J Inf Process Syst, Vol.13, No.1, pp.128~140, February 2017
https://doi.org/10.3745/JIPS.03.0062

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Self-Identification of Boundary's Nodes in Wireless Sensor Networks

Kouider Elouahed Moustafa* and Haffaf Hafid*

Abstract

The wireless sensor networks (WSNs) became a very essential tool in borders and military zones surveillance, for this reason specific applications have been developed. Surveillance is usually accomplished through the deployment of nodes in a random way providing heterogeneous topologies. However, the process of the identification of all nodes located on the network's outer edge is very long and energy-consuming. Before any other activities on such sensitive networks, we have to identify the border nodes by means of specific algorithms. In this paper, a solution is proposed to solve the problem of energy and time consumption in detecting border nodes by means of node selection. This mechanism is designed with several starter nodes in order to reduce time, number of exchanged packets and then, energy consumption. This method consists of three phases: the first one is to detect triggers which serve to start the mechanism of boundary nodes (BNs) detection, the second is to detect the whole border, and the third is to exclude each BN from the routing tables of all its neighbors so that it cannot be used for the routing.

Keywords

Boundary Recognition, Military Applications, Military Surveillance, Wireless Sensor Network

1. Introduction

Sensors permit to link the physical world with the digital environment. The evolution of wireless technology has led to the development of various derived architectures, such as cellular networks and wireless local networks. During the last decade, a new architecture has emerged the wireless sensor networks (WSNs).

WSN consists of a set of nodes able to communicate via wireless links. The main role of WSN is to collect data from the environment around the sensors and route them to a central processing station (Sink). WSNs are often considered as the successors of ad-hoc networks. Due to their ability to satisfy real needs, WSNs have been introduced in an increasing number of application domains. The need for continuous monitoring of a specific environment is important in various human activities like industrial processes, monitoring of habitat, agriculture, natural resources management, health monitoring, and reaction to disasters, and much more in the military domain. All these application domains adopt WSN technology.

However, the development of WSNs still faces obstacles that are real challenges for scientific research,

Manuscript received July 27, 2015; accepted January 19, 2016.

Corresponding Author: Kouider Elouahed Moustafa (moustafakouider@gmail.com)
* Dept. of Computer Sciences, University of Oran1 Ahmed Benbella, Oran, Algeria (moustafakouider@gmail.com, haffaf.hafid@univ-oran.dz)

among which energy constraint [1] because WSN operate with limited batteries.

The choice of the energy design strategy depends on the type of application in order to ensure the required efficiency. Another challenge is self-management [2] because in many applications WSN must operate in remote areas and wild environments while they need to maintain their efficiency. WSN needs also to guarantee transmission power for the network to function properly. It is important to highlight the fact that WSN has to preserve the sensitive information collected (military applications) from malicious intrusions or attacks, which is a security's problem [3].

Propagation and delivery of data in a WSN is the most important feature of the network. Routing protocols for WSN have been studied extensively, and several studies have been published [4]. The methods can be classified according to either network topology criteria or to establishment of the road. In some realistic situations, existing routing protocols often become inoperative such is the case where there is a presence of geographical voids resulting from the random deployment of sensor nodes, and this is a serious drawback.

Any routing protocol that is capable of detecting the nodes located on the edge of the network will have the ability to route the packets while avoiding blocking or losing data, so if each node has the information that one of those neighbors is a border node, it can make a better choice for the next hop.

In our paper, we deal with the problem of detecting network's border. Therefore, it was found that there is a need for a reliable method to detect the network borders as a support for routing protocols.

The deployed WSN for monitoring applications seeks to identify the nodes forming the network edge. This task should be accomplished before starting any other activity on the deployed area.

Compared to previous methods, we will show that our method is able to provide better performance by reducing failure routing protocols, reducing packet loss and increasing network life time. In addition, we highlight the border detection mechanism in WSN using local geographical information. This document is presented as follows: in Section 2, we present some related works on military applications and boundary detection. In Section 3, we expose our solution for the problem of boundary detection. Section 4 is devoted for the empirical results using Omnet++/Castalia simulator and finally conclusion is given in Section 5.

2. Related Works

Previous works are devoted to both boundary nodes identification and border recognition. Several boundary nodes' solutions are based on geometric approaches. Some of them rely on the topology construction and other proposed distributed algorithms. The approach described in [5] is an efficient distributed boundary detection algorithm using local connectivity information, where each node constructs its 2-hop to make a decision on whether this node is on the boundary or not. Wu et al. [6] present a new approach based on local neighboring information. Each node builds the shortest tree path and selects cuts into it, then finds into this the cycle enclosing all holes. The last step in this approach is to select a symbolic node to construct the boundary of the network.

