

## **Adaptive nodes algorithm to solve the orphan nodes problem in a ZigBee Tree WSN**

*Algoritmo de nodos adaptativos para solucionar el problema de los nodos huérfanos en una Red Inalámbrica de Sensores ZigBee en topología de árbol*

Christian David Escobar-Amado<sup>1</sup>    Jose Miguel Celis-Peñaranda<sup>1</sup>  
Sergio Basilio Sepúlveda-Mora<sup>1</sup>    Sergio Alexander Castro-Casadiago<sup>1</sup>  
Byron Medina-Delgado<sup>1</sup>    Dinael Guevara<sup>1</sup>

Recibido 16 de mayo de 2019, aceptado 9 de junio de 2020  
*Received: May 16, 2019    Accepted: June 9, 2020*

### **ABSTRACT**

We have developed an algorithm based on hybrid nodes composed of an XBee S2C module configured in API mode with escape and connected to a microcontroller ATMEGA 328P to solve the orphan node problem of a ZigBee tree topology. The sensor nodes have both features End Device and Router that may alternate depending on the situation, and the node can adapt itself to keep the network integrality as much as possible. On the other hand, the coordinator node comprises an XBee S2C module and the embedded system Raspberry Pi 3, which hosts the algorithm and is set up as an access point, a database server, and a web server. Using the proposed algorithm, we were able to recover up to 42.8% of the network coverage, but it comes with an energy cost due to the switch from End Device to Router. Additionally, a mobile application integrated with notification service and Google Maps was developed to monitor every node's behavior in the Wireless Sensor Network.

Keywords: WSN, ZigBee, Orphan Nodes, Hybrid nodes, mobile application.

### **RESUMEN**

*Para solucionar el problema de los nodos huérfanos en una topología tipo árbol en una red ZigBee, se ha desarrollado un algoritmo basado en nodos híbridos compuesto por un módulo XBee S2C configurado en modo API con escape conectado a un microcontrolador ATMEGA 328P. Los nodos sensores cuentan con las características de End Device y Router simultáneamente, las cuales pueden ser alternadas dependiendo de la situación, de esta forma, el nodo puede adaptarse por sí mismo para mantener la integridad de la red. Por otro lado, el nodo coordinador está conformado por un módulo XBee S2C y el sistema embebido Raspberry Pi 3 que soporta el algoritmo y está configurado como punto de acceso, servidor de base de datos y servidor web. Empleando el algoritmo propuesto, la cobertura de la red puede ser recuperada hasta en un 42,8%, pero viene con un costo de energía debido al cambio de End Device a Router. Adicionalmente, una aplicación móvil integrada con servicio de notificaciones y Google Maps fue desarrollada para monitorizar el comportamiento de cada nodo en la Red Inalámbrica de sensores.*

*Palabras clave: RIS, ZigBee, Nodos huérfanos, Nodos híbridos, aplicación móvil.*

---

<sup>1</sup> Universidad Francisco de Paula Santander. Departamento de Electricidad y Electrónica. Cúcuta, Colombia.  
E-mail: christiandavidea@ufps.edu.co; josemiguelcp@ufps.edu.co; sergio.sepulveda@ufps.edu.co; sergio.castroc@ufps.edu.co; byronmedina@ufps.edu.co; dinaelgi@ufps.edu.co  
\* Autor de correspondencia: christiandavidea@ufps.edu.co

## INTRODUCTION

Wireless Sensor Networks (WSN) typically consists of sensor nodes deployed in some area to achieve certain goals. Each node is equipped with a transceiver for communicating with neighboring nodes and a microcontroller for storing and processing data [1].

A common protocol in WSN is ZigBee, which supports three types of nodes, ZigBee Coordinator (ZC), ZigBee Router (ZR), and ZigBee End Device (ZED). The ZC is the central node responsible for establishing and managing the network and other functions such as security and general operation. The ZR has all the functionalities of the ZigBee protocol. It can join the network for sending, receiving, and routing information to the coordinator node. ZED has the same functions as the routers but has no routing feature; it is mainly responsible for uploading data collection to the coordinator node with lower energy consumption than a router.

ZigBee networks can be constructed into pair topology, star topology, and mesh topology to give the network its structure, as shown in Figure 1. The three network structures could be combined into the cluster tree topology to expand the network's coverage scope [2].

WSNs have been utilized over the years around the world for data collection in several fields. In China, Jiang *et al.* [3] used the ZigBee technology for a smart home system with an Android app to control

and monitor real-time household devices and the house's environment. In Turkey, Kbalci & Kbalci [4] worked on a metering and monitoring system for solar string inverters using ZigBee technology to transmit the measured current, voltage, and power to a host PC. In Taiwan, Chou *et al.* [5] implemented a WSN with biosensors, XBee modules, Arduino Mega, and a computer to monitor Glucose and Lactate using LabVIEW software. In Pakistan, Ali *et al.* [6] designed a WSN monitoring platform for the oil and gas pipeline using a custom sensor board named SimpliMote and a graphical user interface that can run on a Windows or Linux platform. In general, there are many applications where the WSNs can be used, and therefore, there is a necessity to develop new techniques to improve their efficiency and robustness.

