INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Implementation of scalable and energy efficient WSN platform for IoT applications

Junaid Ahmed Uqaili[1]*, **Saleemullah Memon[1]**, **Hafiz Muhammad Bilal[1]**, **Sanaullah Dehraj[2]**, **Muhammad Javed[3]**, **Muzamil Hussain[4]**, **Muhammad Abbas[4]**

1 School of Electronic Engineering, Beijing University of Posts & Telecommunications, Beijing, China
2 Department of Mathematics and Statics, Quaid e Awam University of Engineering, Science and Technology, Nawabshah, Pakistan
3 College of Computer Science and Information Engineering, Guangxi Normal University, Guilin, China
4 School of Computer Science, Beijing University of Posts & Telecommunications, Beijing, China

Abstract

Background/Objectives: The major problems in WSN are the short-range of RF signals and the short life of batteries of each node. We develop a platform that uses the MAC protocols to avoid collisions between packets and decrease the data collection time. To overcome the energy challenges, the sleep modes, idle consumption, and others solutions provided to extend the node's life period **Methods/Statistical analysis:** Based on the IoT applications, we configured different hardware (HW) to collect data from temperature, relative humidity (RH), and carbon dioxide (CO_2), and access to this data, and these devices remotely through the internet. For this, the nodes, transceivers, batteries, and the gateway (GW) are analyzed. Additionally, it is configured with the frequency band or the radio frequency (RF) sensitivity to improve the performance of the system developed. **Findings:** Sensors, repeaters, and the GW were configured to work with each other and be accessed through the internet. The energy consumed on the experiment carried out has been reduced by more than 80% choosing LP modes and enable sleep algorithms over the Node's OS where the time was also reduced around 80% in comparison with the available options. By set MQTT and SSH services, this application is scalable to be integrated with cloud services and be accessed by remote computers. **Novelty/Applications:** Finally, the lifelong period of nodes augmented drastically, more range between nodes is achieved and highly reliable data collected with low power consumption.

Keywords: WSN; sensors; IoT; CO 2; temperature; relative humidity (RH)

1 Introduction

Wireless sensor networks (WSNs) become a hot research topic in various fields such as communication, smart health, industry, agriculture, monitoring, and surveillance, etc. WSN is configured with sensors, actuators, and transceivers. The sensors and actuators are used to collect physical data from different sources which are stored and processed intelligently at the remote server and conveyed through transceivers. The most common internet devices are connected wirelessly through Wi-Fi or cellular communications. The Fifth-generation (5G) of network cellular communication can connect and communicate billions of internet of thing (IoT) devices [1,2]. The improvement of architecture and protocols, as well as mechanisms, is very strong where different solutions are provided and a vast amount of research is done. For example, the Time-division multiple access (TDMA) protocol to route packets through the network, which has several advantages compared with the carrier sense multiple access with collision avoidance (CSMA/CA) protocol it makes sure that each node can access to the medium to communicate with the others, and is deployed by networks to route the packets through its nodes. The microcontrollers are deployed as a control system, which provides an interface application to data processing and creates data dashboards, and nodes which has deep sleep algorithms to prolong their lifetime [3].

Sensor nodes are considered as smart devices and widely used to direct the sensing information. However, they can be susceptible to a variety of security intimidation. Therefore, with the implementation of the secret sharing scheme [4] studies an energy-aware and secure multi-hop routing protocol to increase the energy efficiency performance against malicious adversaries with multi-hop data security. The same issue has been analyzed in [5] and [6]. With the integration of WSN, the authors in [5] have proposed a framework for intrusion prevention of mobile IoT devices. In order to enhance the data reliability, network period and intrusion avoidance in IoT based WSN, the authors have developed an energy-efficient and secure routing protocol in [6]. Furthermore, to balance the load distribution and energy consumption using WSN supervised system, the authors in [7] have proposed an energy-aware graph clustering and intelligent routing design, where a secure and effective key distribution in a hierarchy-based mechanism is considered a solution to improve the network efficiency in terms of routes and links integrity.

