



Detecting Gaps and Voids in WSNs and IoT Networks: the Angle-based Method

Madani Bezoui, Ahcène Bounceur, Loïc Lagadec, Reinhardt Euler,
Hammoudeh Mohammad, Abdelkader Laouid, Abdelkamel Tari

► To cite this version:

Madani Bezoui, Ahcène Bounceur, Loïc Lagadec, Reinhardt Euler, Hammoudeh Mohammad, et al..
Detecting Gaps and Voids in WSNs and IoT Networks: the Angle-based Method. International
Conference on Future Networks and Distributed Systems (ICFNDS), Jun 2018, Amman, Jordan.
10.1145/3231053.3231089 . hal-01836035

HAL Id: hal-01836035

<https://hal.science/hal-01836035>

Submitted on 14 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Detecting Gaps and Voids in WSNs and IoT Networks: the Angle-based Method

Madani Bezoui
Université de Boumerdes
Department of Mathematics
Boumerdes, Algeria
Madani.Bezoui@gmail.com

Ahcène Bounceur
Lab-STICC CNRS UMR 6285
Université de Bretagne Occidentale
Brest, France
Ahcene.Bounceur@univ-brest.fr

Loïc Lagadec
Lab-STICC CNRS UMR 6285
ENSTA Bretagne
Brest, France
Loic.Lagadec@ensta-bretagne.fr

Reinhardt Euler
Lab-STICC CNRS UMR 6285
Université de Bretagne Occidentale
Brest, France
Reinhardt.Euler@univ-brest.fr

Mohammad Hammoudeh
Manchester Metropolitan University
School SCMDT
Manchester, UK
M.Hammoudeh@mmu.ac.uk

Abdelkader Laouid
LIMED Laboratory
University of El-Oued
El-Oued, Algeria
Abdelkader-Laouid@univ-eloued.dz

Abdelkamel Tari
LIMED Laboratory
University of Bejaia
Bejaia, Algeria
tarikamel59@gmail.com

ABSTRACT

A random deployment of Wireless Sensor Networks (WSNs) is often the basic structure used in the context of fire forest detection, military applications or any situation where the zone-of-interest is not accessible by humans. The main problematic in this kind of deployment is the formation of gaps or voids, which represent a zone which is not covered in the network. This reduces significantly its Quality of Service and can lead to serious problems, like a non-detected starting fire, the presence of unexpected persons or attacks, etc. Therefore, detecting zones that are not covered by the WSN is of great importance. In this paper, we present a new method allowing to detect gaps and voids in WSNs or in IoT networks by using some characteristics of the angles of the polygon formed by the boundary as determined by the D-LPCN algorithm. These angles can be interior or exterior. Characterizing the angles of the polygon formed by these boundary nodes allows to specify whether this boundary is a gap or a void, in case where the obtained polygon is interior. Since D-LPCN is fault-tolerant, the simulation results show that it is possible to use it for the detection of faulty nodes and intrusions.¹

¹This project is supported by the French National Research Agency ANR PERSEPTEUR - REF: ANR-14-CE24-0017.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICFNDs'18, June 26–27, 2018, Amman, Jordan
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6428-7/18/06...\$15.00
<https://doi.org/10.1145/3231053.3231089>

CCS CONCEPTS

• **Networks** → Network algorithms; • **Mathematics of computing** → Graph algorithms; • **Theory of computation** → Computational geometry; • **Computing methodologies** → Distributed algorithms;

KEYWORDS

Wireless Sensor Network, IoT, gap, void, distributed algorithms, polygons, interior and exterior angles, security, D-LPCN

ACM Reference Format:

Madani Bezoui, Ahcène Bounceur, Loïc Lagadec, Reinhardt Euler, Mohammad Hammoudeh, Abdelkader Laouid, and Abdelkamel Tari. 2018. Detecting Gaps and Voids in WSNs and IoT Networks: the Angle-based Method. In *ICFNDs'18: International Conference on Future Networks and Distributed Systems, June 26–27, 2018, Amman, Jordan*. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3231053.3231089>

1 INTRODUCTION AND RELATED WORK

Improving the connectivity of a Wireless Sensor Network (WSN) or of an IoT network [1] is one of the main ways to improve its Quality of Service (QoS) in terms of communication, target detection and cyber-security. In general, this kind of problematic arises as a gap or a void in the case of a random deployment of the network. All the existing methods present solutions avoiding gaps or voids, but most of them do not show how these voids are detected. In such a case, it is not possible to use them reliably and in a secured way especially when a void is caused by faulty or hacked nodes. In [2], [3] and [4], it has been highlighted that a wormhole attack poses a serious threat to ad hoc networks and WSNs.