To optimize this type of networks' behavior and efficiency, some algorithms and strategies have been proposed in the literature. For instance, in [3], the authors used a Genetic Algorithm-Particle Swarm Optimization Algorithm (GA-PSOA) to enhance the network's routing for searching for the optimal route with fast speed improving the response rate and energy consumption of the nodes. In [6], a network algorithm has been proposed for a Linear WSN (LWSN) that finds the neighbor nodes and relies on RSSI (Received Signal Strength Indicator) measurements to map each node looking for the best RSSI along with the hop count to reach a gateway node. Also, if there is some network failure, a message is broadcasted, and it will start

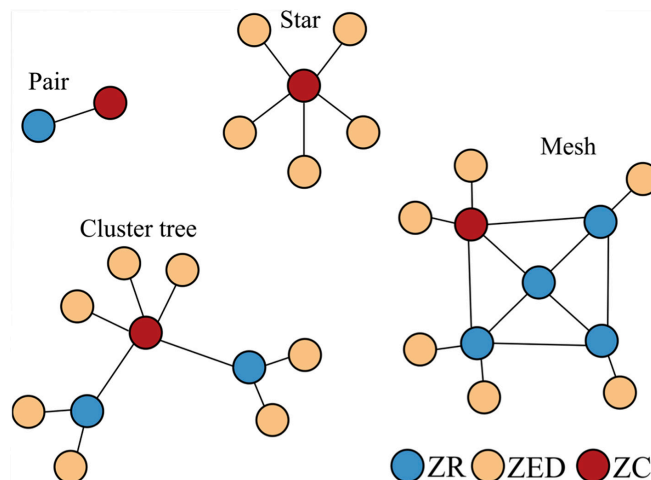


Figure 1. ZigBee pair, star, mesh and cluster tree topologies.

the networking process to repair the LWSN. In [7], to improve the system lifetime in a network, the authors developed a protocol architecture named Low-Energy Adaptive Clustering Hierarchy (LEACH) for avoiding “hot spots” in the network using the randomized rotation of the cluster head positions taking into account the energy level of the node; in this way, the node with higher energy has a higher probability of becoming a cluster head. By using LEACH, the energy in the network will be balanced, and its lifetime will increase. Other frameworks have been applied to the LEACH protocol to improve its functionality, such as the Virtual Field Force framework (LEACH-VF) to maximize the coverage of the network [8]; the K-LEACH-VF protocol to improve the energy balance of the network [9]; and, the LEACH-SP protocol to improve the covered area and the network lifetime [10].

Furthermore, in [11], an adaptive routing optimization technique is proposed to update the ZigBee hierarchical structure in real-time to route the information through the optimal path for energy-balancing. To reduce the congestion in a WSN based on ZigBee technology, Ullah & Youn [12] proposed a routing algorithm called Statistical Multipath Queue-wise Preemption. The purpose of the algorithm is to avoid the overload of some nodes and prioritize the packets based on their type by scheduling statistically multiple paths to improve the network’s performance.

In all these cases, there are chances of some faults to occur. The faults may occur due to compromised sensor nodes, exhausted batteries, or harsh environments [13]. When a failure occurs, the network will have holes, and its overall integrity might be compromised. If the failure of a ZR produces a hole in the network as depicted in Figure 2, then the children nodes of that ZR would lose communication becoming orphan nodes. If there are no paths to transmit information, then the proposed routing algorithms mentioned before might not repair the network, and the coverage of the network will decrease.

Some solutions have been proposed by managing addresses in the network and modifying the connection of joining nodes and their potential parents [14, 15, 16, 17] to solve the orphan nodes problem; also, in [18], Yang, Xu & Qiu proposed an algorithm

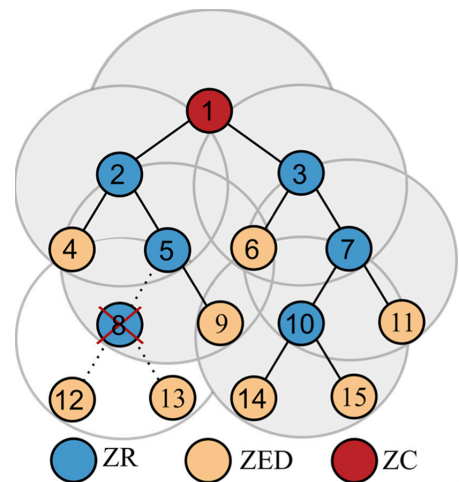


Figure 2. Coverage hole in a WSN.

to transform ZED into ZR by pressing a button. We present a solution based on hybrid nodes in this work, where the sensor nodes have both features, ZED and ZR. With this approach, if some fault occurs, the nodes are capable of switching from a ZED to a ZR and vice versa automatically. Moreover, with this algorithm, we can detect possible failures in the network and generate an alert notification as soon as possible using an Android mobile application, which was also designed to monitor the WSN.

