# Detecting Gaps and Voids in WSNs and IoT Networks: the Minimum x-Coordinate based Method

Ahcène Bounceur, Madani Bezoui, Loïc Lagadec, Reinhardt Euler, Abdelkader Laouid, Mahamadou Traore, Mounir Lallali

# Detecting Gaps and Voids in WSNs and IoT Networks: the Minimum x-Coordinate based Method

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

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

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

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

Mahamadou Traore
Université
Gaston Berger
Saint-Louis, Senegal
Mahamadou.Traore@ugb.edu.sn

Mounir Lallali
Lab-STICC CNRS UMR 6285
Université de Bretagne Occidentale
Brest, France
mounir.lallali@univ-brest.fr

## ABSTRACT

When we deal with the deployment structure of Wireless Sensor Networks (WSNs) used in applications where the zone-of-interest is not accessible by humans, like forest fire detection, military applications, etc., random deployment is often the main or even the only practical solution that can be chosen. One of the main issues in this deployment is that it can lead to a formation of gaps or voids, which represent non-covered zones in the network. This can be very problematic, since it is not possible to detect some serious and dangerous problems, like a starting fire, the presence of non-desired persons or cyber-security attacks, etc. Therefore, detecting non-covered zones is of high importance. In this paper, we present a new method that allows to detect gaps and voids in WSNs and IoT networks after executing the D-LPCN algorithm and using some characteristics related to the value of the angle formed by the node of the gap having the minimum x-coordinate.[1]

## CCS CONCEPTS

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

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


*ICFNDS'18, June 26–27, 2018, Amman, Jordan*

ACM ISBN 978-1-4503-6428-7/18/06…$15.00
https://doi.org/10.1145/3231053.3231090

## KEYWORDS

Wireless Sensor Network, IoT, gap, void, Distributed algorithms, polygons, angles, D-LPCN

**ACM Reference Format:**
Ahcène Bounceur, Madani Bezoui, Loïc Lagadec, Reinhardt Euler, Abdelkader Laouid, Mahamadou Traore, and Mounir Lallali. 2018. Detecting Gaps and Voids in WSNs and IoT Networks: the Minimum x-Coordinate 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/10.1145/3231053.3231090

## 1 INTRODUCTION AND RELATED WORK

This paper comes within the context of the deployment of Wireless Sensor Networks or nodes of an IoT network [1], especially in the case of a random implementation. One of the main issues in this kind of deployment is the risk to obtain non-covered zones. In this case, and depending on the considered application, these zones can be source of serious and dangerous problems, like cyber-security attacks [2][3] [4] or the presence of non detected events, lack of sensed measurements which can lead to a lack of accuracy in the analyzed data, routing problems, etc. This will reduce the Quality of Service (QoS) of the network, in terms of communication, target detection, and cyber-security.

Most of the existing methods deal mainly with the voids in terms of communication and use greedy algorithms. Some approaches deal with target detection and use mainly statistical methods. The ratio of the surface of the zone-of-interest to the zone covered by the sensor nodes is the main metric to measure the QoS of the coverage. The total coverage problem is studied in [5][6][7][8].

In this paper and as a complement, we propose a method 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 improve the QoS as well as to take appropriate decisions about the existence of that zone. Since the proposed method allows to detect non-covered zones caused by both radio communication and target detection, we will use the two words *Gap* and *Void* to differentiate between them. These notions are defined in Section 2.