Also, most of these methods deal mainly with voids in terms of communication, and they make use of greedy algorithms. A few approaches deal with target detection, and they use mainly statistical methods.

In this paper, we present as a complement to the existing methods, a procedure allowing to determine whether there exists a region in the network which is not covered. If that is the case, the location of that zone can be determined. This will help to not only improve the QoS, but also to take appropriate decisions about the existence of that zone.

Since the proposed method deals with the three QoS parameters cited previously, we will use the notation *Gap* and *Void* to designate such a zone in terms of communication and detection. These two terms are defined in Section 2.

In [11], a *right-hand* based method to anticipate voids and a routing protocol are proposed to bypass them. In [5], a method based on the locations of the transmitter node is presented to determine the left or right side of the network to route messages by avoiding voids.

In [7], the authors propose a system avoiding opportunistic voids, which is based on energy variance and depth to achieve energy balance. This technique is used to avoid empty regions and allows to balance energy especially within Underwater Acoustic Sensor Networks.

In [8], a topological approach is introduced that requires only the topology of the network's connectivity, without any prior knowledge of node positions, or network timing. This approach captures the basic topology of deviations and thus locates wormholes tracing the sources leading to such exceptions. Another area of application of the algorithm presented in this work is trap coverage, introduced in [6], where the size of a coverage hole is defined as an indicator of the quality of coverage.

The authors of [5] have classified the existing techniques into two types:

- the right-hand rule which is to share the borders in several communication sessions [10],
- the backpressure rule, in which data packets tend to be pushed back to the upstream nodes for alternative routes [9].

In this paper, we propose a new algorithm based on geometric calculation to determine whether a polygon corresponding to an uncovered zone in the network is interior or exterior. If the polygon is interior then it is possible that it is a gap or a void. Additional calculations on the obtained radio communication polygon or on the area covered by the detection zones of the sensor nodes will be done in order to determine if this interior polygon is a gap or a void.

The remainder of the paper is organized as follows: Section 2 introduces the notions of Gaps and Voids. Section 3 presents the method allowing to determine geometrically if a polygon is interior or exterior. Section 4 is dedicated to the presentation of the proposed algorithm. Simulation results are presented in Section 5. Finally, Section 6 concludes the paper.

2 A VOID AND A GAP

In this section we will define the notions of a void and of a gap and the difference between them. We call a *gap* the area determined by a set of sensor nodes in terms of radio communication, as shown by the light gray area of Figure 1 (a). If this area is greater than a given threshold then it will be considered as a gap. The light gray zone of Figure 1 (b) is not a gap if we assume that its surface is smaller

than a given threshold. In this case, we need to determine the type of the polygon formed, which must be *interior*. We call a *void* the zone which is not covered by the detection area of any sensor node in a gap assuming that the detection radius is smaller than the radio communication range. In this case, we need, in addition, to determine the part of this polygon which is not covered by the detection area of each sensor node forming it, as shown by the dark gray area of Figure 1 (c). However, if we assume that the gray zone of Figure 1 (b) is a gap, this zone is not a void since it is totally covered by the detection zones of its sensor nodes, as shown by Figure 1 (d).

In the following, we will present the main contribution of this paper which is based on a method allowing to determine the nature of a polygon: interior or exterior.