## ADAPTIVE NETWORKING

### Nodes architecture and setup

Sensor nodes (ZED and ZR) are composed of an XBee S2C module configured in API mode with escape and connected to a microcontroller ATMEGA 328P which performs the changes in the node configuration according to the proposed algorithm and takes measurements from environmental sensors. Five environmental variables were considered (temperature, humidity, soil moisture, UV light, and visible light intensity) in the crop, and the data are wirelessly transmitted using ZigBee technology. The Li-Po battery provides electric energy to the nodes to guarantee an autonomy of at least 14 days (actual energy consumption varies according to sleep time mode and operating conditions). The microcontroller processes the sensors’ data and transmits the information of the variables to the Coordinator through the XBee module using the ZigBee protocol. Figure 3a shows the Sensor nodes, and 3b shows one of the nodes placed on the cocoa

crop tree. To switch between ZED and ZR, we used the AT command SM (Sleep Mode), where “1” is to set the node as ZED and “0” as ZR; depending on this parameter, the microcontroller executes a specific code.

Having said that, the timeout must be set up to an appropriate value; if it is shorter than the sleep time, the end device will be disassociated from the network before it wakes up; but, if it is too long and the end device needs to change parent, it will not be able to do it, because the child table will not change on time. To set up this time, we established the parameters SP (Sleep Periods) and SN (Sleep Number of Cycles) to 0 x 68 (104) and 0 x 28 (40), respectively. By replacing these values into equation 1, the timeout is set to 124.8 seconds. The sampling rate is 12 seconds; thus, the ZED is going to be in the child table at least for 10 cycles before it is disassociated, and it has to scan the network again, looking for a new parent.

$$Time = 3 * SP * SN * 10 * 10 - 3s \quad (1)$$

To test the proposed algorithm, we used the ZigBee tree topology (see Figure 4), which has more coverage and can be expanded to a bigger network. The WSN has been deployed on a cocoa plantation. It is an NLOS (No Line of Sight) scenario, and to properly operate, the maximum distance between nodes is 40 m.

**Embedded system and data base**

The collected information of all the sensor nodes is stored in a database hosted on the Raspberry Pi 3 (RPi3) embedded system, responsible for receiving the information from the sensor network using an XBee module connected through a serial port. The RPi3 and the XBee module work

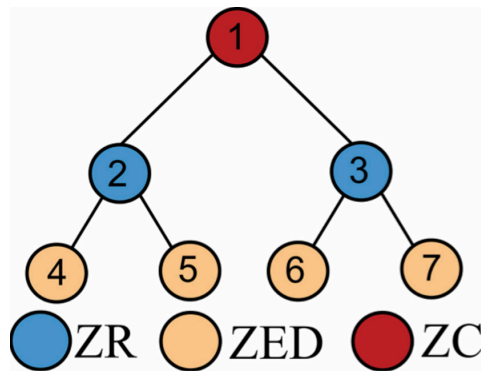


Figure 4. Implemented ZigBee Topology.

together as the network Coordinator node and use the Zigbee protocol to receive the information. The Raspberry Pi 3 model B has an interpreter for Python programming language. An algorithm in Python for processing and storing the data originating from the XBee modules was developed; the database where the data is stored requires the Structured Query Language, which is interpreted and organized by MySQL database manager. We developed a service to allow queries through Web ports using PHP language; this service allowed us to make GET queries to the database and allowed the responses to be interpreted and schematized. For that, relational tables are implemented in an Apache and MySQL server to structure the information, thus avoiding data redundancy and containing the main information about the nodes, as shown in Table 1.

The RPi3 was configured as an access point using its integrated module 802.11 n, a DHCP server, and user validation. In this way, the embedded system establishes its Wi-Fi network so that the mobile application can connect to it.



Figure 3. a) 6 Zigbee wireless sensor nodes. b) Zigbee wireless sensor node on the cocoa crop.

Table 1. Main information about the nodes.

Characteristic	Description
Type	0 - Coordinator
	1 - Router
	2 - End Device
Latitude	Decimal degrees of Latitude
Longitude	Decimal degrees of Longitude
Status	0 - Not Functional
	1 - Functional
	2 - Possible fail
	3 - Trying to restore
	4 - Failure to restore
	5 - End Device turned into Router
	6 - Router turned into End Device
Original Parent	Original Parent's ID node
Current Parent	Current Parent's ID node
Add_64	64 bits address
Add_16	16 bits address
Variables Average	Average value of measured variables
Date	Last time data was stored

**Behavior of the nodes**

The consumed current was measured using an oscilloscope connected to a 1 Ω resistance in series with the node to analyze the network sensor nodes' behavior.

The sampling rate for ZED is 10 seconds. A functional ZED associated with the network behaves as shown in Figure 5, where the average awake time is 440.94 ms, and the mean current is

10.88 mA. On the other hand, a ZED disassociated to the network behaves as shown in Figure 6, and the mean current is 21.88 mA. In contrast to ZED, ZR does not operate in sleep mode because a ZR must listen to its connected nodes all the time. Therefore, a ZR node consumes 40 mA on average, regardless if it is connected to the network or not, as shown in Figure 7.

**PROPOSED ALGORITHM**

The main purpose of the algorithm is to keep the integrality of the original network topology based on the status a node might have. Such statuses are enumerated according to the situation that occurs. We have established the following cases.