In order to reach the demands of the next IoT applications that require a huge number of interconnected wireless network nodes, an autonomous energy-efficient WSN platform for home/office automation applications is investigated in [8] using an optimized power management strategy and an embedded photovoltaic panel linked to a rechargeable battery. To flexibly design the applications of high-level-based techniques, a detailed study of model-based approaches and existing programming methodologies for the development of sensor networks is given in [9]. The WSNs used in the agriculture field have limited resources of transmitting, processing, receiving, memory capabilities, and energy that will cause serious influences on the agriculture production. The safety, efficiency, and security of these IoT-based agriculture sensors play a vital role from malicious adversaries. Hence, a framework for secure and an energy-efficient IoT-based WSN for smart agriculture applications is proposed in [10]. An efficient solution for multiple data collection tasks exploiting edge computing-enabled wireless sensor networks in smart agriculture is presented in [11].

The two routing schemes named as and cooperative effective energy and reliable delivery and effective energy and reliable delivery are studied in [12] to deal with the adverse channel issues that corrupt the packets and challenge their reliability. A sustainable IoT sensing application through a graph-based abstraction layer is studied in [13] to obtain an effective data management and energy conservation in IoT sensing applications. The implementation of a dynamic clustering computing-based WSN genetic algorithm for IoT applications is studied in [14] to support the applications in the real-world such as smart cities, smart transportation, and smart grid in an effective manner.

Due to its easy deployment, scalability, and low cost, this network technology is implemented in a broad variety of areas such as industrial automation, logistics, smart grids, smart cities, healthcare, and others, where developers take part in. On the other hand, the Internet of things (IoT) and Industry 4.0 are strongly related to network technologies. Which requires huge amounts of data to make important decisions. Besides that, data collecting needs reliable and robust systems where wireless technology, as well as other technology, provides different alternatives. To create these systems, developers have to choose between a broad variety of devices, operating modes, architectures, and others, with the flexibility that these technologies provide and the complex decision due to the vast number of them. Moreover, the implementation of these networks must be optimized to assure the accomplishment of the mission of these applications to avoid issues such as overhead energy consumption by nodes, data contamination due to interference or messages lost along the way. The main contributions of this paper are mentioned below:

- Configured the HW used on the WSN and selected proper software to configure the devices deployed.
- Set up the GW that includes the control system used to manage the WSN and performs as a bridge between the WSN and the internet. Also, automate the process of collecting data implemented.
- Programming the data processing needed to store and design an interface to visualize data.

- Evaluate the HW configuration and implement an IoT application for a WSN also assess the data collected by the network created.

The rest of the paper is organized as follows. In section 2, the WSN design is discussed and it includes HW and SW types, processes, and configurations of network devices. The experiment and results are discussed in section 3, also the routing process, performance of the network, live data dashboard, and shows the data collected by different sensors. In section 4, the detailed discussion of deployed WSN is discussed. Finally, section 5 concludes this work.

2 Wireless sensor network design

WSNs are composed of several nodes, the sink, which can be a GW, and the user. Nodes designed to perform over a WSN are sensors or repeaters with low power capabilities and short RF ranges to communicate with each other, which acts as a transmitter and receiver allowing bidirectional communication [15]. The sink/GW receives the data packets collected by the sensors, having a centralized network. Finally, the user accesses the WSN through the internet or other networks [2,16], see Figure 1.
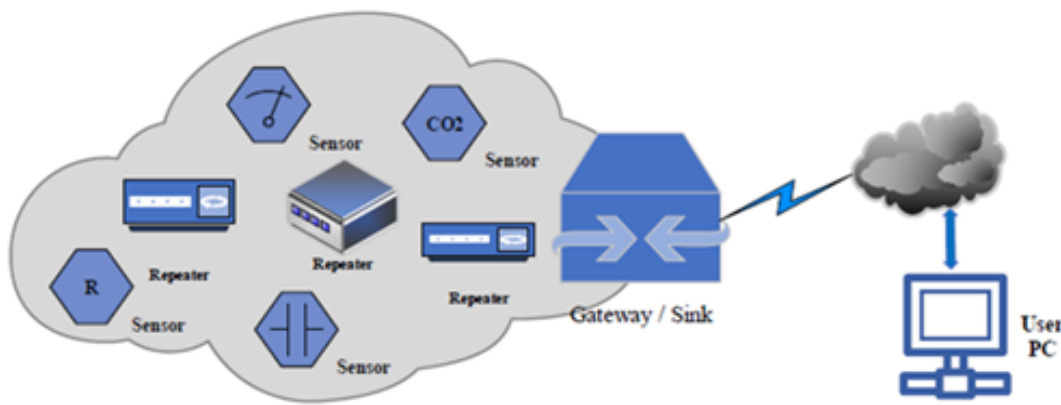