3 INTERIOR AND EXTERIOR POLYGONS

In order to define the notion of a void geometrically, we will first define the notions of *interior polygon* and *exterior polygon*. If we run the D-LPCN [12][13][14] algorithm on the polygon of Figure 2 (a), by starting from node A, then we obtain the polygon shown by Figure 2 (a), where the nodes are visited following their exterior angles. The obtained polygon is exterior. However, if we start D-LPCN from node B, we will obtain the same polygon, shown by Figure 2 (b), but the nodes are now visited following their interior angles. Then the obtained polygon is interior. The main issue in determining interior and exterior polygons is based on a characterization of interior and exterior angles of a polygon. Figures 3 (a) and (b) show that even if we start from the same angle as in Figures 2 (a) and (b), we obtain two polygons of different type.

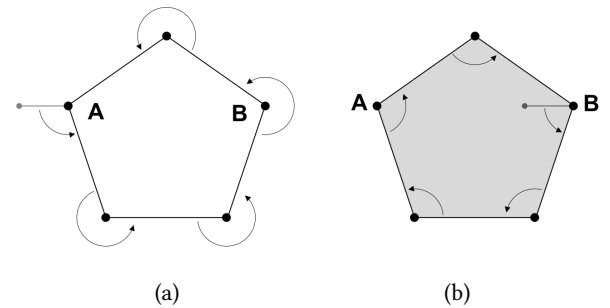


Figure 2: Interior and exterior polygons - Situation 1.

Figure 4 illustrates for three examples what is also known as the *Interior Angle Sum Theorem*, namely that the sum s of interior angles of any polygon with n sides is given by:

$$s = (n - 2) \times 180^\circ \quad (1)$$

This is due to the fact that each polygon with n sides can be divided into $n - 2$ triangles, and we know that the sum of interior angles of a triangle is equal to 180° .

Using Equation (1), we can, therefore, distinguish interior from exterior polygons, where the angles are determined with the D-LPCN algorithm.

One of the advantages of this method is that it works even if there are inside an interior polygon some sub-graphs that are connected

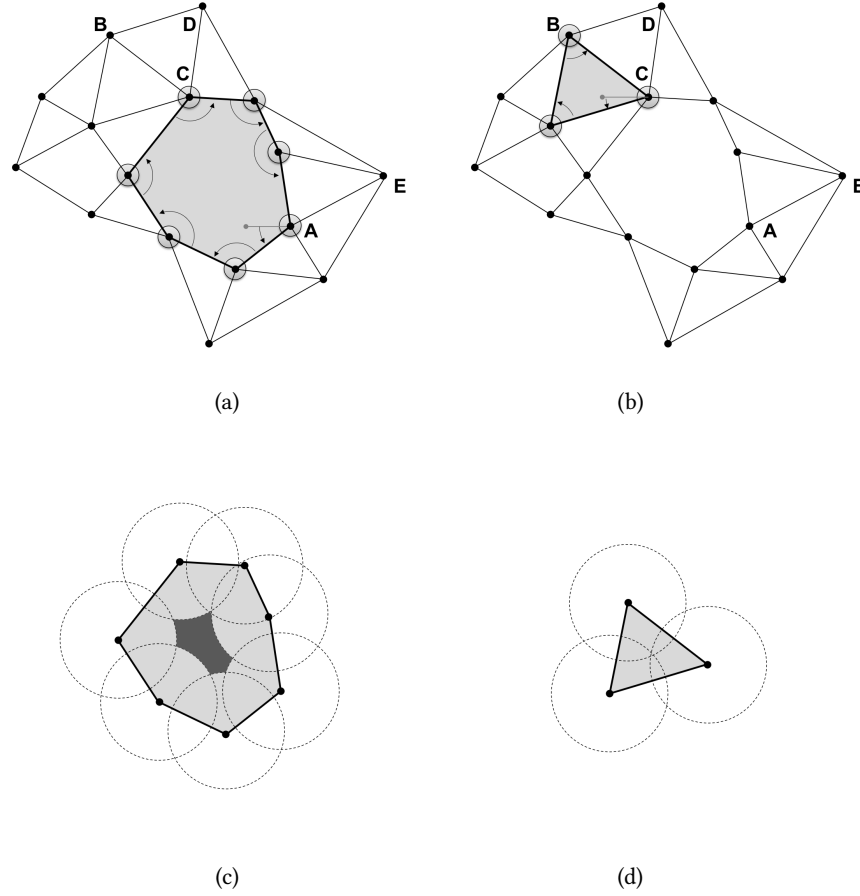


Figure 1: Gaps and Voids of a WSN.