In [7], a new approach is proposed based on two distributed mechanisms—Hole Detection Algorithm (HDA) and Boundary Detection Algorithm (BDA)—to detect nodes situated on the boundaries, it is based on splitting the range of communication (360°) into 4 quadrants of 90° each one then checks the existence of at least 1-hop neighbor within the range of an angle less than 90°. In our case we were inspired by the distributed mechanism presented in [8] to identify the nodes forming the network's border. The network boundary discovery (NBD) is used to identify boundary nodes and computes the external void center as well as the radius. The principle of this method is to choose a node which will trigger the mechanism by routing packet network discovery (ND) around the large void in one direction. This mechanism is efficient but its performance is limited in these cases: 1) critical time monitoring applications, 2) security problems such as malicious intrusions, 3) a huge number of boundary nodes. It is noteworthy that the boundary detection process is time-consuming.

2.1 Problematic

WSNs are widely used in the military applications such as zones' monitoring and area battlefields. The major challenge of these applications is to get real-time information about the environment. Receiving alarms, images or data of monitored areas that have to be transmitted are some examples.

However this task becomes more challenging in case of using geographic routing protocols. The geographic void caused by the random deployment of nodes leads to packets' loss. Those voids are created inside the network or on its outer edge. In this case the void will be considered as big communication hole.

2.2 Military WSNs Applications

Military communications have to be maintained in all situations so that they should resist to jamming and have to ensure end-to-end delivery packets.

There are many classes of military applications such as self-healing land mines (SHLM) [9], aerostat acoustic payload for transient detection (AAP) [10], soldier detection and tracking (SDT) [10], and the perimeter protection (PP) [11]. In this section we will first explain the PP class and then present our mechanism as solution for this applications' class.

After deployment, to monitor the perimeter we must first detect the network's outer edge. Also several holes are raised inside the network and that provides other borders to recognize.

We can define the network boundary by a graphical concept (Fig. 1): "*The boundary of a sensor network is a complex spatial property even when a straight-line embedding of this graph into the two dimensional space is known*" [12] or by the same definition as Khan et al. [13]: "*Boundary of a sensor network means the nodes residing on the edges of a sensor network or of the holes inside the network.*"

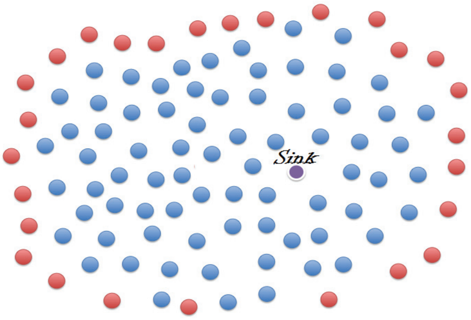


Fig. 1. The network's boundary.

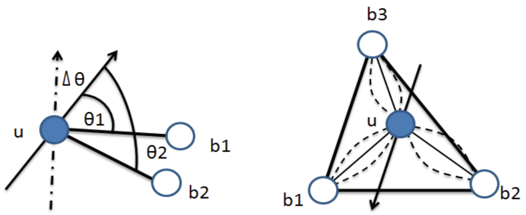
2.3 Localization of Nodes in WSNs

The localization of nodes in WSNs is the source of the information for gets geographic locations.

The boundary detection algorithms' results depend on localization's information. Several localization protocols are proposed [14,15] which are based on circular radio range, symmetric radio connectivity, additional hardware such as the global positioning system (GPS), lack of obstructions, lack of line-of-sight, absence of multipath and flat terrain. It can be classified in three classes: geometric techniques (multilateration, trilateration), multidimensional (MDS) and finally the area-based techniques (centroid, bounding box).

Those methods are getting location information in two phases, distance estimation and distance combining. The most popular are: received signal strength indicator (RSSI), time based methods (ToA, TDoA), angle-of-arrival (AOA), hyperbolic trilateration and maximum likelihood estimation.

For example, AOA is the angle between the propagation direction of the wave and a reference direction (the orientation), it is represented by degrees clockwise direction and the orientation is pointing to the north. The localization of nodes is gotten by triangulation with or without orientation Fig. 2.



(a) (b)

Fig. 2. Triangulation in angle-of-arrival (AOA) localization. (a) Localization with orientation information, (b) localization without orientation information [15].

