International Journal of Future Generation Communication and Networking
Vol. 2, No. 2, June, 2009

Monitoring Connectivity in Wireless Sensor Networks

Benahmed Khelifa[1], H. Haffaf [2], Merabti Madjid[3], and David Llewellyn-Jones[3]

[1] Béchar University, Algeria, benahmed_khelifa@yahoo.fr
[2]University of Es-Sénia Oran, Algeria, haffaf_hafid@yahoo.fr
[3]School of Computing & Mathematical Sciences,
Liverpool John Moores University, U.K,
{M.Merabti, D.Llewellyn-Jones}@ljmu.ac.uk

Abstract

It is important to have continuous connectivity in a wireless sensor network after it is deployed in a hostile environment. However, such networks are constrained by the low user-to-node ratio, limited energy and bandwidth resources, entities that are usually mobile, networks without fixed infrastructure and frequent failure due to problems of energy, vulnerability to attack, etc. To address these difficulties, there is a need for wireless sensor networks to be self-organizing and self-configuring so as to improve performance, increase energy efficiency, save resources and reduce data transmission. In this paper, we present a method for monitoring, maintaining and repairing the communication network of a dynamic mobile wireless sensor network, so that network connectivity is continuously available and provides fault tolerance. Specifically, we propose an algorithm for the detection and surveillance of articulation points in graph connectivity, including an algorithm for -network auto-organization in the event that this occurs.

Keywords: *Wireless Sensor Networks, Monitoring, Connectivity, Fault Tolerance.*

1. Introduction

A wireless sensor network consists of many nodes generally communicating by radio waves. The sensors are not integrated into any existing network architecture, so they communicate through a network of ad hoc wireless connections. The power of each sensor is provided by a battery, for which individual consumption for communication and calculation must be optimized. There are many fields of use for such networks including monitoring biological, chemical, environmental and seismic applications, etc.

The use of these sensor networks in hostile environments means that providing quality of service is essential and requires the implementation of fault-tolerant mechanisms that can ensure availability and continuity of service. For example, the maximum coverage of the regions monitored by the network and connectivity of the various nodes of the network must be maintained. However in an environment where each node can fail unexpectedly resulting in the isolation of some parts of the network, this guarantee is neither automatic nor easy to achieve.

The integration of mechanisms for surveillance, topology control and fault tolerance are crucial for the effective use of wireless sensor networks. There are many current management approaches, but each provides only partial solutions to the problems of monitoring and fault tolerance, and they do not adapt to the properties and constraints of many wireless sensor networks. Therefore, the work presented in this paper gives a new approach for monitoring connectivity in wireless sensors networks. We provide a rigorous analysis for the development of fault-tolerance to ensure both ongoing monitoring of network connectivity and self organization, mainly to enhance the degree of connectivity at critical nodes presenting articulation points in the network.

The rest of this paper is organized as follows. The following sections 2 and 3 introduce the concepts of connectivity, monitoring and fault tolerance. Section 4 gives a brief summary of related research and comparison with our approach where applicable. We model our problem in Section 5. Sections 6 and 7 describe our solution. In Section 8, we present our simulation results. Section 9 concludes the paper.

2. Connectivity

A network of sensors is considered to be connected only if there is at least one path between each pair of nodes in the network. Connectivity depends primarily on the existence of paths. It is affected by changes in topology due to mobility, the failure of nodes, attacks and so on. The consequences of such occurrences include the loss of links, the isolation of nodes, the partitioning of the network, the upgrading of paths and re-routing.

Connectivity can be modeled as a graph $G(V, E)$ where V is the set of vertices (nodes) and E the set of edges (links). This graph is said to be k-connected if there are at least k disjoint paths between every pair of nodes $u, v \in V$. Connectivity is a measure of fault tolerance or diversity of paths in the network. The need for 1-connectivity of the network graph is a fundamental condition of it being operational. Indeed, the connectivity of a network can be expressed as follows [1].

$$\mu(R) = \frac{N.\pi.R^2}{A}, \qquad (1)$$

where R is the radius of transmission, A the area and N the number of nodes in the area A.

Kleinrock and Silverster have shown that when connectivity $\mu(R)$ reaches 6 nodes, the probability that a node is connected tends to 1, *i.e.* that the network forms a connected graph [2].