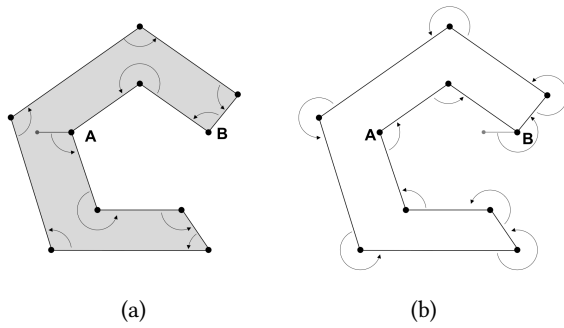


Figure 3: Interior and exterior polygons - Situation 2.

to some of its nodes, as shown by Figure 5 (a). This polygon has 17 nodes and, based on Equation (1), the sum of its interior angles should be equal to $(17 - 2) \times 180^\circ = 2700^\circ$. However, if we calculate the real sum of its angles, we will find 3060° which is equal to $(19 - 2) \times 180^\circ$. It is as if the polygon had 19 nodes instead of 17, which is correct since the nodes A and B are visited twice. Figure 5

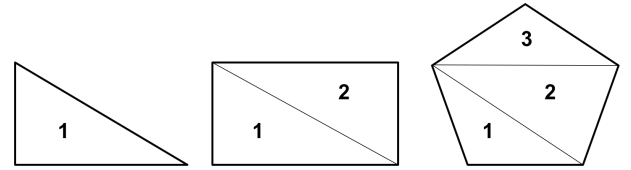


Figure 4: Sum of the interior angles of a polygon.

(b) shows the polygon which has the same angles as the polygon of Figure 5 (a) but which has 19 instead. This polygon clarifies how the nodes are visited.

We can conclude from this situation that Equation (1) can be generalized by changing the definition of n which represents in reality the number of times that all the nodes of a polygon have been visited.

Now, if we assume that there is no interior connection between nodes, as shown by Figure 1 (f), and if we start D-LPCN from node E we will obtain the same polygon as in Figure 1 (e), but it is now an interior polygon and the formula of Equation (1) will be verified.

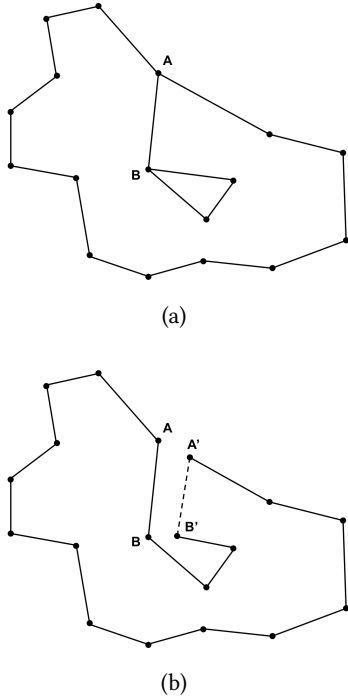


Figure 5: Visiting many times the nodes of an interior polygon.

As we can see, the dark-gray zone is not covered by the detection zones of the nodes of the polygon.

Another problematic in detecting voids in Wireless Sensor Networks is the calculation of the detection area's surface. In other words, once the boundary nodes determined, how can we know that there is an uncovered area? A simple answer is by calculating the surface of the obtained polygon hull. We can fix a certain threshold and if the surface is greater than this threshold, we will consider that a void is detected.