- (0) Not functional. It occurs when a node has lost communication with the network, and there are no near end devices with a status different from 0.
- (1) Functional. When a node is working with its original settings.
- (2) Possible Fail. It occurs when a node has lost communication with the network, and there is at least one near-end device with a "Functional" status (1).
- (3) Trying to restore. It occurs when the node is with a status of "Possible Fail" (2), and there is at least one near node whose status is "ZED turned into ZR" (4).
- (4) ZED turned into ZR. When an original ZED is turned into ZR to re-associate a near ZED with no communication.

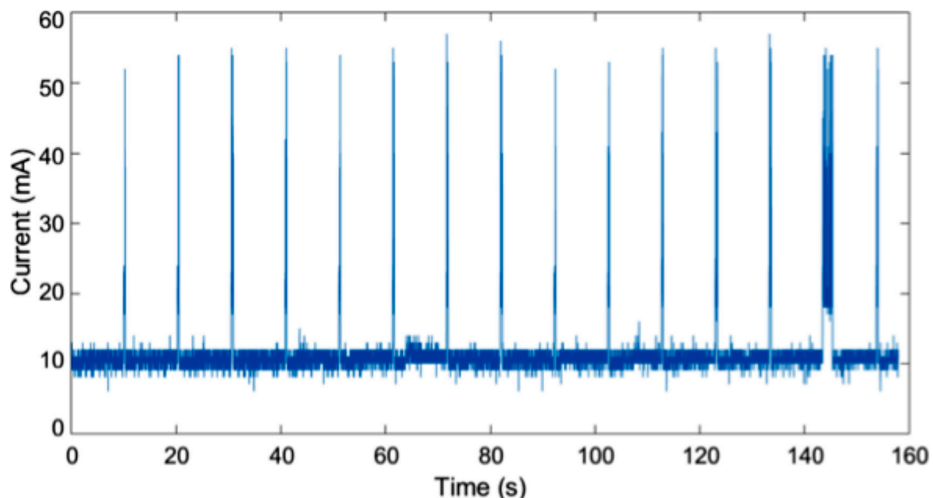


Figure 5. Associated ZED current behavior.

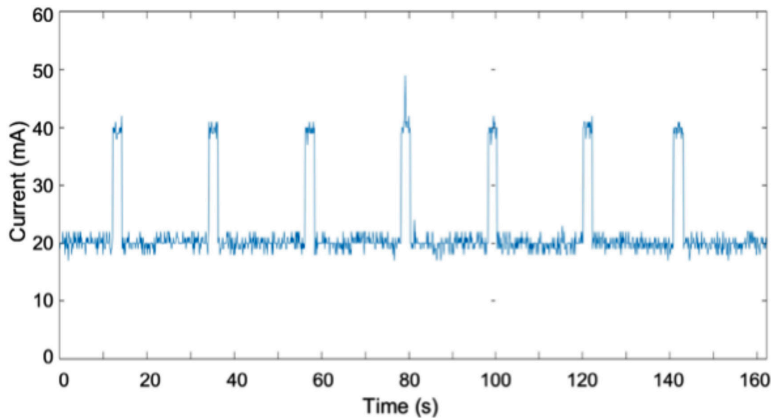


Figure 6. Disassociated ZED current behavior.

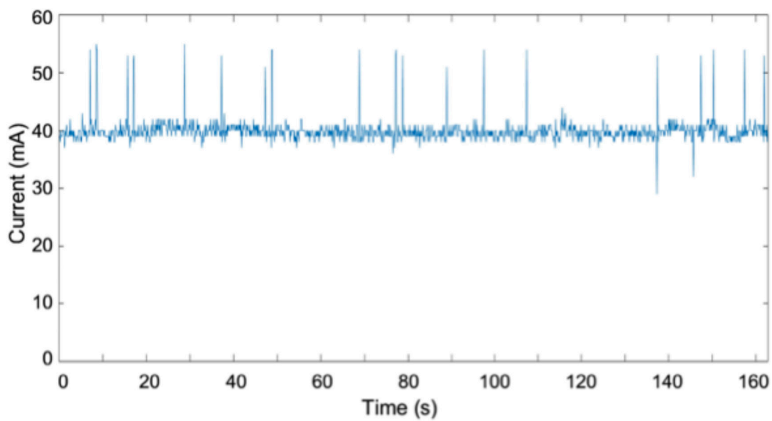


Figure 7. ZR current behavior.

- (5) ZR turned into ZED. When an original ZR is turned into ZED because it has no children.

As for energy consumption, considering the analysis made in the previous section, a ZR consumes 40 mA in every status, a ZED consumes 10.88 mA if the status is either 1 or 5, otherwise, it consumes 21.88 mA.

Depending on the problem in the network, there are different strategies to deal with it. We classified these problems as four main cases, and then, we analyzed the coverage and the energy consumption of each node based on the experimental data described in the previous sections.