3. Fault Tolerance

Wireless sensor networks are commonly deployed in hostile environments and are susceptible to numerous faults in several layers of the system. Figure 1 depicts the source of these failures and demonstrates the potential for propagation to higher layers. The source of failures in this classification is divided in to four layers: node, network, sink and the base station.

To address these problems it is useful to implement a system that allows monitoring of the network. At any moment such a system must be able to provide the operational status of different devices and to establish mechanisms that provide fault tolerance. By definition fault tolerance [2] is a technique that has been proven to make systems capable of providing a good service, even in the presence of accidental phenomena such as disturbance of the environment (external faults), failure of hardware components (internal physical faults), or design faults, particularly software faults (bugs). Under the terms of dependability, faults are the causes of errors, mistakes are part of the abnormal state of the system and when errors are propagated to the system interface – *i.e.* when the service provided by the system is incorrect – this results in a failure. When mistakes are accidental and sufficiently rare, it is possible to tolerate them. This requires detecting errors before they occur, with error handling in case

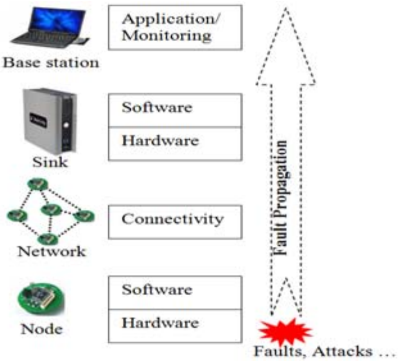


Figure 1. Fault classification and propagation

they can't be rectified. We must also make a diagnosis, in other words identify the fault, isolate faulty components, replace or repair and reset the system in case there is no alternative. In a wireless sensors network, fault tolerance is the ability to ensure the functionality of the network in the face of any interruption due to failures of sensor nodes.

4. Related Work

Connectivity is particularly important for wireless sensor networks. In a wireless sensor network, the deployment strategy often involves using more nodes then necessary and turning off the ones that are not being used for communication or sensing. When the network becomes disconnected, one or more of the redundant nodes can be turned on to repair connectivity [3]. The main problem with this technique is the requirement for extra nodes, and when several nodes in a limited region fail it may no longer be possible to repair the network. Li and Hou study the problem of adding as few nodes as possible to a disconnected static network so that the network remains connected [4]. They show that the problem is NP-Complete and propose some heuristic solutions. These algorithms require global knowledge of the graph and they are time-consuming to apply. Consequently they are typically not applicable in real-time with dynamic networks.

Using mobility to maintain connectivity has attracted many researchers. The general approach has been the use of mobility to carry data between disconnected components of the network [5]. Another approach is the storage of data when connectivity is disrupted, and sending the data when connectivity is subsequently repaired [6, 7]. A significant problem with these approaches is the latency in data transfer for time critical applications.

There are also approaches that can be used to maintain uninterrupted connectivity with dynamic networks. Spanos and Murray propose a technique for providing radio connectivity while moving a group of robots from one configuration to another [8]. However, this analysis is not valid when there are obstacles.

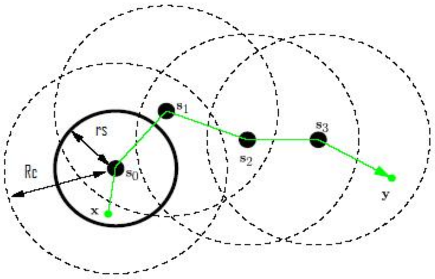
Several other solutions for fault tolerance are based on the nature of redundant sensor networks. Fusion techniques [9, 10] may merge or aggregate the different readings of the sensors. Multi routing paths [8, 11] and techniques to ensure *k*-connectivity between nodes [12, 13, 14, 15, 16] can be applied to increase the reliability of the transmission of messages in wireless sensors networks.

