2009. IEEE Computer Society.

[22] G. Sharma and R. Mazumdar. "Hybrid sensor networks: a small world". In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 366–377, New York, NY, USA, 2005. ACM.

[23] L. Wang, J. Dang, Y. Jin and H. Jin. "Scale-free topology for large-scale wireless sensor networks". In *Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia on*, pp. 1 –5, sept. 2007.

[24] A. L. Barabasi and R. Albert. "Emergence of Scaling in Random Networks". *Science*, vol. 286, no. 5439, pp. 509–512, October 1999.