To deal with the problematic of detecting non-covered zones in Wireless Sensor and IoT Networks, many algorithms have been proposed in the literature. In [15], the proposed method allows to anticipate voids and to route messages by bypassing them. In [9], a method based on the transmitter locations is presented. In [11], a system avoiding opportunistic voids is proposed, which allows to balance energy, especially in Underwater Acoustic Sensor Networks. In [12], a topology-based approach is introduced which only requires the topology of the network 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, which is introduced by [10], where the size of a coverage hole is defined as an indicator of the quality of coverage. The authors of [16] have proposed a heuristic which guarantees an optimal network coverage as well as a good connectivity. The proposed algorithm requires that the selected sensors should be able to communicate with each other. First, it starts by discovering a subset of sensors for partial coverage that guarantee a given coverage, while ensuring a reduced energy consumption and in which the communication graph induced by the chosen sensors is connected. A greedy anti-void routing (GAR) protocol is proposed in [17] to ensure packet delivery and enhance the routing efficiency by resolving the void problem based on the Unit Disc Graph (UDG) principle. The GAR protocol combines both the conventional Greedy forwarding (GF) algorithm and the Rolling-ball UDG boundary Traversal (RUT) scheme. The author of [18] aims to resolve three main issues in WSN topology control: routing void, isolated node and sleeping control. A Mobile Agent-based Topology Control algorithm (MATC) is proposed as a solution for the three issues, where any given sensor selects the sensor which is closer to the sink as its next-hop. The routing void problem is resolved by adding a mobile agent. A geographic routing algorithm is proposed in [19], called Distance Upgrading Algorithm (DUA), which avoids the wholes without the use of the right-hand rule. The algorithm eliminates the wholes that cause non-optimal routing paths in geographic routing. The basic idea is that, if a packet passes through a void from a sensor to the base station, then the distance between the sensor and the base station is larger than the Euclidean distance. The paper [20] proposes a greedy routing algorithm to handle voids in a localized way by using the acknowledgment of the base station depth table. Its principle is to focus on the distance in hops from the set of base stations in the network.

Authors of [9] classified the existing techniques into two types:

- the right-hand rule which is to share the borders in several communication sessions [14],
- the backpressure rule, in which data packets tend to be pushed back to the upstream nodes for alternative routes [13].

In this paper, we propose a new method based on geometric calculation to determine whether a polygon is interior or exterior. If the polygon is interior then it is possible that it represents a void or a gap. 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 whether this interior polygon is a void or a gap.

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

Before presenting the method proposed in this paper, let us first explain the notions of *void* and *gap* and the difference between them. A gap is based on the area delimited 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.

A void is a zone which is not covered in the gap by the detection zones of its sensor nodes assuming that the detection radius is smaller than the radio communication range. In case of gaps, we need to determine the polygon formed by a given set of sensor nodes, which must be an interior polygon. In case of a void, we need to determine, in addition, whether the zone of this polygon is not totally covered by the detection areas of the sensor nodes 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 next section, we will present the main contribution of this paper, which represents a new method allowing to characterize the type of a polygon: interior or exterior.

## 3 MINIMUM X-COORDINATE-BASED METHOD

In this section, we denote by *Polygonal Global Minimum (PGM)* the vertex of a polygon having the smallest $x$-coordinate. This parameter will be useful for a characterization of internal and external polygons. Also, we designate by *local minimum* a vertex which has not a neighbor with an $x$-coordinate smaller than its own. Figure 2 shows such a local minimum B, where in (a) and (b), the y-coordinate of A is smaller than that of C, and in (c), the y-coordinate of A is greater than that of C. It is obvious that this property also applies to the global minimum.

We use the parameter PGM to characterize a polygon with respect to local and global minima, which is related to the order of their visit with respect to their neighbors. Figures 3 and 4 show examples of a local/global minimum visited differently. In Figure 3 (a), vertex B is visited after A and forming a polar angle greater or equal to 180°, although, in Figure 4 (a), vertex B is visited after C with a polar angle smaller or equal to 180°.

The cases of Figures 3 (b) and 4 (b), where the visited angle is equal to 180°, show that the order of visiting a local minimum