4 THE ALGORITHM

Algorithm 1 is based on D-LPCN presented in [14] where we have added the code allowing to determine the nature of the found polygon (exterior or interior). We have added line 15 to calculate the first angle. Since the first angle is based on a fictitious node situated to the left of the starting node, only a part of the first angle is calculated. The variable f_angle allows to calculate the other part which is the angle situated between the fictitious node and the neighbor node forming the maximum polar angle. This node is also the one forming the minimum anti-polar angle between the fictitious node and the neighbors of the starting node. The variable ϕ_max is calculated in lines 30 to 32. The variable t_angle of line 16 allows to calculate the sum of the angles of the polygon. Each node will receive this value from its previous neighbor (p_angle), and then add it to the value of its own angle. The same procedure is used for the number of visited nodes nbr_bn , which is to be compared with the value given by Equation (1). Once these values are updated,

they will be sent to the next neighbor n_id in line 18. Lines 36 to 42 concern only the starting node where the received sum of the angles of the polygon can be correct or not. If it is correct then the obtained polygon is interior, otherwise it is exterior.

5 SIMULATION AND RESULTS

5.1 Simulation tool: CupCarbon

We have used the simulator CupCarbon [16][17][18] to implement the proposed algorithm. The advantage of using this simulator is that it offers an ergonomic interface allowing to implement algorithms in an easy way and to visualize the simulation results during execution. Figure 6 shows an example of the graphical user interface of this simulator. It mainly represents an Openstreet-map where sensor nodes are deployed in a city. The simulation results can be visualized in terms of sending/receiving messages and marked and unmarked nodes. It is possible to display messages on each node. In our case, the nodes of a gap or a void will be marked, and we will use the option of creating buildings in order to add obstacles to the network.

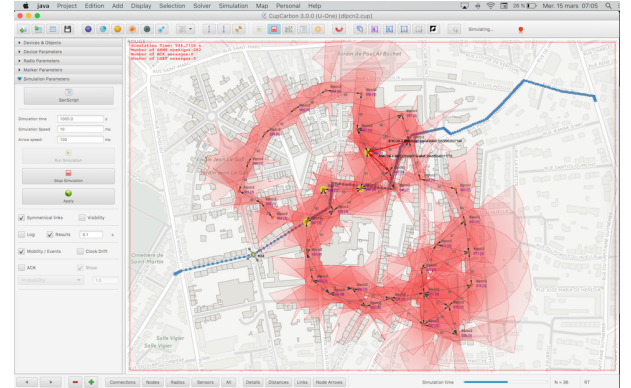


Figure 6: CupCarbon simulator.

5.2 Simulation results

In this section we will show the results obtained by executing Algorithm 1 using the simulator CupCarbon. Note that in this paper, we assume that the starting node is determined manually.

First, we fix the node with identifier 1 as a starting node and we run the proposed algorithm. Figure 7 shows the obtained result. As we can see, the nodes of the gap are marked, and the message displayed by the starting node is "INTERIOR", which means that the obtained polygon is interior. Figure 8 shows the detection area covered by the nodes of the gap. As we can see, the gap is completely covered by the node and there is no formation of a void.

Second, we move the node with identifier 1 to the boundary of the network and we run the proposed algorithm. Figure 9 shows the obtained result. As we can see, the nodes of the gap are marked, but the starting node is displaying the message "EXTERIOR", which means that the obtained boundary is exterior and it cannot be considered as a gap.

Now, let us present two examples, where a gap is caused by an obstacle. Figure 10 shows some nodes that are isolated because of

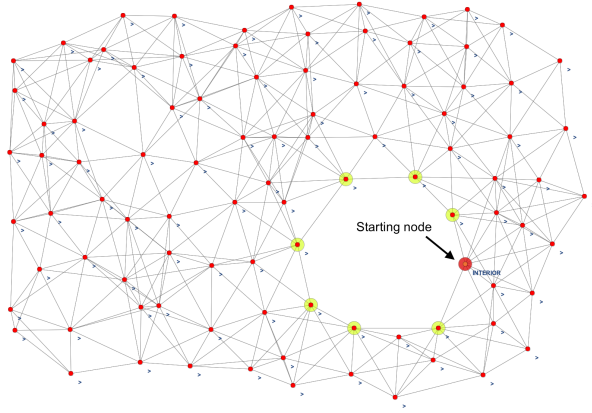


Figure 7: Simulation results (a gap).