**Case I. ZR loses communication with the WSN. Restoration of ZED**

In Figure 8, if N2 loses communication, then N4 and N5 will lose their parent, and the network

would have lost three nodes. However, with the proposed algorithm, the network will attempt to recover the lost nodes as follows (see the stages in Figure 8). 1) The network is working with all of its nodes. 2) N2 loses communication, so N2, N4, and N5 change their status to 2; after that, the embedded system selects the closest node to the coordinator with no communication, in this case, N2. The algorithm then looks for ZEDs close to that node with status different to 0 and different to its original children nodes in case of being a ZR; if there are no close end devices, then the node changes its status to 0 as occurred with N2 in Figure 8. After that, if the node cannot be restored, then the algorithm tries to restore the next closest one. 3) There are no functional end devices close to N4 but, N5, whose status is 2, could rejoin the network. N6 is a functional ZED close to N5 so, it is turned into a ZR and tries to become the parent

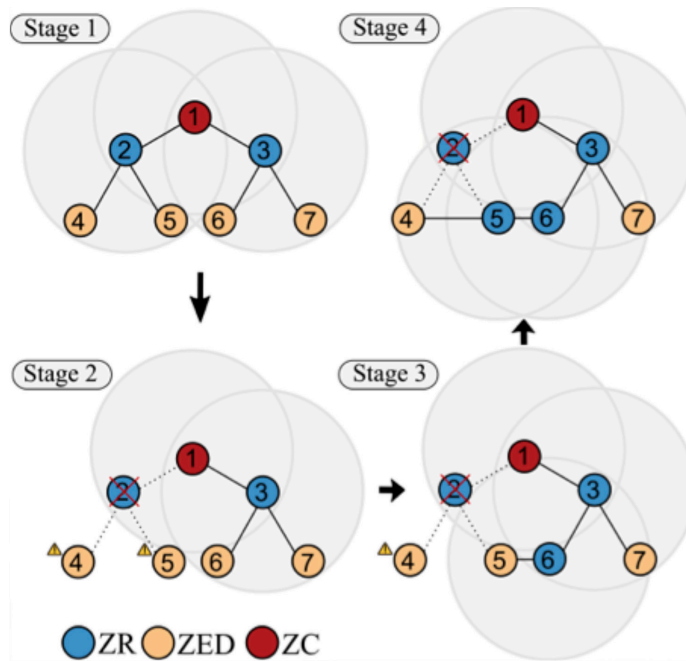


Figure 8. Case I.

of N5 to restore it to the network. 4) The just recovered N5 turns into a ZR and tries to restore N4 to the network.

Table 2 shows the parameters per node in every stage; in the end, the network consumes more current than the original topology, but after the fault in N2, it is working with two more nodes. Thereby, the WSN will work with five nodes instead of three, restoring 33.33% of its integrity.

**Case II. ZED turned into ZR with no children nodes**

In step 3, if for some reason N5 does not connect to N6 as shown in step 2.1 in Figure 9, then: 2.2) N6, whose status is 4, will be turned back into ZED and change its status to 1 because it has no children nodes. Also, N5 will be considered as a not functional node and change its status to 0. 2.3) Thus, there are no end devices close to N4, so it will also be considered as a not functional node.

Table 2. Status per stage in Case I.

Stage	Node Param.	2	3	4	5	6	7	Observations
1	Status	1	1	1	1	1	1	6 Working nodes (2 ZR and 4 ZED)
	Type	ZR	ZR	ZED	ZED	ZED	ZED	
	Current (mA)	40	40	10.88	10.88	10.88	10.88	Total Current 123.52mA
2	Status	0	1	2	2	1	1	3 Working nodes (1 ZR and 2 ZED)
	Type	ZR	ZR	ZED	ZED	ZED	ZED	
	Current (mA)	40	40	21.88	21.88	10.88	10.88	Total Current 145.52mA
3	Status	0	1	2	1	4	1	4 Working nodes (2 ZR and 2 ZED)
	Type	ZR	ZR	ZED	ZED	ZR	ZED	
	Current (mA)	40	40	21.88	10.88	40	10.88	Total Current 163.64mA
4	Status	0	1	1	4	4	1	5 Working nodes (3 ZR and 2 ZED)
	Type	ZR	ZR	ZED	ZR	ZR	ZED	
	Current (mA)	40	40	10.88	40	40	10.88	Total Current 181.76mA

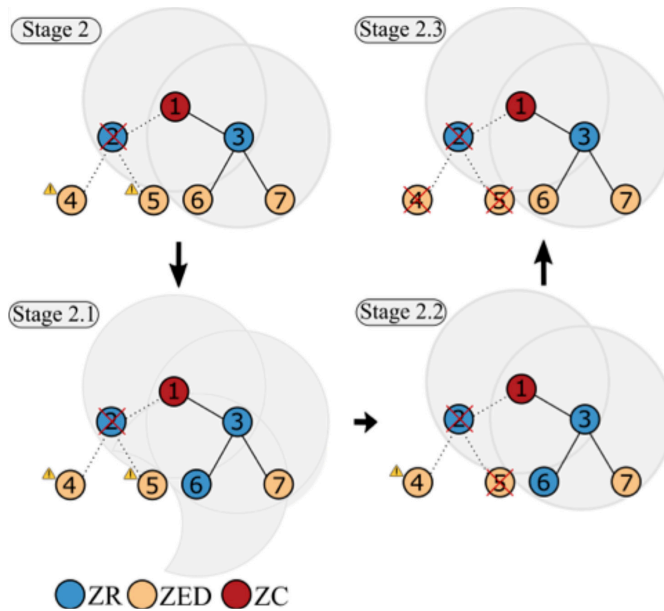


Figure 9. Case II.