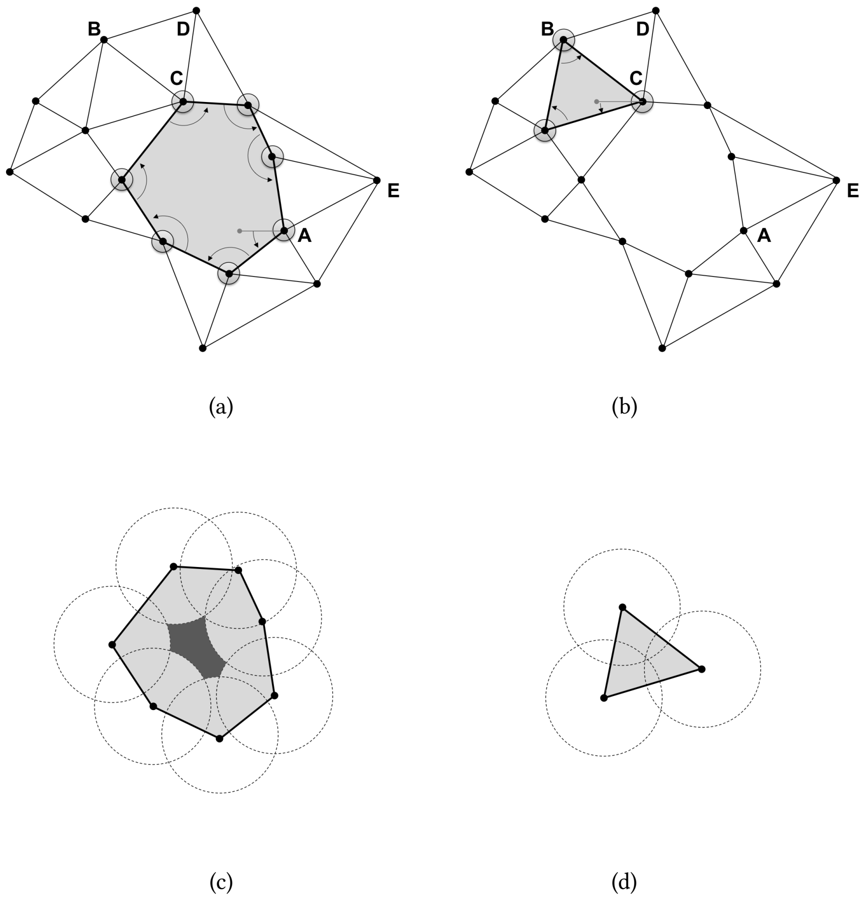


**Figure 1: Gaps and Voids of a WSN.**

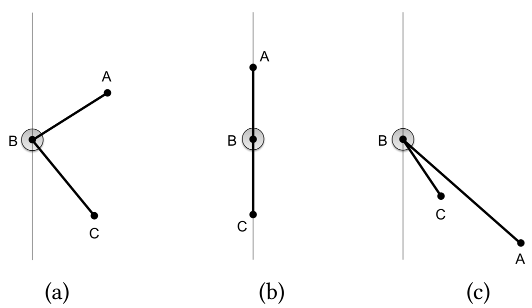


**Figure 2: Possibilities for local minima with respect to the x-coordinate.**

*can* depend on the y-coordinates of its neighbors. In other words, if the y-coordinate of A is smaller than that of C, then visiting B after A will lead to an angle $\widehat{ABC}$ greater or equal to 180°, and visiting B after C will lead to an angle $\widehat{CBA}$ smaller or equal to 180°. However, Figures 3 (c) and 4 (c), show the same situation with the same conditions for the angles as in (a) but with a y-coordinate of A greater than that of C. Thus, the order of visiting a local minimum vertex depends on the value of the angle formed with its neighbors in case where this value is greater or smaller than 180°, and only if this value is equal to 180°, the order depends on the values of the neighbors' y-coordinates.

We conclude from both situations that if the visited angle formed by a local minimum with its neighbors is greater than 180°, then this angle is visited from its exterior part, which means to the left of the local minimum and if this angle is smaller than 180°, then it is visited from its interior part, which means to the right of the local minimum. Finally, if this angle is equal to 180°, then we look at the y-coordinates of the neighbors of the local minimum. If the previous neighbor has a y-coordinate smaller than that of the subsequent neighbor, then the angle is visited from its exterior part. Otherwise, it is visited from its interior part.