Fig 1. WSN scheme

The main problems in WSN networks are the short range of the nodes to communicate with each other, and the short lifelong period of the batteries of each node. Besides, to solve these problems we deploy intermediate nodes [17], the limit on the RF range of the nodes the proper topology is determined by the positions of the nodes and its features. On the other hand, communication requires synchronization over the network to send and receive packets among devices. Medium access control (MAC) protocols, routing protocols, and data collecting methods are implemented to avoid collisions between packets and decrease the time of data collecting [2]. Furthermore, low power capability is a fact present over a WSN, where communication protocols are developed based on this requirement and special operating modes are available on devices deploying this technology, where sleep modes, idle consumption, and others are solutions provided by commercial devices to prolong the lifetime period of the network's node [18]. The nodes architecture of a WSN network is composed of five components i.e. sensors and actuators, controller, communication device, power supply, and memory as shown in Figure 2.
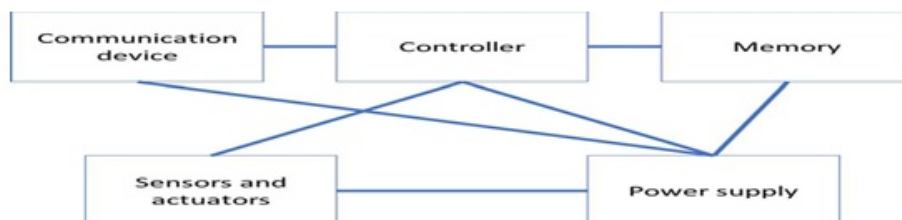

Fig 2. Main hardware components and connections of WSN

The controller exchanges the data with transceiver/communications devices and controls the behavior of actuators. The microcontroller exists inside controllers which provides user flexibility and is connected with the power supply to reduce energy consumption through sleep states. The data and programs are stored in memory controlled by the controller.

2.1 Hardware and Software Types and Process

Sensors, transceivers, and different development kits are used in this work. The SN-THC-0$_2$ sensors are used which measures Temperature, RH, and CO_2 concentration sensing. The transceiver module is TR-72DAT. The specifications and features of the hardware used in this work are as[19,20]. The sensor communications work under the Direct Peripheral Access (DPA) protocol, and the Intelligent Mesh (IQMESH) protocol. IQMESH is based on the TDMA protocol and the flooding mechanism[21]. On the other hand, DPA is a byte-oriented protocol, which means that the devices of the network are controlled through byte structured messages which are sent through the network. To determine the behavior of each node, that works under the DPA protocol, IQRF plug-ins are provided by this protocol, which can be uploaded on the microcontroller of the TR[22].

- The development kit DK-EVAL-04A is used, it contains an accumulator battery and a subscriber identity module (SIM) connector to host IQRF TRs. Therefore, this component is compatible with the TRs provided. The main features checked to analyze the device and its compatibility with the TR[23].
- The Programmer and debugger kit CK-USB-04A allows programming the OS of the MCU[24]. This kit contains a SIM connector to host the TR and a micro USB port to connect this module to the PC.
- The gateway GW-IND-01-ETH is used, it provides the connection between the WSN and the internet due to the connection through Ethernet[25].

The IQRF IDE 4 software is used to configure the WSN devices, it provides a graphical user interface (GUI). It is also used to configure the MCU upon the IQRF TRs which allows uploading DPA plug-ins on the OS of this MCU[22]. The GW is configured through Link It! software. Moreover, for implementation of data processing, storage, and visualization the Node-Red programming tool was installed on the GW[26].