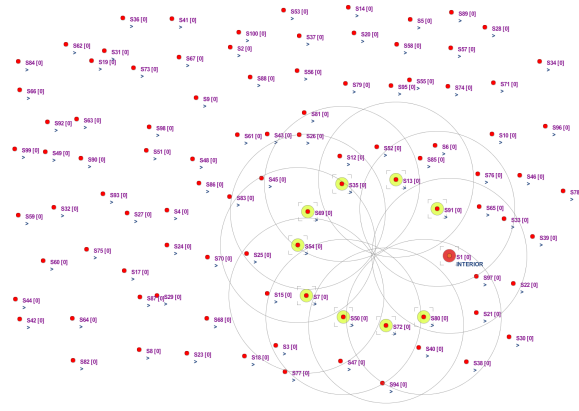


Figure 8: Simulation results (a void).

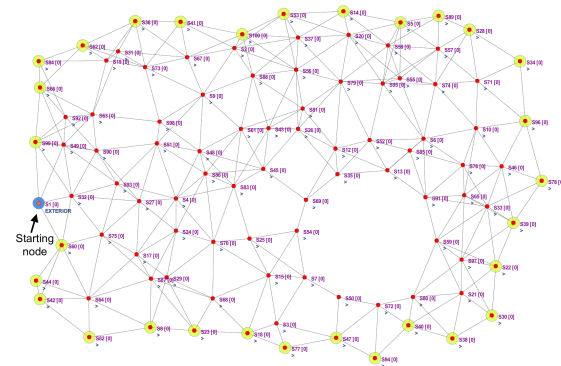


Figure 9: Simulation results (boundary nodes).

an obstacle. Here, the starting node is the same as in Figure 7 and the execution of Algorithm 1 will lead to the same result. Figure 11 shows another example where the obstacle forms a special situation where some nodes are connected to the node forming the interior polygon. Even in this situation we can see that the obtained polygon is considered as interior.

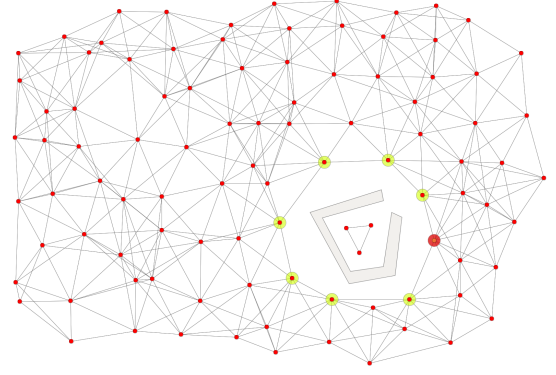


Figure 10: Simulation results (obstacle 1).

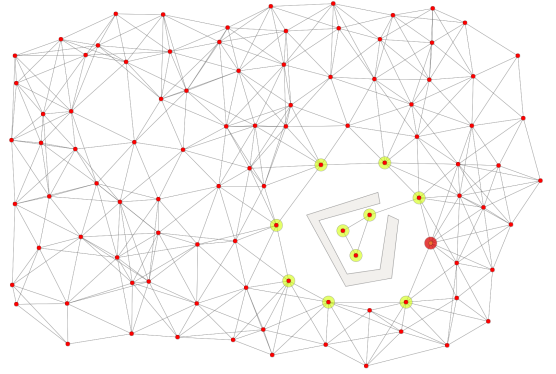


Figure 11: Simulation results (obstacle 2).

Finally, Figure 12 shows a detected gap or a void which contains a set of faulty or hacked nodes.

We conclude from these results that the proposed algorithm allows to determine, in a distributed way, the boundary nodes of a gap by detecting interior polygons. The starting node is fixed manually in the presented simulation results. However, it is clear that this node must be fixed automatically and the gaps and voids must be detected automatically, too. As a solution to this issue, we propose to run the proposed algorithm by starting from each node sequentially. This can be done by the *Wait-Before-Starting* (WBS) algorithm presented in [15]. Then if the obtained polygon has an area greater than a given threshold, the obtained boundary nodes can be considered as a gap. In addition, if the detection zones lead to a non-covered area, then we can consider that the obtained boundary nodes represent a void.