The principal strength of this characteristic is that if the vertex B is the global minimum of the found polygon, the value of the angle formed with its predecessor and successor vertices can help to know if this polygon is interior or exterior. Note, that if the vertex B is visited more than once, then we only count the last visited angle. Figure 5 shows how to distinguish an interior polygon from

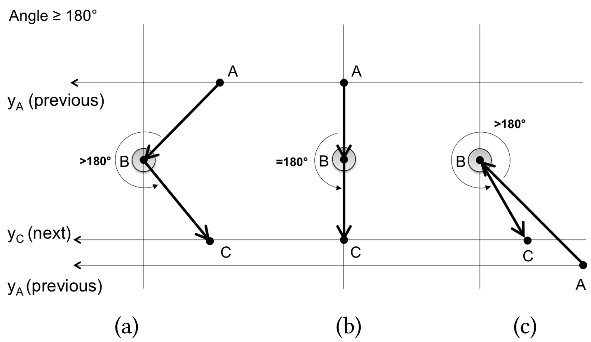


**Figure 3: Possibilities of visiting a local minimum - Situation 1.**

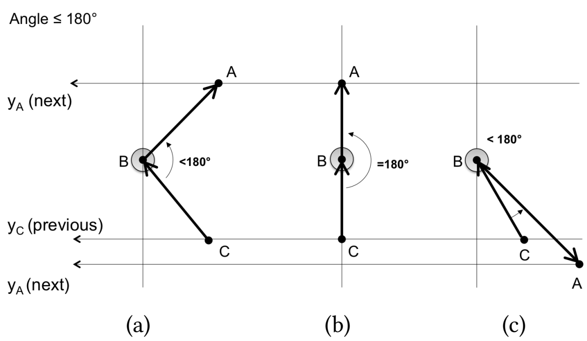


**Figure 4: Possibilities of visiting a local minimum - Situation 2.**

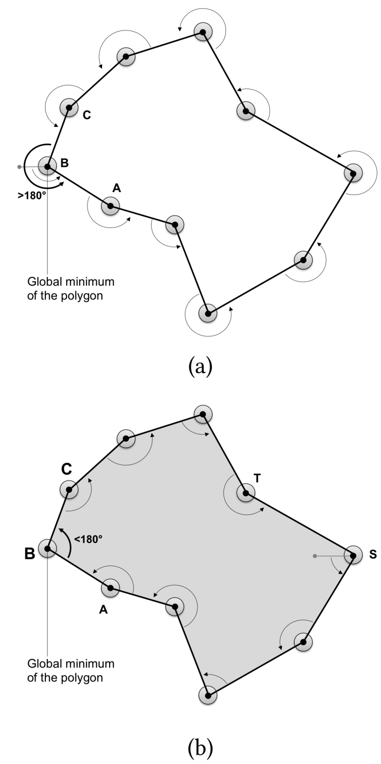


**Figure 5: An exterior and an interior polygon.**

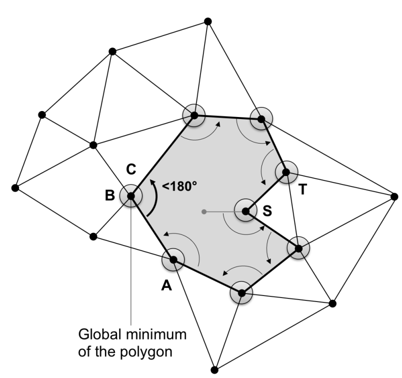


**Figure 6: An interior polygon within a connected Euclidean graph.**

an exterior polygon after visiting the vertices in polar order. In Figure 5 (a), the visit starts from the global minimum B. Then, once the stop condition is verified, i.e., the first angle or vertex C is visited a second time from vertex B, the last angle obtained for B is greater than 180°. Therefore, the polygon obtained is an external polygon. However, in the case of Figure 5 (b), the visit starts from vertex S, and once the stop condition is verified, i.e., vertex T is visited a second time from vertex S, the last angle obtained for B is less than 180°. Consequently, the polygon obtained is an interior polygon.

If we use the same principle in the case of graphs instead of polygons, we can get the same results, as shown in Figure 6 where for a visit from the starting point S, the last angle obtained for the global minimum B is less than 180°. As a result, the polygon obtained is an interior polygon.