2.2 HW Configuration and Tests

To configure the devices, several steps followed, and different tests carried out to assure a correct procedure the flowchart is shown in Figure 3. Access to the LAN is also allowed on the GW to be discovered by Link It!. After that, the configuration of the GW started, where the control system of the WSN and the connection to the internet are configured. The repeaters are set where each repeater is uploaded with a DPA plug-in to link the coordinator. The IQRF IDE deployed, it is connected to the GW through the Link It! SW via LAN. Once the nodes of the network linked to the coordinator, several communication tests are carried out to check the correct operation of the WSN. As communication is fixed successfully, data processing, storage, and visualization are programmed through Node-Red. Finally, the data storage, visualization, and communications between the control system and the RF network are checked

- Allow access to TCP / UDP ports
  The GW is connected to the LAN through an Ethernet wire and it is accessible from the LAN. The UDP port 1313 configured to discover the GW through Link It! and the access given to the outgoing TCP ports 1883 and 8883, which provides the connection to the MQTT services. The IP from the GW over the LAN is fixed with the IP address 130.243.14.22.
- Gateway
  The GW is configured using Link It! and the WSN control system is set on the GW. The request-response pattern is selected in the control system visualization through Link It! where the coordinator sends a request to the network, and it receives the data from the sensors, which works together with the TDMA protocol. Moreover, the fast response commands (FRC) technique is used for data collection, it is faster than the polling method[27]. It takes a shorter time to collect data and saves energy consumption per node. The time duration used to collect data through the FRC and LP mode can be calculated by:

$$t_{FRC} = N*30 + (N+2)*160 + t_{proc,FRC} + 26 \tag{1}$$

Where t$_{proc, FRC}$ is the value of the FRC time slot, which is 40ms by default on IQRF TRs[28].
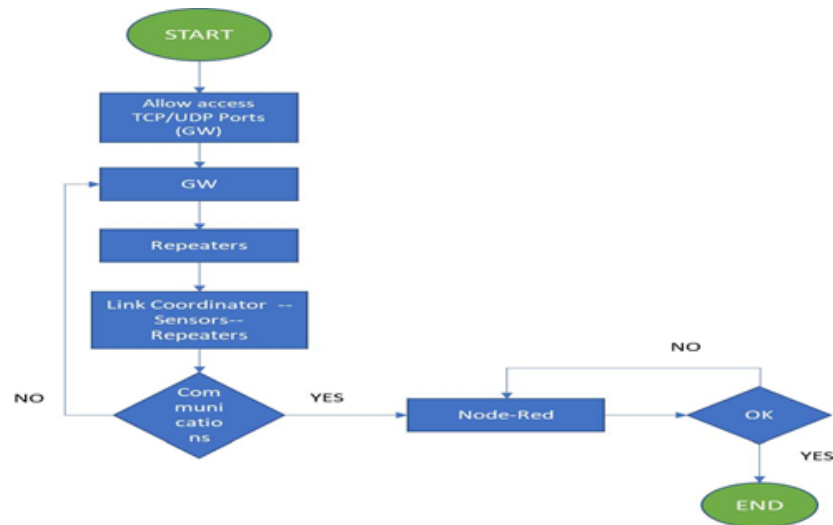
Fig 3. Workflow to configure the hardware devices

- **Repeaters**
  The repeaters deployed are created by plugin the transceiver (TR) TR-72DAT into the battery kit. The TR connected to the IQRF IDE by connecting the TR to the programmer and debugger kit (CK-USB-04A) and the laptop through a micro USB, where the IQRF IDE is running. Each repeater is uploaded with an IQRF plug-in, which is responsible for the behavior of the MCU.

- **Link sensors and repeaters to the coordinator**
  Each sensor and repeater bonded to the coordinator through the IQRF IDE. The GW hosts the coordinator. The Link It! is connected to the GW through the same LAN and the computer is provided with the IP address 172.16.116.89. Therefore, the IQRF IDE and Link It! connected through the UDP port 55000, where the IQRF IDE was running, and the UDP port 55300, where the Link It! was performing, shown in Figure 4.

- **Node-Red**
  Node-Red deployed in this work to store and visualize the data. The performance of the network displayed on the live data dashboard was created visualized through Node-Red. It is also used to analyze the performance of the network and ease the process of setting up the network on the place selected to be analyzed and installed. Detection of failures on the WSN and the control system programmed is detected through the live data dashboard created. This dashboard developed with an additional tab where an analysis of the network is carried out.