5. Modeling the Problem

In most cases a wireless sensors network can be modelled as a unit graph $G\ (V, E)$ where V is the set of nodes (with each sensor in the network a vertex in the graph) and E the set of all arcs giving opportunities for direct communication between nodes (we assume that the communication is symmetric, meaning that if a node can hear another, it can also be understood by him). The corresponding graph is undirected. If we set $\mathrm{d}(u, v)$ as the physical distance between nodes u and v, and R_c the radius of communication, then E is defined as follows [17, 11].

$$E = \{(u, v) \in V^2 \mid \mathrm{d}(u, v) \leq R_c\} \tag{2}$$

For sensor coverage – i.e. the collection of information by sensors – we need the coverage radius r_s, with $R_c \geq 2r_s$. Figure 2 shows these two ranges (connectivity and coverage).



x, y are Events
R_c is the radio range
S_i are sensor nodes
r_s is the sensing range

Figure 2. Connectivity and coverage in wireless sensor network

6. Connectivity Strategy

In this section we will consider methods used for predicting the partitioning of the network. The prediction algorithm acts as a tool to help provide fault tolerance, aimed at improving the life of the service by detecting critical nodes that might induce a breach of network connectivity should they fail. The mobility of nodes, energy loss, vulnerability to attack and the limited range of their communication implies that the existence of such nodes may result in it becoming impossible to find a route between a source and destination nodes.

The algorithm that we propose for the prediction of partitioning of the network includes the following steps.

- Assess the robustness of the link between nodes.
- If this robustness is below a given threshold, send an alert to self organize the network.

For the assessment of the robustness of communication links, we propose an evaluation based on sets of node-disjoint paths and properties of k-connected graphs.

Theorem (Menger, 1927): In an undirected graph the maximum number of node-disjoint paths from a nonadjacent summit x and summit y is equal to the minimum number of nodes to remove to disconnect x of y [18].

The search for node-disjoint paths between pairs of nodes can be reduced to the search for nodes whose removal disconnects them. Such nodes are called critical points or articulation points and can be detected using a centralized in-depth search algorithm [19]. Figure 3 illustrates this idea.

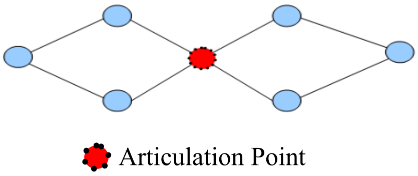


Figure 3. Topology with an articulation point

Our case is limited to 2-connectivity: we require at least two paths between the source and destination to ensure fault-tolerant connectivity.

Definition 1: A graph is *biconnected* if for each pair of summits *u*, *v* with $u \neq v$, there are two summit-disjoint paths that join *u* and *v* [20].

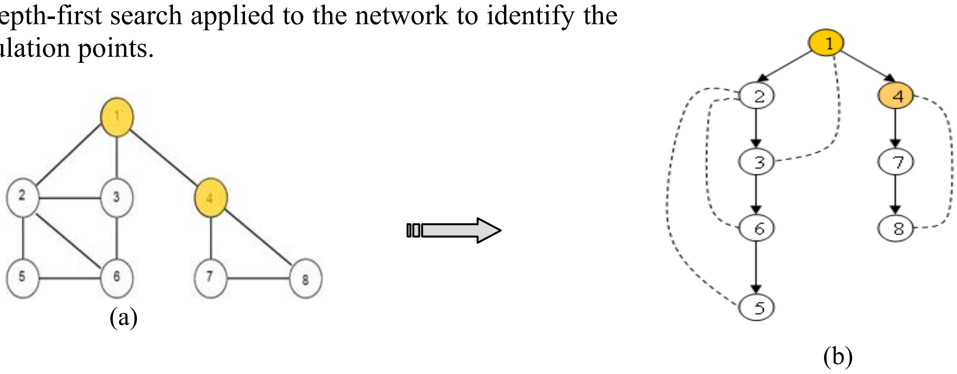
Property: A graph is *biconnected* if and only if it has no articulation point [20].

Algorithm 1: Detection of articulation points in an undirected graph.

Input: *G* (*V*, *E*) Unit Disk Graph
Output: Set of articulation points

- Depth search in graph *G* and generation of spanning tree *T*, (in which back edges are shown as dotted lines) to facilitate computing articulation points.
- A vertex x is not an articulation point if it has no successor, or if each of its successor admits a descendant who has a back edge to an ancestor of x in the tree,
- Particular case: the root is an articulation point if it has more than one successor in the tree.