## 4 THE ALGORITHM

Algorithm 1 is based on the algorithm D-LPCN presented in [23] to which we have added the code allowing to determine the nature of the found polygon (exterior or interior). We conclude by presenting the algorithm which is based on the global minimum of the found polygon. As in the previous case, it is the D-LPCN algorithm which is modified. In line 48, each node will test whether its x-coordinate is smaller than the one received from its neighbor. This will help to determine whether the variable is_min is true or false (lines 47 to 51), and also in lines 15 and 16 to determine which angle will be sent, the received one (line 46) or the currently calculated one (line 26). As soon as the values of min_x and min_angle have been determined, they will be sent to the next neighbor (line 19).

# 5 SIMULATION AND RESULTS

## 5.1 Simulation tool: CupCarbon

We have used the simulator CupCarbon [25][26][27] 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 7 shows an example of the graphical user interface of this simulator. It represents mainly 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 at 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 include obstacles in the network.

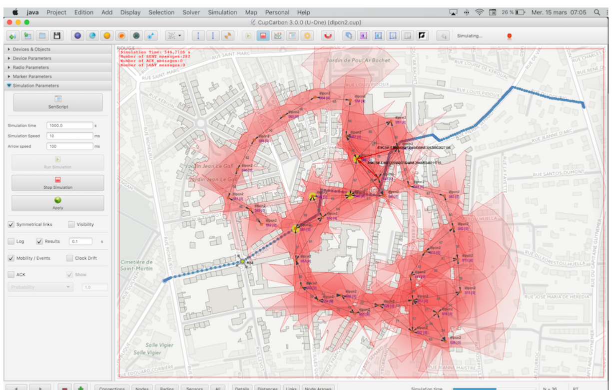

**Figure 7: CupCarbon simulator.**

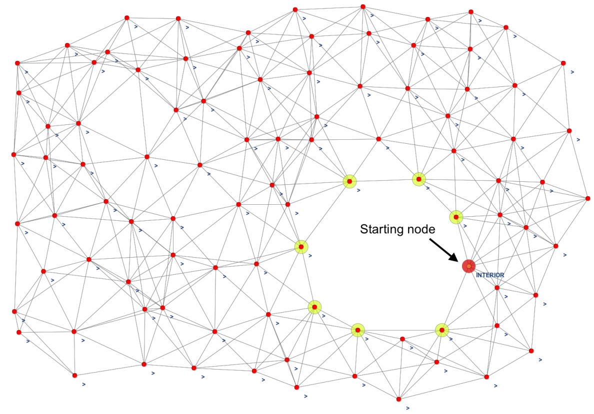


**Figure 8: Simulation results (a gap).**

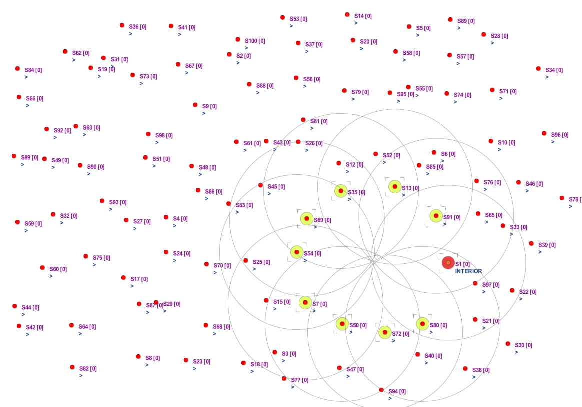


**Figure 9: Simulation results (a void).**

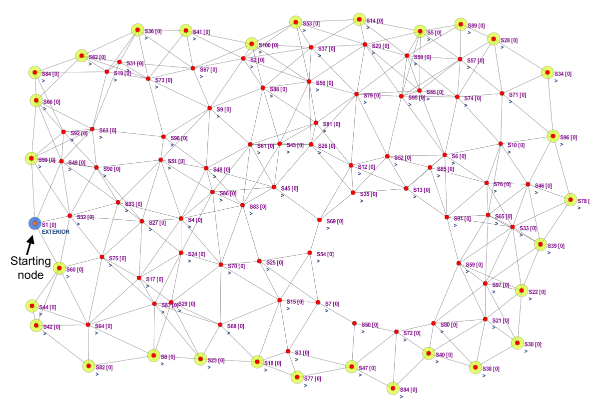


**Figure 10: Simulation results (boundary nodes).**

## 5.2 Simulation results

In this section, we will present 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 the starting node and we run the proposed algorithm. Figure 8 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 9 shows the detection area covered by the nodes of the gap. As we can see, the gap is completely covered by the nodes and thus 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 10 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.