Fig 4. Connection between IQRF IDE and Link It!

3 Experiment and Results

In this work, the low power SN-THC-02 sensors are used that measure temperature, relative humidity (RH), and CO_2. Several wireless sensors are placed in different positions. The data collection is done for 28 days every 60 seconds. Deployed devices use RF low power wireless technology and the mesh topology is used in this network, where the data packets are sent among the coordinator and the sensors taking multi-hops, to cover the whole area. Figure 5 shows the repeaters and sensors are installed in the buildings.
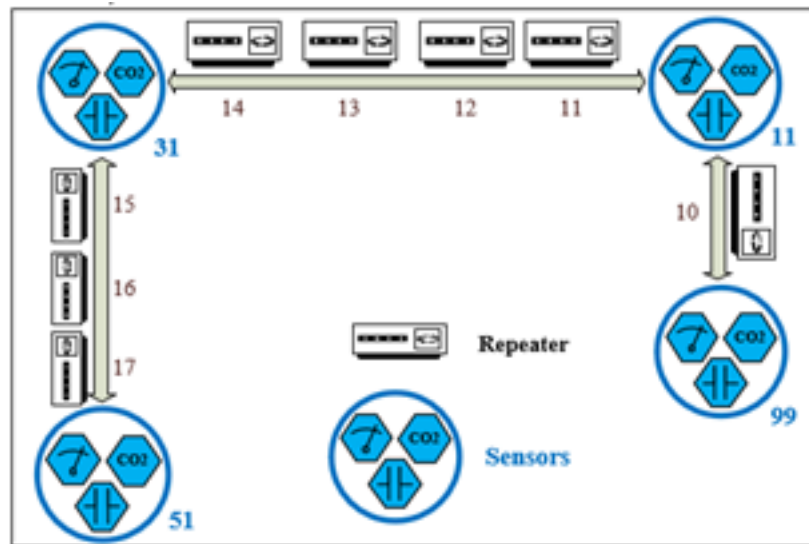

**Fig 5.** The installed WSN network contains sensors and repeaters

3.1 Routing process

After placing each sensor and run the discovery process, that creates the routing structure of the WSN, sensors from building 99 and 11 are the only ones accessed by the coordinator due to the low distance between the sensors and the GW. Based on the requirements to reach sensors from the other sensors, several repeaters are included in the network. The repeaters are used at different positions based on requirement and feasibility position. After comparing different positions of repeaters, the final positions are observed with no loss of packets and a redundant path basis. Therefore, if any issue occurred on one node means no losing the communication between different zones.

3.2 Performance of deployed WSN network

To be able to assess the performance of the WSN implemented, a program through Node-Red was developed. This program calculates the percentage of messages received by the control system in comparison with how many it must receive, which was repeated every 5 minutes. In this experiment, the number of messages received by the control system must be 3 every 60 seconds, once per each type of parameter. To evaluate the performance of the network, two random days are analyzed through 100 samples calculated from 8 am to 6 pm. The X-axis indicates the percentage of messages received by the control system. Y-axis indicates the number of samples that have the percentage of messages indicated in the horizontal axis.