2.4 Boundary Recognition

Boundary recognition algorithms in sensor networks can be classified into three main groups: geometric, statistical, and topological methods [16].

Geometric algorithms: Node's geographic location is calculated by making a device as the GPS. They are based on geometric rules such as Delaunay triangulation which constructs planar graphs using local Delaunay triangulations on neighbor sets [17]. Other methods adopt two types of simplicial complexes called Cech complex and Vietoris–Rips to capture coverage holes. Coverage holes are classified in triangular and non-triangular methods [18].

Statistical algorithms: in this kind of algorithms, boundary nodes recognition is performed without need to location information. Due to the uniform sensor nodes distribution on the sensing area, statistical properties and probabilistic rules are used [19]. Many works belong to this category.

Topological algorithms: they use the topological properties and connectivity information which are shared and exchanged with neighbors [20]. In our work, we have been interested by algorithms proposed by Aissani et al. [21]. Void boundary discovery (VBD) and NBD based on the connectivity information with one hope neighbors on the outer and the inner boundaries. The essence of the NBD is to indicate a fictive destination which will be a starter for the mechanism by routing a packet named network discovery (ND) around the large void in one direction; however, VBD identify all nodes forming an internal void in the network and then calculates its center and radius. The principle of this method is to route a void discovery (VD) packet around the boundary. This propagation mechanism stops once the packet has reaches a full turn around the void.

3. Our Proposition

The solution presented in this paper belongs to the topological methods. The major problem in geographical routing protocols is that all boundaries detection solutions are started after whole detection or blocking packets, see Fig. 1.

Our proposition is to detect and select the nodes situated on the boundary of the network before starting any routing or surveillance task.

The specificity here is to detect more than one starter node and bidirectional routing scheme [22]. This method is decomposed into three phases. The first one is the process initiation, the second one has been devoted to the network's border discovery, and the third one is dispensing all border nodes from the routing task.

3.1 Process Initiation (Self-Identification)

Border nodes of the network are located at the farthest distance from the sink compared with its one hope neighbors. Each node in the network has its own neighbors list. In this phase, the node initiates the process by identifying itself as starter (or trigger nodes) as follow: in the first step, each node computes its distance from the Sink, then compares it with other distances of its immediate neighbors. The node that is farthest from the Sink represents a Trigger. The latter is called a Starter. To accomplish this task, we add a Boolean field (isBoundary) that is initially, set to false. A node is called Starter when the isBoundary field is set true (see Algorithm 1 and phase 1 in Fig. 3).

Algorithm 1. Self-identification

```
Begin:
Input: The node (α)'s coordinates (Xα, Yα),Sink coordinates  and node α's
        neighbor's coordinates(β) within one communication hop.
Output: Identification State as Boolean: (IsBoundaryNodeα) True or False.
Distance = 0; Index=0; IsBoundaryNode$_{\alpha}$= True;
// Calculate the distance between the local node and the sink
Distance = Dist (coordinates$_{\alpha}$,coordinates$_{Sink}$)
// Compare the calculated distance with the distance between neighbors and the sink
For each node β in α's one communication hop do
    If (Distance <Dist$_{\beta}$);
    Then
    //If there is one neighbor farther from the sink than this node we break
       IsBoundaryNode$_{\alpha}$= FALSE;
         Break;
    End;
    Index ++;
End.
```

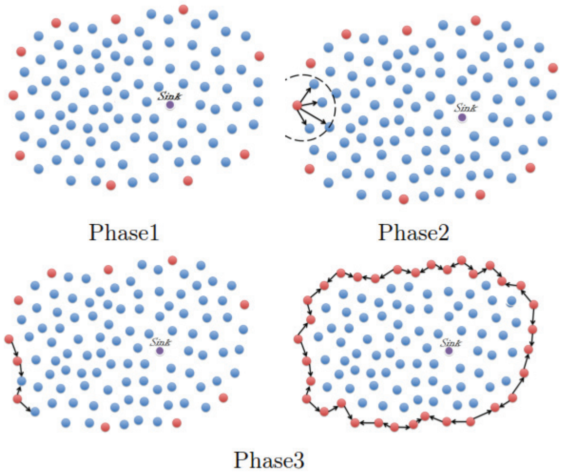


Fig. 3. Phases of self-identification of boundaries.

3.2 Discovery of the Network Border

When a border node (BN) is self-identified, it must trigger the discovery mechanism and bypass the external boundary.