This algorithm has a binomial complexity of the order of $O(N + M)$ for a graph with N vertices and M edges. This algorithm is demonstrated in the example shown in Figure 4, with the depth-first search applied to the network to identify the articulation points.



(a)

(b)

Figure 4. (a) Unit disk graph and (b) the tree T generated by the depth-first Search

7. Self Organizing Network

Recent scientific study has considered the behavior of birds, insects and viruses and their capacity to organize themselves. Noting also the pervasive presence and potential benefits of self-organization in natural systems, many researchers have now begun to look at how such models of self-organization can be applied to the design of distributed systems.

The mechanisms of self-organization have the potential to provide many solutions in wireless sensor networks. For example, self-organization can be used to change the density of sensor nodes and traffic patterns, or help to reconfigure the network topology in cases where nodes fail or relocate.

Inspired by the behavior of ants that organize themselves (moving to form a bridge) and the capabilities of sensors to move or raise their range of connectivity, we propose the following algorithm to allow the self organization of the network, especially around the articulation points discussed above (AP).

Algorithm 2: Self-organization: the principle

```
Input:   G (V, E), with the set of articulation points (AP) previously detected
Output:  G (V, E), with a minimum set of articulation points so that G will be at least biconnected.

1. For any articulation point (AP) do
   • If there is a neighbour redundant of AP then turn on and go to the AP following (step 1).
   • Else discover the neighbours of AP at one hop,
         • If neighbours have redundant nodes, select at least one node with the greatest energy
           capacity, and move it to the coordinates (x, y) of the AP or increase its communication range;
           go to step 1.
         • Else "no solution at one hop of AP"; go to step 1.
2. End For
```

This algorithm is demonstrated in the example shown in Figure 5, this algorithm applied to the network to auto-organize and increase connectivity around articulation points.

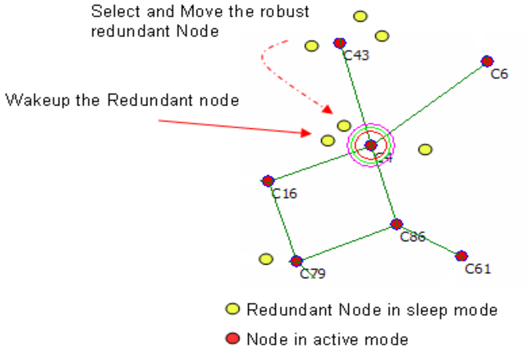


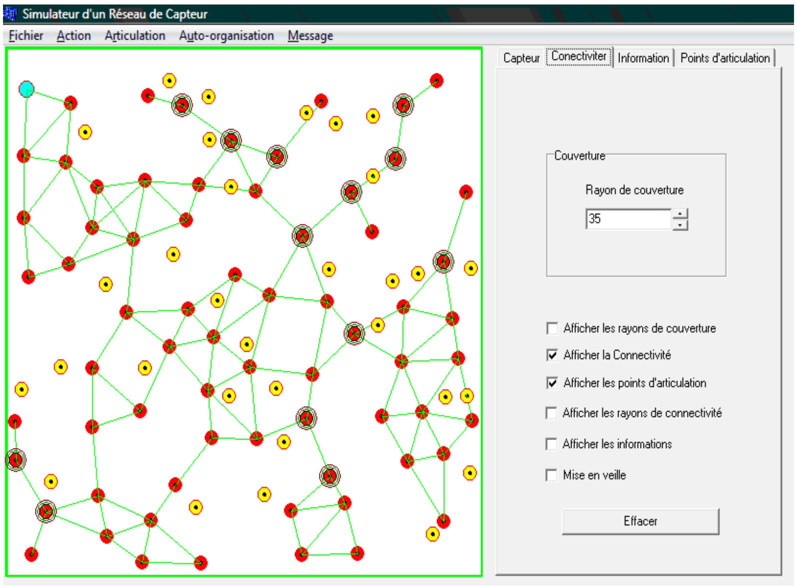
Figure 5. Self-organization around articulation point

8. Simulation

We have tested and validated our algorithm using a simulator implemented in C++, which operates in discrete time. One hundred sensor nodes are distributed randomly on a surface without obstacles. Adjacent nodes at a distance R_c can communicate to form a unit disk graph. The result in figure 6 shows the detection of articulation points (the points surrounded by circles).