Figures 6 and 7 show that the control system collected 90% and 92% of the messages during day 1 and day 2 respectively. Whereas, Figure 8 shows 95% of the messages collected by the control system. During this evaluation, some samples are higher than 100% due to the clock created through Node-Red to calculate the percentages. The Node-Red clock programmed was not synchronized with the clock of the control system. Therefore, this asynchronous clock makes that the counted messages can have differences in respect to reality.
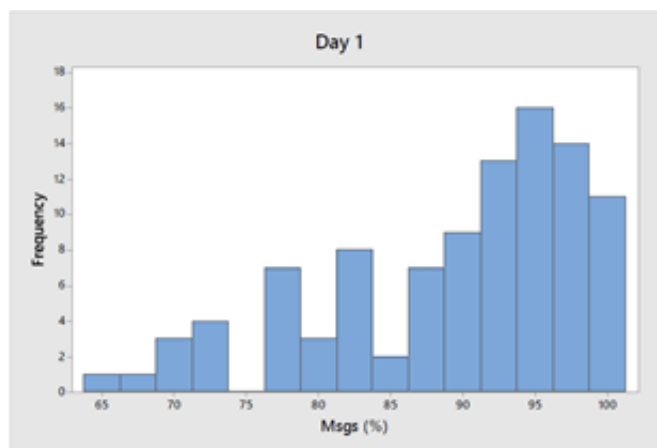
Fig 6. Messages(%) collected for 100 random samples from 8 am to 6 pm. (Day 1)
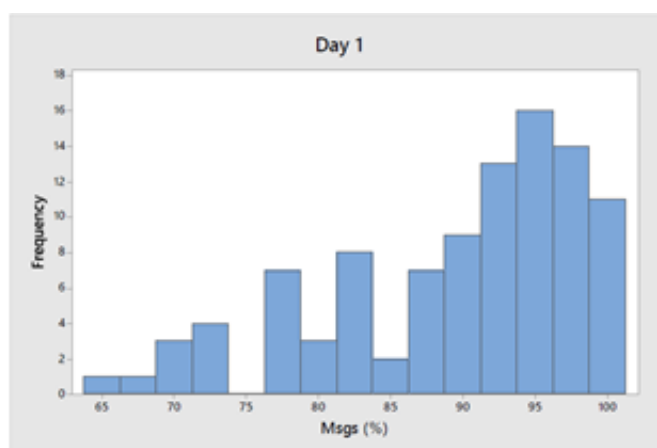
Fig 7. Messages(%) collected for 100 random samples from 8 amto 6 pm. (Day 2)
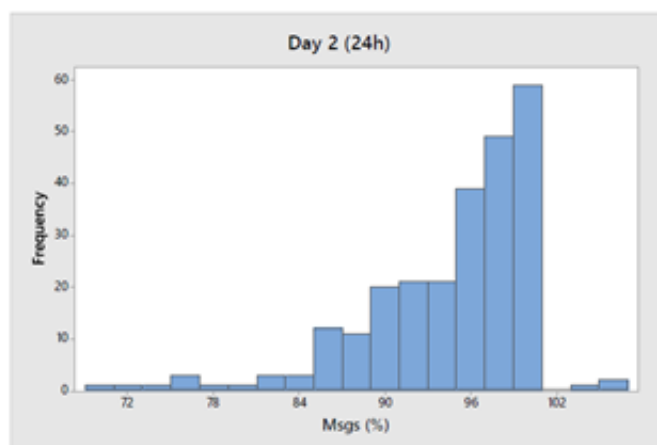
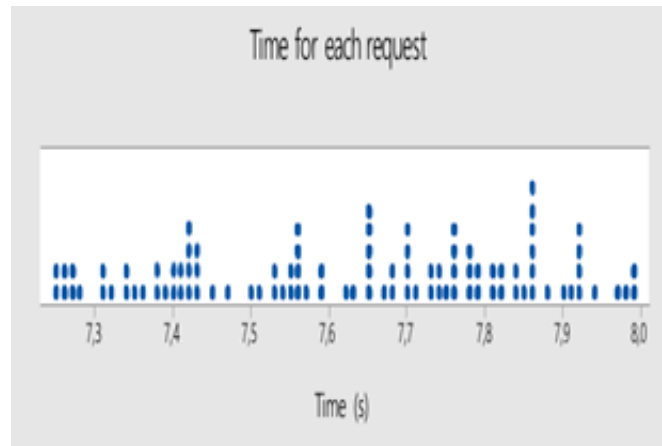**Fig 8.** Messages(%) collected for 250 random samples during 24 hours (Day 2)

Fig 9. Time deployed by the WSN to response the request from the control system

Figure 9 shows the time deployed by the WSN to send the measured values to the control system, where the average of 100 random samples is 7.62 seconds. These samples are taken from the requests stored on the CSV file, where the time of each response is stored. This time to collect the data is higher than the 3.9s predicted by equation 1, considering that the WSN created has 17 nodes. The reason can be due to the time deployed by the SPI communication between the coordinator and the control system and also the time to copy the data in the buffer of the GW. However, checking the data saved in the CSV file, the time deployed to collect the data by the control system is about 4 seconds on most occasions, which is closer to the time predicted.