3.2.1 Construction of the neighboring groups L and R (left and right)

Each node has to sort out both its right neighbors (R, right neighbor group) and left neighbors (L, left group of neighbors). To perform this, we use a geometric method to found nodes (whose coordinates are known) located below and above a specific line. A is on the left of the line If sin(α) < 0, where $\alpha = (A\ \widehat{Bn\ Sink})$. B is on the right of the BN, because sin(α) > 0, where $\alpha = (B\ \widehat{Bn\ Sink})$ (Algorithm 2, Fig. 4).

Algorithm 2. Build left & right neighbor tables

```
     Begin:
     Input: The node (α)'s coordinates (Xα,Yα),Sink coordinates and node α's
     neighbor's coordinates(β) within one communication hop.
     Output: Left and right neighbors tables.
    // Calculate the Sin the angle between the two Victors $\overrightarrow{\alpha S}$ and $\overrightarrow{\alpha\beta}$
Calculate ($\mathrm{Sin}(\beta\widehat{\alpha}S)$)
For each (node β in α's one communication hop) do
     // If the Sin is greater than 0 then it is right neighbor
        else it is a left one
     If ($\mathrm{Sin}(\beta\widehat{\alpha}S)$>0) then
           Add to Right table(β);
        Else
Add to Left table(β);
End;
End.
```

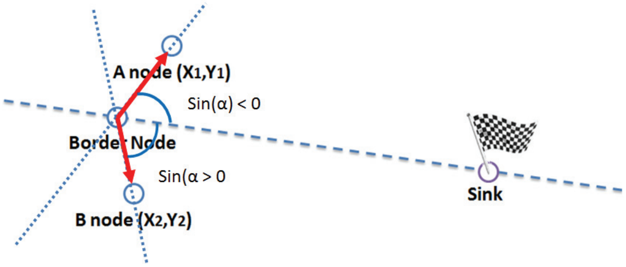


Fig. 4. Geometric method of neighbor positioning.

Once the two sub-lists of neighbors L and R are formed (see Fig. 5), the trigger node creates a BP (boundary packet) package by including a simple message boundary node. To accelerate this process, the trigger node will send the packet to its immediate neighbors choosing the farthest from the Sink (border node) right and left.

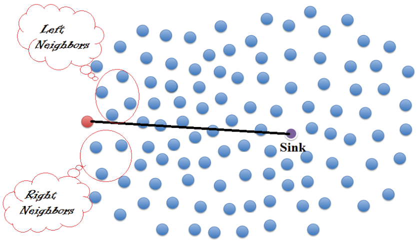


Fig. 5. Left and right neighbors groups.

3.2.2 Sending boundary packet

Whenever the trigger node has found the BP's next destinations (right and left), it must multicast the BP. See the flowcharts depicted in Fig. 6.

This packet contains two destination addresses (Table 1), that gives the possibility for the starter to send the same packet to the left and the right neighbor in the same time (see phase 2 in Fig. 3).

Table 1. The boundary packet

Source ID	Left destination	Right destination	………………………

In a wireless network, when a node 'A' sends a packet to another node 'B', all nodes located on the same radio range than 'A' will automatically receive the same packet. The first task is then to check if the received packet is intended to the node or not by looking at the destination addresses specified on the packet.

We add a column in the routing table. This new column contains a Boolean field state initially set to true indicating that the node will participate to the routing task. The node which receives the BP compares its id with the destination address fields. If its id is equal to the destination address, it deduces that it is a border node and sets its isBoundary field to true. Otherwise, it means that the sender is a border node, so it must update the sender neighbor's state field in the routing table to "false".