A self-organization of the network around the articulation points can increase the degree of network connectivity, the disappearance of the articulation points and finally a fault tolerant network.

We have also simulated the detection of certain targets deployed on the same surface (see figure 9). Consequently any event distant to a sensor with radius r_s will be captured. Figures 10 and 11 give us an idea of the strength of ties between coverage targets. As can be seen in figure 10, every target is covered by at least 2 sensors, ensuring a fault tolerant network. In other words even if some sensors fail there are always other sensors able to provide coverage.



Rayon de couverture: 35

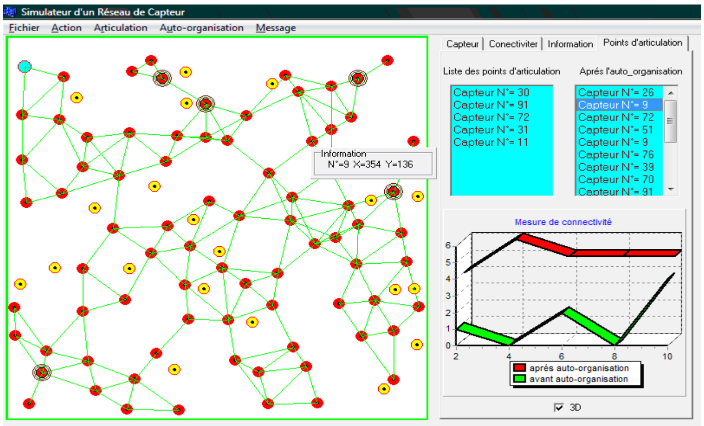
☐ Afficher les rayons de couverture
☑ Afficher la Connectivité
☑ Afficher les points d'articulation
☐ Afficher les rayons de connectivité
☐ Afficher les informations
☐ Mise en veille

Figure 6. Articulation point detection

After launch of self-organization algorithm, in first iteration some of articulation points are disparate by wake-up or move redundant nodes near articulation points. Following screenshot illustrate this.

Per example: the node number 26 which was the articulation point has become a normal node after self-organization. The degree of connectivity around the point of articulation is increased, as shown in the figure 7; the green graph shows the connectivity before self-organization, the red graph shows the connectivity after self-organization. For next's iterations of self- organization we see the same for nodes number 9, 30, 31, 11 and 72. In the last iteration of self-organization, on notice that it remains one articulation point unresolved. As shown in figure 8, graphs of the degree of connectivity are the same.

Figure 11 shows us results for the statistical state of target coverage. This highlights the heavily covered targets, indicating increased fault tolerant areas. Those that are poorly covered may require a reconfiguration or self-organization of the network.

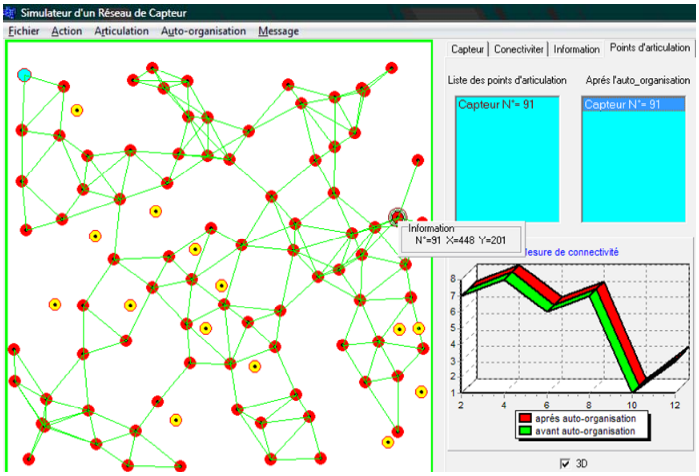


Information
N°=9 X=354 Y=136

Liste des points d'articulation
Capteur N°= 30
Capteur N°= 91
Capteur N°= 72
Capteur N°= 31
Capteur N°= 11

Aprés l'auto_organisation
Capteur N°= 26
Capteur N°= 9
Capteur N°= 72
Capteur N°= 51
Capteur N°= 9
Capteur N°= 76
Capteur N°= 39
Capteur N°= 70
Capteur N°= 91