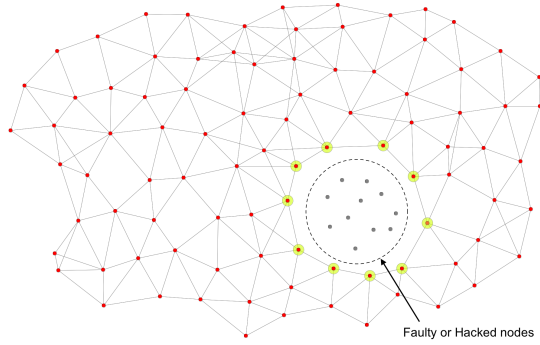


Figure 12: Simulation results (faulty or hacked nodes).

6 CONCLUSION

In this paper, we have proposed a new algorithm to detect gaps and voids in Wireless Sensor and IoT Networks. The algorithm is based on geometrical calculation, where the D-LPCN Algorithm [14] is executed first to determine the boundary nodes of the network in the form of a polygon. Then we use a geometrical calculation based on the sum of the obtained angles to determine whether the polygon is interior or exterior. In case that the polygon is interior and that its area is greater than a given threshold, this area is considered as a gap. In addition, if the detection areas of all the nodes forming this polygon do not cover the entire gap, then it will be considered as a void. The simulation results show that the algorithm can detect gaps and voids by taking into account the presence of obstacles. The main drawback of the proposed method is the determination of the starting node. We have to start the algorithm from each node and based on the proposed method, the obtained boundary can be considered as a void, a gap or none of both. We are working on methods allowing to find the starting node in an efficient way.

REFERENCES

- [1] Mohammad Hammoudeh. 2015. Applying Wireless Sensor Networks to Solve Real-world Problems. In Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC'15), Djallel Eddine Boubiche, Faouzi Hidoussi, and Homero Toral Cruz (Eds.). ACM, New York, NY, USA. DOI: <http://dx.doi.org/10.1145/2816839.2816935>
- [2] Papadimitratos, P., and Haas, Z. J. Secure routing for mobile ad hoc networks. In SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNSD 2002) (2002), vol. 31, San Antonio, TX.
- [3] Wang, W., and Bhargava, B. Visualization of wormholes in sensor networks. In Proceedings of the 3rd ACM workshop on Wireless security (2004), ACM, pp. 51-60.
- [4] Andrew Carlin, Mohammad Hammoudeh and Omar Aldabbas, Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges, International Journal of Advanced Computer Science and Applications, 2015, 6(6), 1–15.
- [5] Aissani, M., Mellouk, A., Badache, N., and Djebbar, M. A preventive rerouting scheme for avoiding voids in wireless sensor networks. In Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE (2009), IEEE, pp. 1-5.
- [6] Balister, P., Zheng, Z., Kumar, S., and Sinha, P. Trap coverage: Allowing coverage holes of bounded diameter in wireless sensor networks. In INFOCOM 2009, IEEE (2009), IEEE, pp. 136-144.
- [7] Bouk, S. H., Ahmed, S. H., Park, K.-J., and Eun, Y. Edove: Energy and depth variance-based opportunistic void avoidance scheme for underwater acoustic sensor networks. Sensors 17, 10 (2017), 2212.
- [8] Dong, D., Li, M., Liu, Y., Li, X.-Y., and Liao, X. Topological detection on wormholes in wireless ad hoc and sensor networks. IEEE/ACM Transactions on Networking (TON) 19, 6 (2011), 1787-1796.
- [9] Jia, W., Wang, T., Wang, G., and Guo, M. Hole avoiding in ad- vance routing in wireless sensor networks. In Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE (2007), IEEE, pp. 3519-3523.
- [10] Lu, C., Blum, B. M., Abdelzaher, T. F., Stankovic, J. A., and He, T. Rap: A real-time communication architecture for large-scale wireless sensor networks. In Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE (2002), IEEE, pp. 55-66.
- [11] Zhang, D., and Dong, E. An efficient bypassing void routing protocol based on virtual coordinate for WSNs. IEEE Communications Letters 19, 4 (2015), 653-656.
- [12] A. Bounceur, R. Euler, A. Benzerbadji, F. Lalem, M. Saoudi, T. Kechadi, M. Sevaux, Finding a polygon hull in wireless sensor networks, in: European Conference on