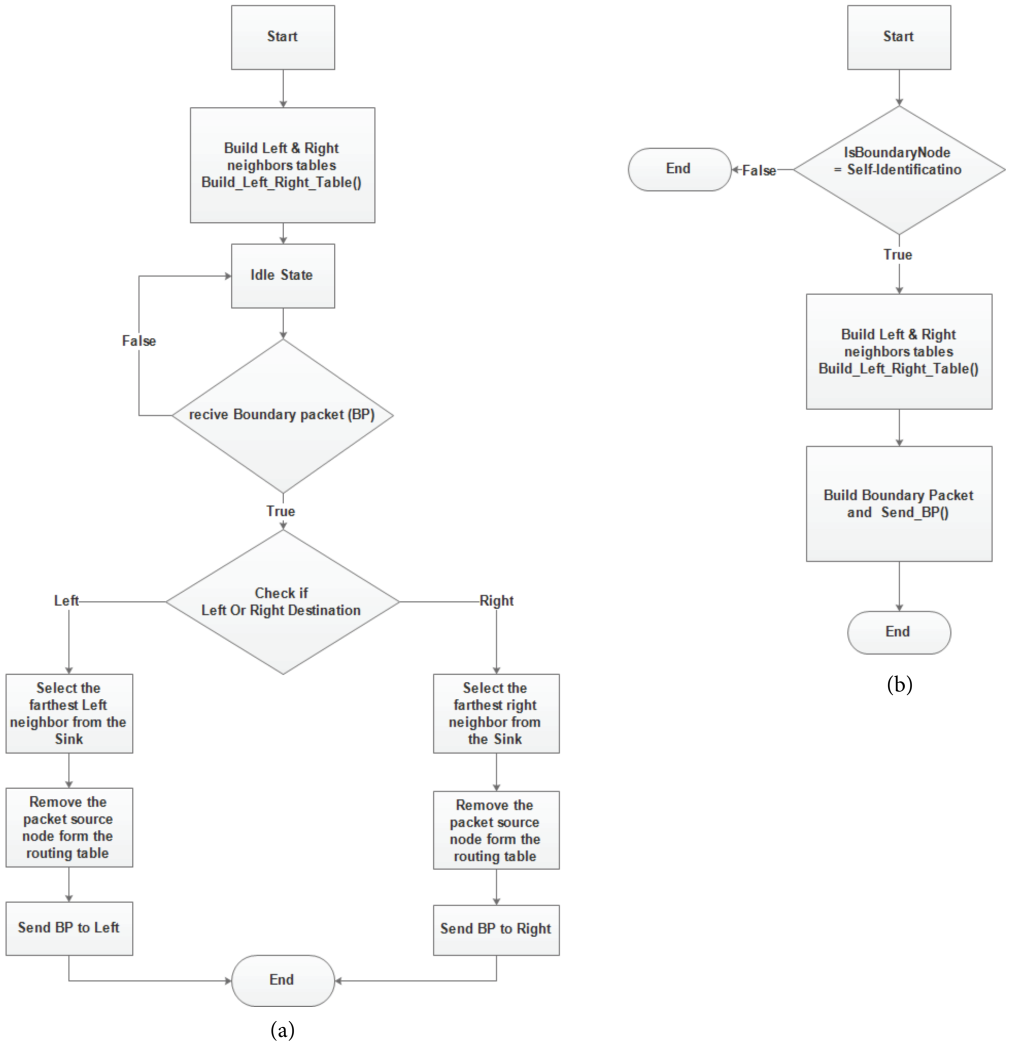

Start
Build Left & Right neighbors tables Build_Left_Right_Table()
Idle State
False
recive Boundary packet (BP)
True
Check if Left Or Right Destination
Left
Right
Select the farthest Left neighbor from the Sink
Remove the packet source node form the routing table
Send BP to Left
Select the farthest right neighbor from the Sink
Remove the packet source node form the routing table
Send BP to Right
End
(a)
Start
IsBoundaryNode = Self-Identificatino
False
End
True
Build Left & Right neighbors tables Build_Left_Right_Table()
Build Boundary Packet and Send_BP()
End
(b)

Fig. 6. Flowcharts explain programs of boundary node and the receiver node. (a) The receiver node program, (b) the boundary node (Starter) program.

3.3 Exclusion of Border Nodes from the Routing

At this stage, the network border is detected (phase 3 in Fig. 3) and border nodes are identified. During the routing process, network nodes will forward packets only to nodes with “true” routing state in the routing table (Table 2). This exclusion does not preclude the border nodes to play their role of sentinel against malicious intrusions or any other task devoted to the border.

Table 2. Routing table

Field	Function
Node Id	get the node ID
Position	get the (X,Y) position coordinates
State	‘True’ and ‘False’

Algorithm 3. Main algorithm

```
Begin:
Input: The node (α)'s coordinates (Xα,Yα),Sink coordinates and node α's
        neighbor's coordinates(β) within one communication hop.
Output:   all the boundary nodes
Self - Identification ();
If (!self_Identification()) then  //test if the node is starter or not
If (receive (packet))
If (BP.RightDistId == Self.Id ) then// if the node is right destination
SelectNext_right_Dist();
Create (BP, DistId);
Send (Packet, DistID);
NTable.SrcId.state = False;
Else if (BP.LeftDistId == Self.Id) then //if the node is left destination
SelectNext_left_Dist(); //select from neighbors the next destination
Create (BP, DistId); // create the boundary packet with the next destination
Send (Packet, DistID);
NTble.SrcId.State = False;//remove the source of the BP from the routing table
Else
NTble.SrcId.State = False;
    end;
    End else
IsBoundary = True;
BuildLeftRightTable();;
BuildBP ();
End.
```

Algorithm 3 summarizes our solution. It encompasses all the phases and functions used for select and detect in the same time, all nodes situated on the outer boundary of the network.