As shown in Table 3, if the algorithm fails in restoring a node, the current is going to rise unnecessarily. This can be seen in stage 2.1, where the current rises from 145.52 mA to 174.64 mA because of the amount of ZRs in the network; thus, the algorithm keeps the original configuration of the nodes consuming once again 145.52 mA as in stage 2.3.

**Case III. ZR restores communication with the WSN**

Taking step 4 as the starting point, the algorithm will work as follows: 5) N2 restores communication with the coordinator. 6) The original children nodes

of N2 (N4 and N5) with status 5 will turn back into end devices. So, N4 loses communication with N5 and looks for a new parent, becoming N2’s child again. 7) The current parent of any original child node of N2 turns into ZR again if it is not the parent of another node and its status is 5. So, N6 turns into ZED, and N2 becomes the parent of N5 again. Hence, the network will re-establish its original configuration, as depicted in Figure 10.

In step 5 the network is completely functional, but it is composed of four ZRs and two ZEDs consuming

Table 3. Status per stage in Case II.

Stage	Node Param.	2	3	4	5	6	7	Observations
2	Status	0	1	2	2	1	1	3 Working nodes (1 ZR and 2 ZED) Total Current 145.52 mA
	Type	ZR	ZR	ZED	ZED	ZED	ZED	
	Current (mA)	40	40	21.88	21.88	10.88	10.88	
2.1	Status	0	1	2	3	4	1	3 Working nodes (2 ZR and 1 ZED) Total Current 174.64 mA
	Type	ZR	ZR	ZED	ZED	ZR	ZED	
	Current (mA)	40	40	21.88	21.88	40	10.88	
2.2	Status	0	1	2	0	1	1	3 Working nodes (1 ZR and 2 ZED) Total Current 145.52 mA
	Type	ZR	ZR	ZED	ZED	ZED	ZED	
	Current (mA)	40	40	21.88	21.88	10.88	10.88	
2.3	Status	0	1	0	0	1	1	3 Working nodes (1 ZR and 2 ZED) Total Current 145.52 mA
	Type	ZR	ZR	ZED	ZED	ZED	ZED	
	Current (mA)	40	40	21.88	21.88	10.88	10.88	



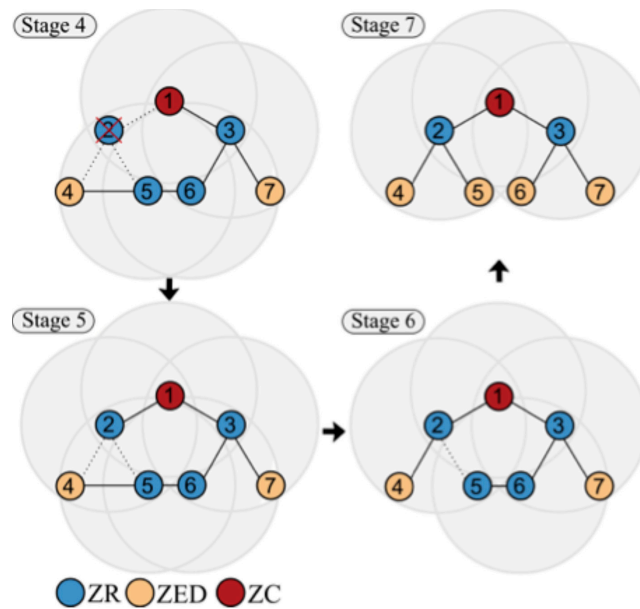


Figure 10. Case III.

181.76 mA as shown in Table 4; in contrast, in step 7 the network is composed of two ZRs and four ZEDs, reducing the consumption to 123.52 mA because the adaptive algorithm tries to keep the original topology as much as possible.

**Case IV. ZR loses communication with the WSN. With restoration of ZR**

If the network is larger, as shown in Figure 11, then the algorithm works in the same way. For instance, in this case, the WSN will behave as follows: 8) The network is working normally with all of its nodes.

9) N2 loses communication with the network, and therefore, its whole branch loses communication too. Thus, the network loses N2, N4, N5, N8, N9, N12, and N13, whose statuses change to 2. Thereby, the WSN is working only at 50% capacity. 10) The algorithm selects the closest node to the coordinator, being N2 in this case, and search for near ZEDs different to its original children with status different to 0. Again, as in step 2 shown in Figure 8, N2 has no near functional ZEDs; so, the algorithm looks for the next closest nodes, N4 has no near functional ZEDs either, but N5 does have

Table 4. Status per stage in Case III.