Figure 7. Self-organization after the first iteration



Information
N°=91 X=448 Y=201

Liste des points d'articulation
Capteur N°= 91

Aprés l'auto_organisation
Capteur N°= 91

Figure 8. Self-organization after the last iteration

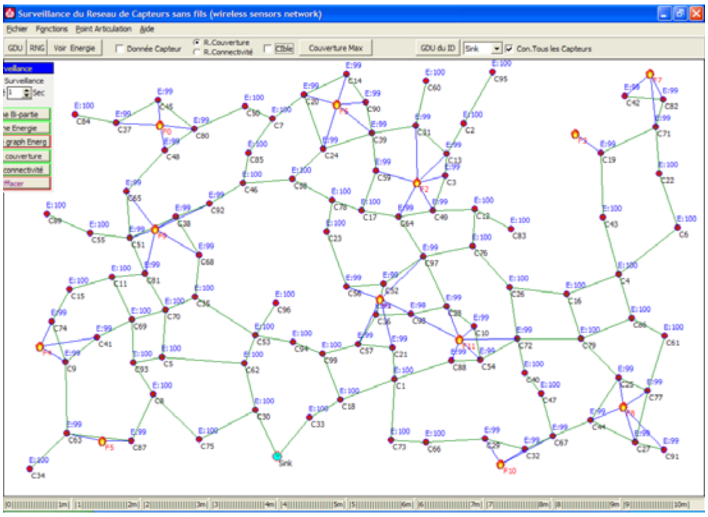


Figure 9. Deployment and coverage targets

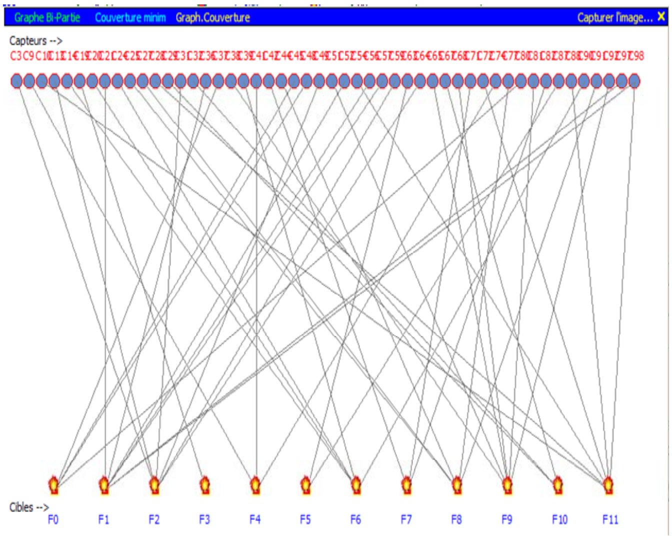


Figure 10. Bipartite graph showing the maximum coverage of the various targets

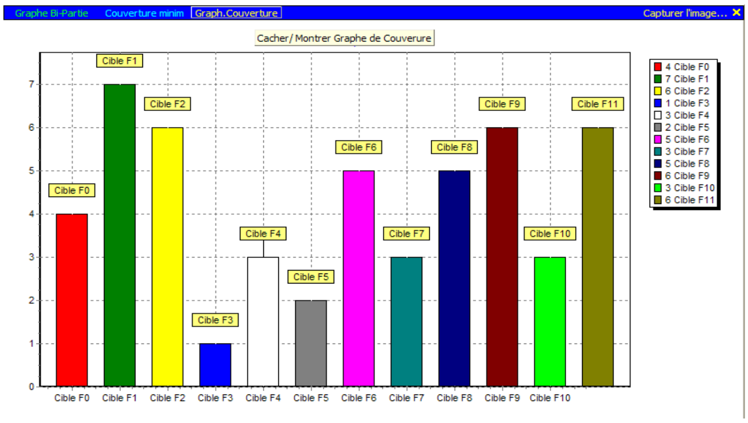


Figure 11. Statistical state of the cover of each target

9. Conclusion and Future Work

In this work we have developed a new monitoring mechanism to guarantee strong connectivity in wireless sensors networks. The mechanism monitors sensor connectivity and at any time is able to detect the critical nodes that represent articulation points. Such articulation points are liable to cause portions of the network to become disconnected and we have therefore also developed a mechanism for self-organization to increase the degree of connectivity in their vicinity, thereby increasing fault tolerance. Since connectivity is closely related to the coverage of targets, we have also developed a way to monitor the robustness of the coverage between fixed targets and sensor nodes. The main advantage of our approach is the ability to anticipate disconnections before they occur. We are also able to reduce the number of monitoring node and assume mechanisms for fault tolerance by auto organization of nodes to increase connectivity. Finally, we have demonstrated the effectiveness of our approach and algorithms with satisfactory results obtained through simulation. Our future work will focus on the use of distributed algorithms, implementation and evaluation of the self organization algorithm and consideration of other relevant network parameters such as optimal coverage, energy and so on.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci, "Wireless Sensor Networks: A Survey.", Computer Networks, vol. 38, no.4, pp. 393-422, 2002.

[2] L. Kleinrock and J. Silvester. "Optimum transmission radio for packet radio networks or why six is a magic number. In National Telecommunications Conference, Birmingham, Alabama, pages 4.3.2–4.3.5, December 1978.

[3] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor networks topologies" IEEE Transactions on Mobile Computing, vol. 3, no. 3, pp. 272–285, 2004.

[4] N. Li and J. C. Hou, "Improving connectivity of wireless ad hoc networks", in MOBIQUITOUS '05: Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. Washington, DC, USA: IEEE Computer Society, 2005, pp. 314–324.