4. Experimentation

To evaluate the performance of the proposed distributed algorithm, we have implemented it under OMNeT++/Castalia Simulators specifically Castalia version 3.0 on Lunix platform of Ubuntu and more of details of simulation parameters are presented into Table 3.

Castalia provides a modular view of network's nodes [23], a realistic radio and channel models and C++ as a programming language.

It is based on the OMNeT++ [23,24], so to use it, the OMNeT++ simulation platform must be available. The Castalia code source structure is hierarchical. Each module is a directory that contains a C++ code to describe the behavior of the module. OMNeT++ includes simple and compound modules that communicate with each other by sending messages that represent packets.

Table 3. Simulation parameters

Simulation parameter	Function
Number of nodes	150
Topology	Random, circle, square
Size of sensing area	200×200
Routing protocol	GPSR

The structure of a module is defined by the user in the OMNeT++ topology language NED. Also OMNeT++ environment includes a graphical editor. For having an acceptable presentation, we have used video sensor model with 150 nodes.

Fig. 7 presents the graphical results of the outer boundary detection applied on a network prototype of 150 nodes, the boundary nodes who are selected as starters of the mechanism are colored in red and others are still in blue. The first phase has detected several nodes as starters depends to the topology of the network in this phase every node starter or BN will send the boundary packet in the left and the right direction. Notice that the used protocol is Greedy Perimeter Stateless Routing Protocol. The latter is a suited tool in our case (local information based) because of when a packet reaches a region where greedy forwarding is impossible; the algorithm recovers by routing around the perimeter of the region [25].

The results shown on this figure are the boundaries detected with our algorithm; we have tested different topologies with some concave borders, and showed that all nodes situated on the boundary are detected. We have analyzed theoretically the speed of our mechanism relative to the number of starter nodes and Fig. 8 illustrates the result.