Stage	Node Param.	2	3	4	5	6	7	Observations
4	Status	0	1	1	4	4	1	5 Working nodes (3 ZR and 2 ZED) Total Current 181.76mA
	Type	ZR	ZR	ZED	ZR	ZR	ZED	
	Current (mA)	40	40	10.88	40	40	10.88	
5	Status	1	1	1	4	4	1	6 Working nodes (4 ZR and 2 ZED) Total Current 181.76mA
	Type	ZR	ZR	ZED	ZR	ZR	ZED	
	Current (mA)	40	40	10.88	40	40	10.88	
6	Status	1	1	1	1	4	1	6 Working nodes (3 ZR and 3 ZED) Total Current 152.64mA
	Type	ZR	ZR	ZED	ZED	ZR	ZED	
	Current (mA)	40	40	10.88	10.88	40	10.88	
7	Status	1	1	1	1	1	1	6 Working nodes (2 ZR and 4 ZED) Total Current 123.52mA
	Type	ZR	ZR	ZED	ZED	ZED	ZED	
	Current (mA)	40	40	10.88	10.88	10.88	10.88	

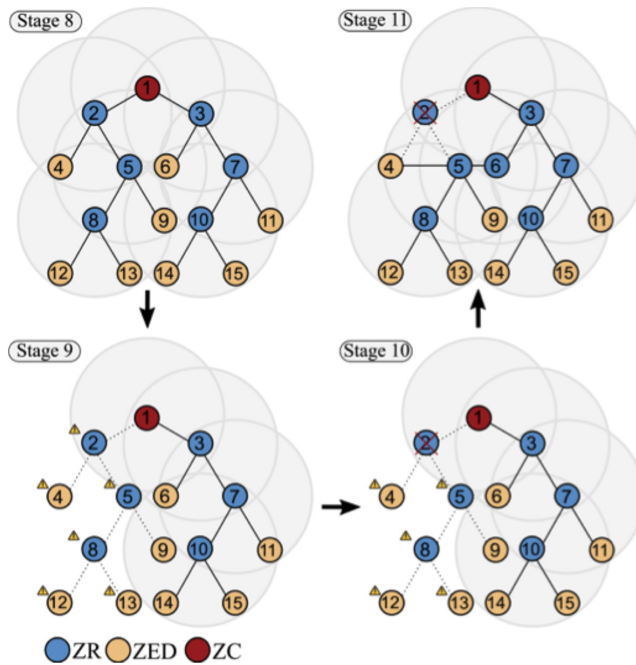


Figure 11. Case IV.

it (N6). N6 is turned into a ZR and becomes the parent of N5. After that, N5 and all its branches rejoin the network, recovering N4, N8, N9, N12, and N13. Now the network is working with 13 sensor nodes instead of 7.

### MOBILE APPLICATION

The mobile application was developed in Android Operative System following the proposed Android developer's community methodology. Using async tasks the mobile device can connect to the RPi's Wi-Fi Network. When the status of each node changes, a notification is sent to a mobile application. Furthermore, every node's status is shown to the user on a list (Figure 12), allowing the network behavior monitoring. If the mobile application is connected to the Wi-Fi network of the RPi, then the information about the nodes is updated in the smartphone (Figure 12.a). Otherwise, the stored data in the app is shown to the user (Figure 12.b).

Each node location is shown in the app using the Google Maps API and considering the latitude and longitude values stored in the database. The final user interface for geographic location is shown in Figure 13.

### CONCLUSIONS

By exploiting the ZigBee protocol's functionalities, we obtained versatile nodes that may switch their role depending on the situation and can adapt to keep its integrality as much as possible. In this way, we were able to recover up to 42.8% of the network coverage. As an improvement to other works reported in the literature, the proposed algorithm offers an automated solution that helps mitigate the problem of the orphan nodes by maintaining the network's integrity and coverage as much as possible using the available nodes.

The proposed algorithm presented in this work may be useful in future studies to optimize the integrity and coverage of a WSN, but it comes with an energy cost. As shown in section III, to recover a node, a ZED needs to switch to a ZR and become a parent, increasing the current consumption, in this case, from 10.88 mA to 40 mA. Although the energy consumption varies depending on the devices used, the environment, and the sampling rate, it is still the same principle. The algorithm works based on the location and minimum distance for communication between nodes.