We will also show two examples, where a gap is caused by an obstacle. Figure 11 shows some nodes that are isolated because of an obstacle. The starting node here is the same as in Figure 8 and the execution of Algorithm 1 will lead to the same result. Figure 12 shows another example where the obstacle forms a particular situation in which there are some nodes connected to the node forming the interior polygon. As we can see, even in this situation, the obtained polygon is considered as interior.

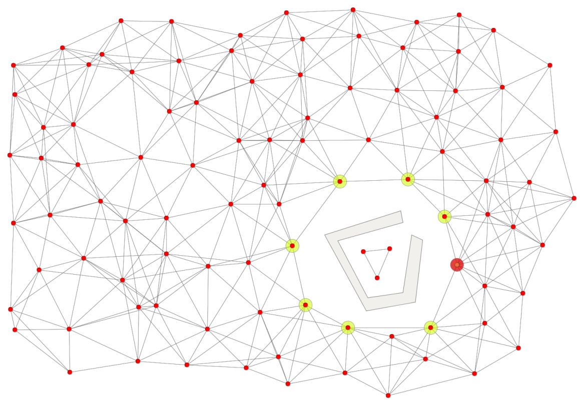

**Figure 11: Simulation results (obstacle 1).**

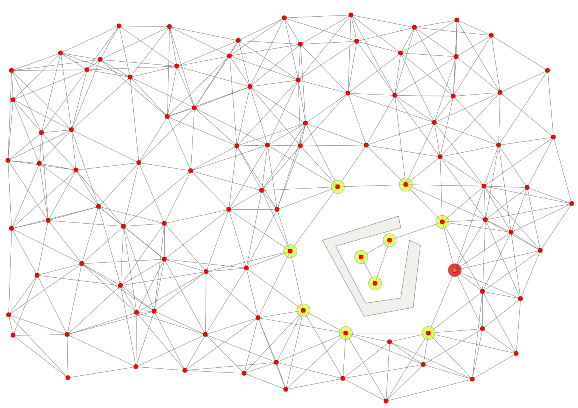

**Figure 12: Simulation results (obstacle 2).**

Finally, Figure 13 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 the 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 [24]. 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 circumscribe a void.

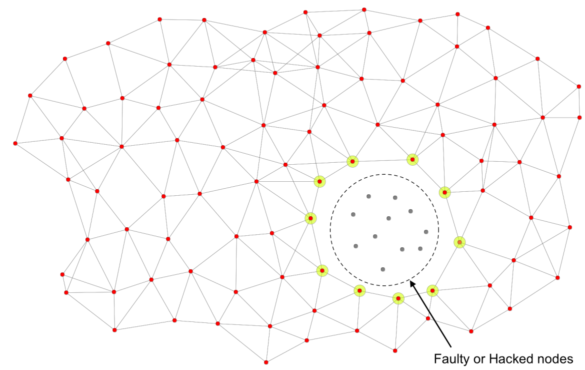


**Figure 13: Simulation results (faulty or hacked nodes).**

## 6 CONCLUSION

In this paper, we have proposed a new method to detect gaps and voids in Wireless Sensor and IoT Networks. The algorithm is based on geometric calculation, where the D-LPCN algorithm is executed first to determine the boundary nodes of the network in the form of a polygon. Then, we use a geometric calculation based on the value of the angle formed by the last visited node having the minimum x-coordinate to determine whether the obtained polygon is interior or exterior. In the case where the polygon is interior, if its area is greater than a given threshold, then this area is considered as a gap. Besides, if the detection areas of all the nodes forming this polygon do not cover the entire the 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 choice of the starting node, which requires initiating the D-LPCN algorithm on each node to determine if the obtained boundary is a void/gap or not. Future improvement of this algorithm includes a way of identifying the starting node.