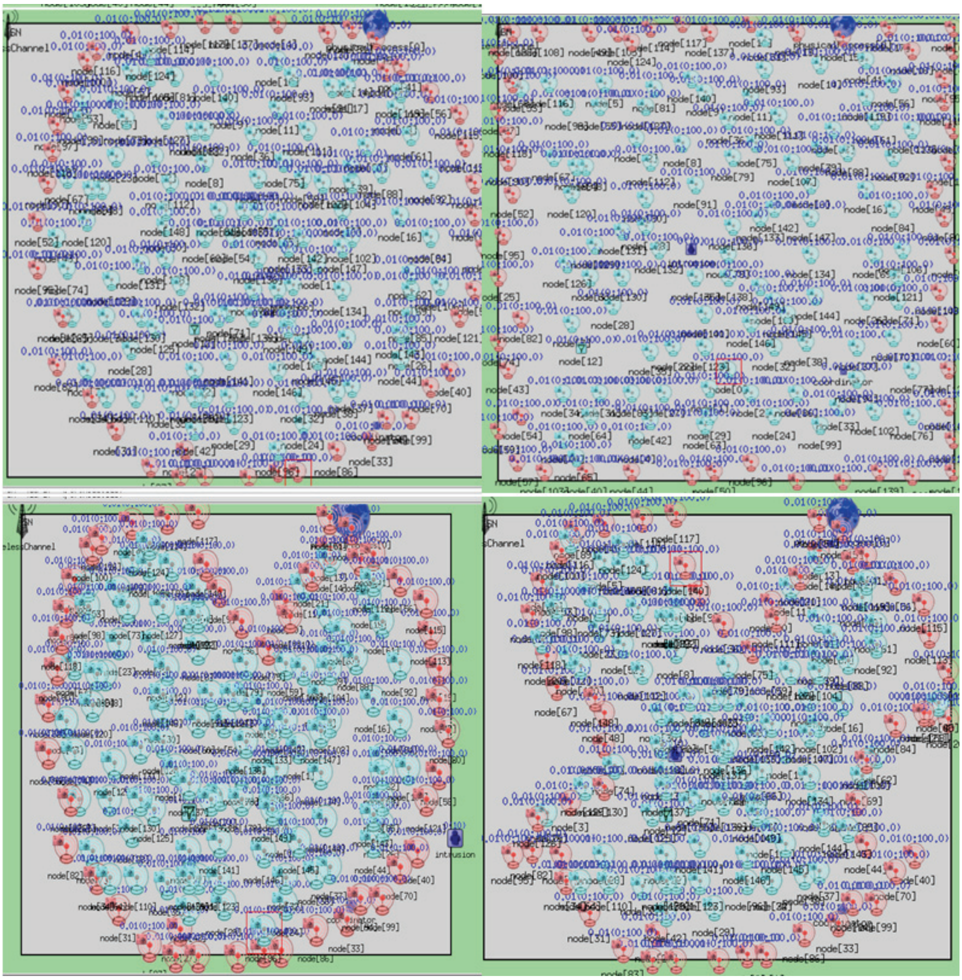


Fig. 7. The recognized boundaries tested on different topologies.

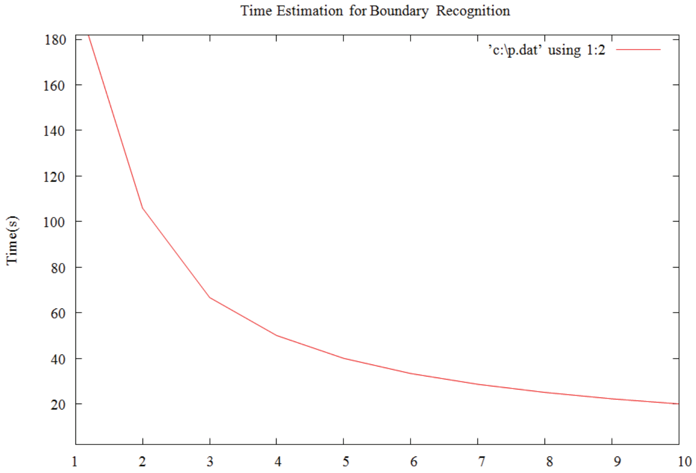


Fig. 8. Theoretical plot of time estimations relative to the number of starter nodes.

5. Conclusion

By early discovery of the border of the network, we avoid failure of routing protocols; we increase the lifetime of the network. We presented a new approach to detect nodes situated on the external boundaries for the network in the WSNs context. In our method, increasing starter nodes improves the network performance in boundaries detection problem and provides the WSNs ability to give a good results in surveillance applications specially in military domains. Since we are interested in military applications and border surveillance based on WSNs, our future work will focus on real application by building a real prototype for survey military zones and battlefields. Also we shall apply our method to recognize internal voids.

Acknowledgement

The authors are grateful to Belhadia Amar, Dahane Amine and all anonymous referees for their insightful comments and valuable suggestions, which greatly improved the quality of the paper.

References

[1] T. T. Vu, V. D. Nguyen, and H. M. Nguyen, "An energy-aware routing protocol for wireless sensor networks based on k-means clustering," in *AETA 2013: Recent Advances in Electrical Engineering and Related Sciences*. Heidelberg: Springer, 2014, pp. 297-306.

[2] 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.

[3] A. S. K. Pathan, H. W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT)*, Phoenix Park, Korea, 2006.

[4] J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6-28, 2004.

[5] Khan, H Mokhtar, and M. Merabti, “A new self-detection scheme for sensor network boundary recognition,” in *Proceedings of IEEE 34th Conference on Local Computer Networks*, Zurich, Switzerland, 2009, pp. 241-244.

[6] G. X. Wu, T. L. Huang, P. Liu, and X. Y. Zhou, “Boundary recognition by topological methods in wireless sensor networks,” in *Proceedings of IEEE Conference Anthology*, China, 2013, pp. 1-5.

[7] V. Sukumaran and T. P. Saravanabava, “Modified sensor deployment algorithm for hole detection and healing using NS2,” *International Journal of Engineering Research and Applications*, vol. 4, no. 4, pp. 43-50, 2014

[8] M. Aissani, S. Bouznad, S. E. Allia, and A. Hariza, “Efficient forwarding approach on boundaries of voids in wireless sensor networks,” *International Journal on Advances in Telecommunications*, vol. 5, No. 3-4, pp. 2012.

[9] W. Merrill, L. Girod, B. Schiffer, D. McIntire, G. Rava, K. Sohrabi, F. Newberg, J. Elson, and W. Kaiser, “Defense systems: self healing land mines,” in *Wireless Sensor Networks: A Systems Perspective*. Boston, MA: Artech House; 2005, pp. 273-288.