3.3 Live data dashboard

To check the function of the live data dashboard developed through Node-Red, the interface was accessed through (http://130 .243.14.22:1880/ui). The CPU usage shows that the GW was optimized after increase the synchronization time of the live data dashboard and debug the code of the application on Node-Red. Once the modifications on Node-Red was carried out, the user could visualize the live data dashboard and the data displayed, shown in Figure 10.
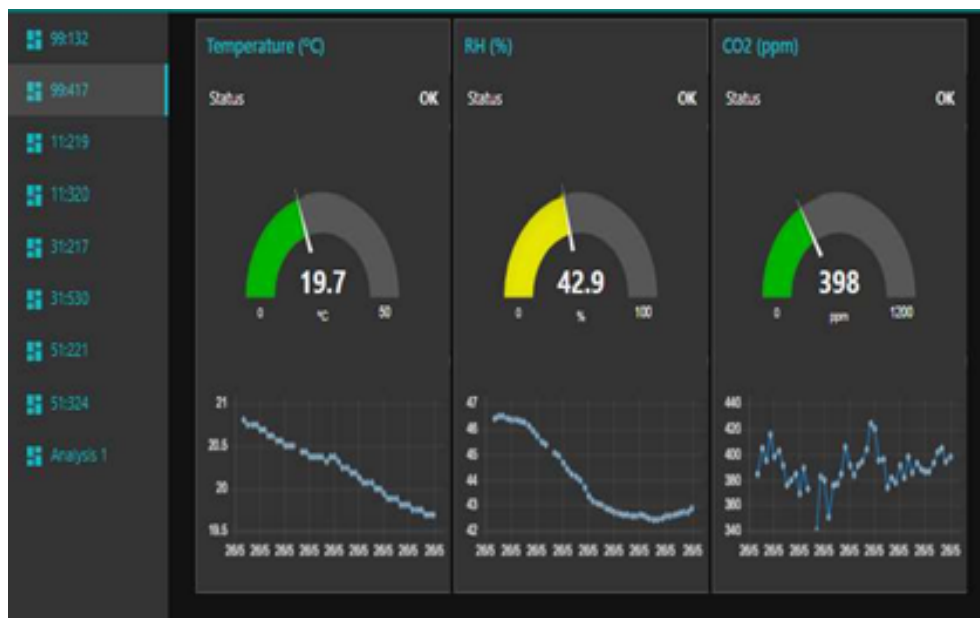

Fig 10. "Analysis1" tab displayed by the live data dashboard

The RF interference is tested by running the RF signal tool several times after each modification of the network and during its operation. During this test, no interference on the frequency band 868 MHz and channel 52 (868.35 MHz) was detected through this tool provided by IQRF IDE. The dashboard produced contains nine tabs, one tab for each hall, and the other one to analyze the network, which is called "Analysis 1". Each hall's tab displays the measurements of temperature, RH, and CO_2 concentration in real-time and the measures for seven days ago. Nevertheless, if the GW is reset or disconnected, the information of the graphs is lost, as can be seen in Figure 10, where only the data of one day can be shown. Additionally, the status text shows if the last measurement was collected or not. Besides that, "Analysis 1" allows the user to analyze and visualize the status of the network for each sensor and each type of data, see Figure 11.

Fig 11. Live data dashboard developed through Node-Red

The live data dashboard works quite well-performing with some users, where the GW acts as a server. On the other hand, the visualization interface cannot be deployed for several users, and the GW gets collapse when the synchronization of the data is under 1 minute. Data visualization collected by the sensors was displayed, which is synchronized with the data collected every 5 minutes, and the status of the network as well. Additionally, the user or the developer can run a custom analysis of the network, calculating the percentage of messages received by the control system in total, for each sensor, and each parameter, through the Analysis 1 tab of the live data dashboard.