## 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 (CNDS 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] Cardei, M., MacCallum, D., Cheng, M. X., Min, M., Jia, X., Li, D., and Du, D. Z. Wireless sensor networks with energy efficient organization. Journal of Interconnection Networks, 3(03n04), 2002, 213–229.

[6] Cardei, M., and Du, D. Z. Improving wireless sensor network lifetime through power aware organization. Wireless Networks, 11(3), 2005, 333–340.

[7] Carle, J., and Simplot-Ryl, D. Energy-efficient area monitoring for sensor networks. Computer, 37(2), 2004, 40-46.

[8] Zhou, Z., Das, S., and Gupta, H. (2004, October). Connected k-coverage problem in sensor networks. In Proceedings of the 13th IEEE International Conference on Computer Communications and Networks (ICCCN), 2004, pp. 373–378.

**Algorithm 1:** Type of polygon determined by global minimum

```
boundary = false; phi_min = 10; phi_max = -10;
c_id = getId(); c_coord = getCoord();
boundary_set = ∅; i=0; n = getNumberOfNeighbors(); selected = false;
first_node = any node;
if (first_node) then
    boundary = true;
    p_coord = (c_coord.x−1, c_coord.y);
    send(c_id+"|"+"AC", *);
end
repeat
    id = read();
    type = read();
    if (i==n) then
        boundary_set = boundary_set ∪ {c_id};
        if (is_min) then
            min_angle = phi_min;
        end
        send(c_id+"|"+"SN"+"|"+c_coord+"|"+boundary_set, n_id);
        send(min_x+"|"+min_angle, n_id);
    end
    if (type=="AC") then
        send(c_id+"|"+"CS"+"|"+c_coord, id);
    end
    if (type=="CS") then
        n_coord = read(); i=i+1;
        phi = angleWI(p_coord, c_coord, n_coord, boundary_set);
        if (phi<phi_min) then
            phi_min = phi;
            n_id = id;
        end
    end
    if (type=="SN") then
        if (selected and first=_node) then
            if (min_angle ≤ 3.141595) then
                print "INTERIOR";
            else
                print "EXTERIOR";
            end
            stop();
        else
            selected = true; boundary = true; phi_min = 10; i=0;
            p_coord = read();
            boundary_set = read();
            min_x = read();
            min_angle = read();
            is_min = false;
            if (c_coord.x < min_x) then
                is_min = true;
                min_x = c_coord.x;
            end
            send(c_id+"|"+"AC", *);
        end
    end
until false;
```


[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Yuzhen Liu and Weifa Liang, Approximate coverage in wireless sensor networks, The 30th Anniversary of the IEEE Conference on Local Computer Networks (LCN'05), Sydney, NSW, 2005, pp. 68–75.

[17] W. J. Liu and K. T. Feng, Greedy Routing with Anti-Void Traversal for Wireless Sensor Networks, in IEEE Transactions on Mobile Computing, vol. 8, no. 7, pp. 910–922, July 2009.

[18] Lu Hong, Mobile agent based topology control algorithms for wireless sensor networks, 2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Shanghai, 2013, pp. 195–199.

[19] Shigang Chen, Guangbin Fan, and Jun-Hong Cui. Avoid 'void' in geographic routing for data aggregation in sensor networks. International Journal of Ad Hoc and Ubiquitous Computing. Vol. 1, Issue. 4, July 2006, 169–178.

[20] Gustavo Weber Denardin, Carlos Henrique Barriquello, Alexandre Campos, Ricardo Nederson do Prado, A geographic routing hybrid approach for void resolution in wireless sensor networks, Journal of Systems and Software, Volume 84, Issue 10, 2011, Pages 1577–1590.

[21] A. Bounceur, R. Euler, A. Benzerbadj, F. Lalem, M. Saoudi, T. Kechadi, M. Sevaux, Finding a polygon hull in wireless sensor networks, in: European Conference on Operational Research, University of Strathclyde, Glasgow, UK, Invited talk, EURO 2015, July 2015.

[22] 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.

[23] 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.

[24] A. Bounceur, M. Bezoui, R. Euler, R. 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.

[25] 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.

[26] 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.

[27] CupCarbon simulator, http://www.cupcarbon.com