[5] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Data muling over underwater wireless sensor networks using an autonomous underwater vehicle.", in IEEE International Conference on Robotics and Automation (ICRA), 2006, May 15- 19 2006, pp. 2091–2098.

[6] G. Yang, L.-J. Chen, T. Sun, B. Zhou, and M. Gerla, "Ad-hoc storage overlay system (asos): A delay-tolerant approach in manets.", in Proceeding of the IEEE MASS, 2006, pp. 296–305.

[7] N. Rao, W. Qishi, S. Iyengar, and A. Manickam, "Connectivity-through-time protocols for dynamic wireless networks to support mobile robot teams.", in IEEE International Conference on Robotics and Automation (ICRA), 2003, vol. 2, Sept 14-19 2003, pp. 1653–1658.

[8] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin Highly-Resilient, "Energy-Efficient Multipath Routing in Wireless Sensor Networks.", Mobile Computing and Communications Review, 1(2), 1997.

[9] D. Spanos and R. Murray, "Motion planning with wireless network constraints.", in Proceedings of the 2005 American Control Conference, 2005, pp. 87–92.

[10] D. Desovski, Y. Liu, and B. Cukic. "Linear randomized voting algorithm for fault tolerant sensor fusion and the corresponding reliability model.", In IEEE International Symposium on Systems Engineering, pages 153–162, October 2005.

[11] A. Boukerche, "Handbook of Algorithms and Protocols for Wireless and Mobile Networks", Chapman CRC/Hall, 2005.

[12] N. Li and J. C. Hou. "FLSS: A Fault-Tolerant Topology Control Algorithm for Wireless Networks.", In Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, pages 275–286, 2004.

[13] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus. "Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance.", In Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, pages 309–319, 2005.

[14] Bahramgiri, M., Hajiaghayi, M., and Mirrokni, "Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks.", 2002.

[15] N. Li, J. Hou, and L. Sha. "Design and analysis of an mst-based topology control algorithm." , In Proceedings of the IEEE INFOCOM, 2003.

[16] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Chih-Wei Yi. "Fault tolerant deployment and topology control in wireless networks.", In Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc), pages 117.128, 2003.

[17] Michaël Hauspie , "Contributions à l'étude des gestionnaires de services distribués dans les réseaux ad hoc ", Thèse de doctorat, Université des Sciences et Technologies de Lille, 2005.

[18] Bruno Courcelle, " Introduction à la théorie des graphes :Définitions, applications et techniques de preuves ", Université Bordeaux 1, LaBRI (CNRS UMR 5800), 20 Avril, 2004.

[19] R. Tarjan., "Depth First Search and linear graph algorithms.", SIAM Journal of Computing, 1:146_160, 1972.

[20] Wies law Zielonka , "Algorithmique ", LIAFA, Université Denis Diderot, Septembre 2006.