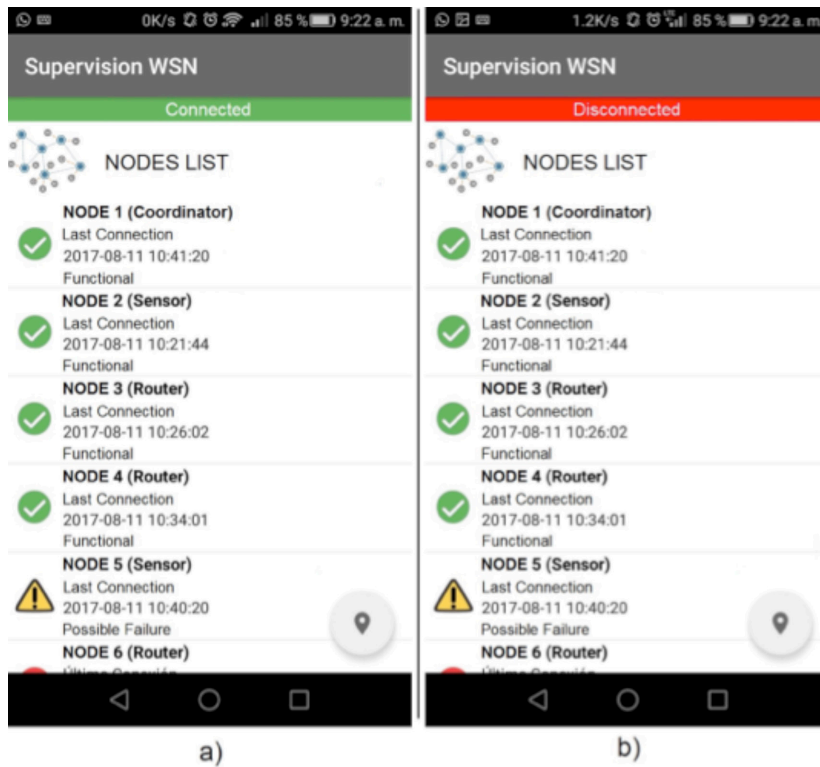


Figure 12. Nodes List in the app. a) Connected. b) Disconnected.



Figure 13. Geographic location in the mobile app.

The mobile application allows the researchers and users to monitor the network behavior and take action in case of a failure. Also, a

notification service in the app is necessary to timely alert the users about the change of status on each node.

## REFERENCES

- [1] J. Kanno, J.G. Buchart, R.R. Selmic, and V. Phoha. "Detecting coverage holes in wireless sensor networks". 2009 17th Mediterr. Conf. Control Autom., pp. 452-457. 2009.
- [2] S.B. Qiu, Y. Xu, and X.W. Yang. "A novel cluster head selection method using energy for ZigBee cluster-tree network". IEEE Int. Conf. Autom. Logist. ICAL, N° 2010, pp. 178-181. 2011.
- [3] D. Jiang, L. Yu, F. Wang, X. Xie, and Y. Yu. "Design of the smart home system based on the optimal routing algorithm and ZigBee network". PLoS One. Vol. 12 N° 11, pp. 1-15. 2017.
- [4] E. Kabalci and Y. Kabalci. "A wireless metering and monitoring system for solar string inverters", Int. J. Electr. Power Energy Syst. Vol. 96, September 2017, pp. 282-295. 2018.
- [5] J.C. Chou, S.J. Yan, Y.H. Liao, C.H. Lai, Y.X. Wu, and C.Y. Wu. "Remote Detection for Glucose and Lactate Based on Flexible Sensor Array". IEEE Sens. J. Vol. 18 N° 8, pp. 3467-3474. 2018.
- [6] S. Ali, *et al.* "SimpliMote: A Wireless Sensor Network Monitoring Platform for Oil and Gas Pipelines". IEEE Syst. J. Vol. 12 N° 1, pp. 1-12. 2016.
- [7] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. "An application-specific protocol architecture for wireless microsensor networks". IEEE Trans. Wirel. Commun. Vol. 1 N° 4, pp. 660-670. 2002.
- [8] F. Awad, E. Taqieddin, and A. Seyam. "Energy-Efficient and Coverage-Aware Clustering in Wireless Sensor Networks". Wirel. Eng. Technol. Vol. 3, pp. 142-151. 2012.
- [9] E. Taqieddin, M. Mowafi, F. Awad, O. Banimelhem, and H. Maher. "An energy-balanced cluster-based protocol for wireless sensor networks". Int. J. Inf. Technol. Web Eng. Vol. 8, N° 3, pp. 50-63. 2013.
- [10] E. Taqieddin, F. Awad, and H. Ahmad. "Location-Aware and Mobility-Based Performance Optimization for Wireless Sensor Networks". J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl. Vol. 8 N° 4, pp. 37-59. 2017.
- [11] J. Mu, W. Wang, B. Zhang, and W. Song. "An adaptive routing optimization and energy-balancing algorithm in ZigBee hierarchical networks". J. Wirel. Commun. Netw., pp. 1-11. 2014.
- [12] I. Ullah and H.Y. Youn. "Statistical Multipath Queue-Wise Preemption Routing for ZigBee-Based WSN". Wirel. Pers. Commun., pp. 1-15. 2018.
- [13] S. Ozdemir and Y. Xiao. "FTDA: outlier detection-based fault-tolerant data aggregation for wireless sensor networks". Secur. Commun. NETWORKS. Vol. 6 N° 6, pp. 702-710. 2013.
- [14] D. Liu, Y. Gao, and Y. Liu. "An improved orphan nodes-reduced mechanism in zigbee network". Vol. 542-543. 2012.
- [15] M.D. Sarr, F. Delobel, M. Misson, and I. Niang. "Automatic and dynamic network establishment for linear WSNs". Wirel. Networks, pp. 1-14. 2017.
- [16] S.-C. Hu, C.-K. Lin, Y.-C. Tseng, and W.-T. Chen. "Distributed address assignment with address borrowing for ZigBee networks". In 2014 IEEE International Conference on Communications Workshops, ICC 2014, pp. 454-459. 2014.
- [17] L. Bai and X. Zhao. "Borrowed address assignment algorithm based on recursive reduction model in zigbee networks". In 14th IEEE International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, pp. 1240-1245. 2018.
- [18] X.W. Yang, Y. Xu, and S.B. Qiu. "A solution of orphan problem in ZigBee tree network". 7th Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCOM 2011, pp. 1-4. 2011.