Algorithm 1: Type of polygon determined by angles

```

1 boundary = false; phi_min = 10; phi_max = -10;
2 c_id = getId(); c_coord = getCoord();
3 boundary_set = ∅; i=0; n = getNumberOfNeighbors(); selected = false;
4 first_node = any node;
5 if (first_node) then
6   boundary = true;
7   p_coord = (c_coord.x-1, c_coord.y);
8   send(c_id+"|"+"AC", *);
9 end
10 repeat
11   id = read();
12   type = read();
13   if (i==n) then
14     boundary_set = boundary_set ∪ {c_id};
15     f_angle = 6.28319-phi_max;
16     t_angle = phi_min + p_angle;
17     send(c_id+"|"+"SN"+"|"+"c_coord"+"|"+"boundary_set, n_id);
18     send(t_angle+"|"+"nbr_bn, n_id);
19   end
20   if (type=="AC") then
21     send(c_id+"|"+"CS"+"|"+"c_coord, id);
22   end
23   if (type=="CS") then
24     n_coord = read(); i=i+1;
25     phi = angleWI(p_coord, c_coord, n_coord, boundary_set);
26     if (phi < phi_min) then
27       phi_min = phi;
28       n_id = id;
29     end
30     if (phi > phi_max) then
31       phi_max = phi;
32     end
33   end
34   if (type=="SN") then
35     if (selected and first_node) then
36       t_angle = f_angle+p_angle;
37       ref_angle=(nbr_bn-2)×3.141595;
38       if (t_angle == ref_angle) then
39         print "INTERIOR";
40       else
41         print "EXTERIOR";
42       end
43       stop();
44     else
45       selected = true;
46       boundary=true; phi_min=10; phi_max=-10; i=0;
47       p_coord = read();
48       boundary_set = read();
49       p_angle=read(); nbr_bn=read();
50       nbr_bn = nbr_bn + 1;
51       send(c_id+"|"+"AC", *);
52     end
53   end
54 until false;

```


Operational Research, University of Strathclyde, Glasgow, UK, Invited talk, EURO 2015, July 2015

- [13] F. Lalem, A. Bounceur, M. Bezoui, M. Saoudi, R. Euler, and M. Sevaux. LPCN: Least Polar-angle Connected Node Algorithm to Find a Polygon Hull in a Connected Euclidean Graph, *Journal of Network and Computer Applications (JNCA)*, Elsevier, 93 (2017), pp. 38-50. DOI: 10.1016/j.jnca.2017.05.005.
- [14] M. Saoudi, F. Lalem, A. Bounceur, R. Euler, A. Laouid, M. Bezoui, and M. Sevaux, D-LPCN: A Distributed Least Polar-angle Connected Node Algorithm for Finding the Boundary of a Wireless Sensor Network, *Ad Hoc Networks*, Elsevier, Volume 56, 1 March 2017, Pages 56-71.
- [15] A. Bounceur, M. Bezoui, R. Euler, F. Lalem, A Wait-Before-Starting Algorithm for Fast, Fault-Tolerant and Low Energy Leader Election in WSNs Dedicated to Smart-Cities and IoT. In the IEEE Sensors conference, Oct 2017, Glasgow, United Kingdom.
- [16] K. Mehdi, M. Lounis, A. Bounceur. CupCarbon: A Multi-Agent and Discrete Event Wireless Sensor Network Design and Simulation Tool, In IEEE 7th International Conference on Simulation Tools and Techniques (SIMUTools'14), Lisbon, Portugal, 2014.
- [17] A. Bounceur, CupCarbon: A New Platform for Designing and Simulating Smart-City and IoT Wireless Sensor Networks (SCI-WSN), In the International Conference on Internet of things and Cloud Computing (ICC 2016), The Moller Center Churchill College, University of Cambridge, United Kingdom, 22-23 March 2016.
- [18] CupCarbon simulator, <http://www.cupcarbon.com>