[10] M. P. Durisic, Z. Tafa, G. Dimic, and V. Milutinovic, “A survey of military applications of wireless sensor networks,” in *Proceedings of 2012 Mediterranean Conference on Embedded Computing (MECO),* Bar, Montenegro, 2012, pp. 196-199.

[11] R. Dulski, M. Kastek, P. Trzaskawka, T. Piatkowski, M. Szustakowski, and M. Zyczkowski, “Concept of data processing in multisensor system for perimeter protection,” in *Proceedings of SPIE 8019: Defense, Security, and Sensing*. Bellingham, WA: International Society for Optics and Photonics, 2011.

[12] Saukh, R. Sauter, M. Gauger, P. J. Marron, and K. Rothermel, “On boundary recognition without location information in wireless sensor networks,” in *Proceedings of International Conference on Information Processing in Sensor Networks*, St. Louis, MO, 2008, pp. 207-218.

[13] Khan, H. Mokhtar, and M. Merabti, “A survey of boundary detection algorithms for sensor networks,” in *Proceedings of the 9th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, UK, 2008, pp. 1-5.

[14] A. Boudhir, B. Mohamed, and B. A. Mohamed, “New technique of wireless sensor networks localization based on energy consumption,” *International Journal of Computer Applications*, vol. 9, no. 12, pp. 25-28, 2010.

[15] A. Pal, “Localization algorithms in wireless sensor networks: current approaches and future challenges,” *Network Protocols and Algorithms*, vol. 2, no. 1, pp. 45-73, 2010.

[16] K. Y. Hsieh and J. P. Sheu, “Hole detection and boundary recognition in wireless sensor networks,” in *Proceedings of 2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, Tokyo, Japan, 2009, pp. 72-76.

[17] Zhang, Y. Zhang, and Y. Fang, “Detecting coverage boundary nodes in wireless sensor networks,” in *Proceedings of 2006 IEEE International Conference on Networking, Sensing and Control*, Ft. Lauderdale, FL, 2006, pp. 868-873.

[18] F. Yan, P. Martins, and L. Decreusefond, “Connectivity-based distributed coverage hole detection in wireless sensor networks,” in *Proceedings of IEEE Global Telecommunications Conference*, Houston, TX, 2011, pp. 1-6.

[19] P. Antil and A. Malik, "Hole detection for quantifying connectivity in wireless sensor networks: a survey," *Journal of Computer Networks and Communications*, vol. 2014, article ID. 969501, 2014.

[20] A. Varga, “The OMNeT++ discrete event simulation system,” in *Proceedings of the European Simulation Multiconference (ESM2001),* Prague, Czech Republic, 2001, pp. 1-7.

[21] M. Aissani, S. Bouznad, A. Hariza, and S. E. Allia, “An effective mechanism for handling open voids in wireless sensor networks,” in *Proceedings of the 5th International Conference on Sensor Technologies and Applications*, Nice, France, 2011, pp. 24-29.

[22] N. Senouci, M. K. El Ouahed, and H. Haffaf, “Detecting boundary nodes in WSN,” in *Proceedings of the International Conference on Wireless Networks (ICWN)*, Las Vegas, NV, 2014, pp. 47-52.

[23] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*. Heidelberg: Springer, 2010, pp. 35-59.

[24] A. Boulis, "Castalia: revealing pitfalls in designing distributed algorithms in WSN," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, Sydney, Australia, 2007, pp. 407-408.

[25] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, MA, 2000, pp. 243-254.

Kouider Elouahed Moustafa

He received his license from UHBC on artificial intelligence in 2009 and his Master degree on databases and distributed information systems in 2011 at University of Oran, Algeria. He is a PhD student since 2011 at the University of Oran in the R.I.I.R Laboratory, Graduate School of Advanced Data Models and Emerging Networks. His main research area is on the wireless sensor networks, military applications and also on voids avidness in geographic routing protocols.

Haffaf Hafid

He obtained Doctor degree in computer Science in 2000 and is a senior lecturer at the University of Oran, Es-Senia, Algeria. He actually heads the R.I.I.R Laboratory at computer science department, Oran University. His researchers concern different domain as Automatic control and diagnosis, optimization reconfiguration using matroid theory, system of system approaches and their applications in Bond graph and monitoring. He has many collaborations projects with European laboratory; Polytech Lille where he worked in intelligent transport systems infrastructures and LIAUPau (France) in the domain of wireless sensor networks (CMEP project).