3.4 Data collected

To evaluate the temperature ($^{\circ}$C), RH (%), and CO_2 concentration (ppm) levels of each hall, where data was stored in a CSV file format on the USB connection to the GW, the data can be used to create statistical measures to analyze the halls selected. Figures 12 and 13 shows the CO_2 concentration levels (ppm), the difference between the closest point shows the accuracy of the CO_2 sensors, which is of 50 ppm[27].
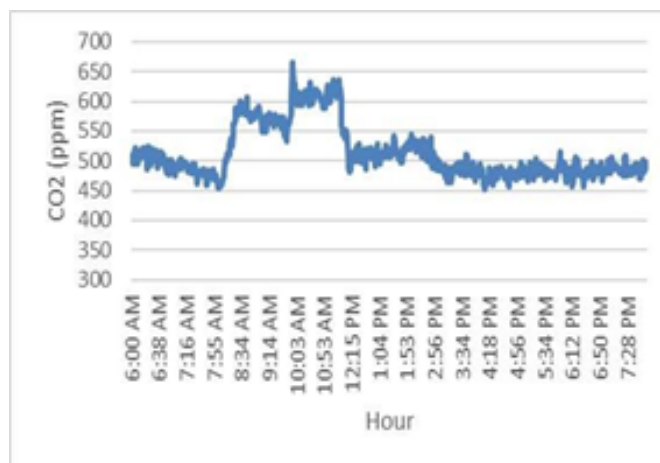
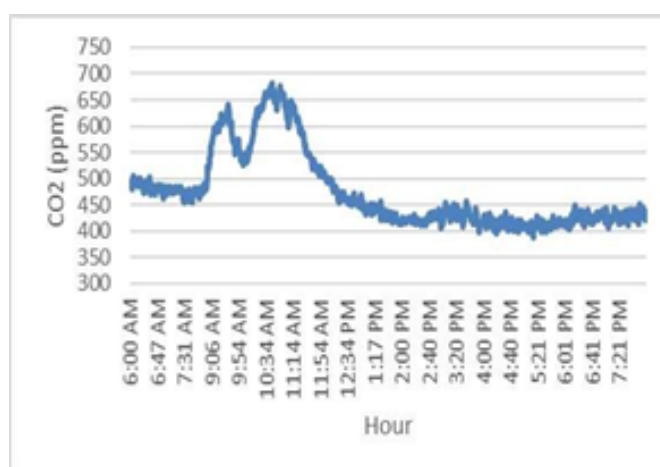**Fig 12.** CO_2 concentration levels (ppm) at hall 99:417

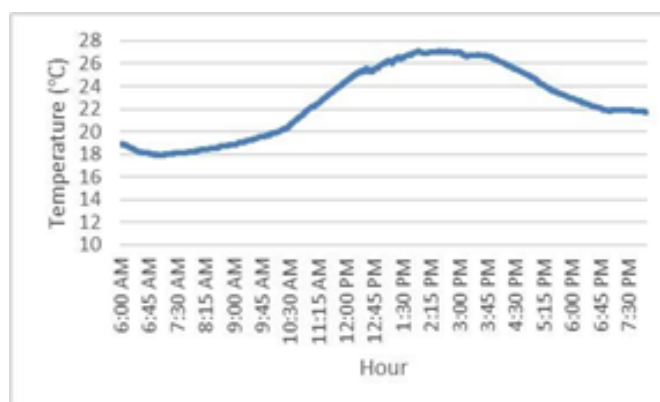**Fig 13.** CO_2 concentration levels (ppm) at hall 51:221

Fig 14. Temperature(ºC) levels at hall 11:320

Figures 12, 13, 14, 15, 16 and 17 shows that the data stored in the CSV file, where the data processing and the storage of the data related to the file done through Node-Red, an assessment of the different halls/buildings can be done.
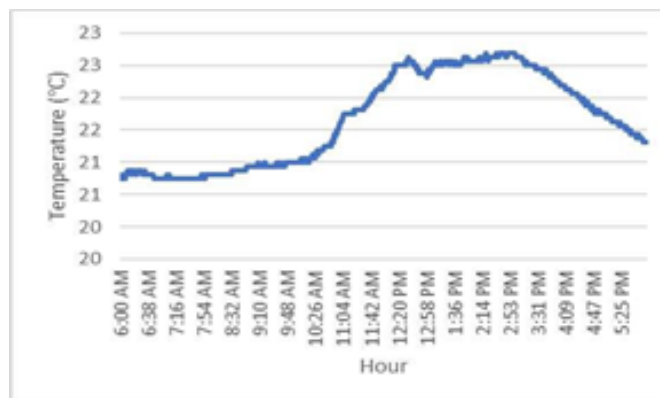
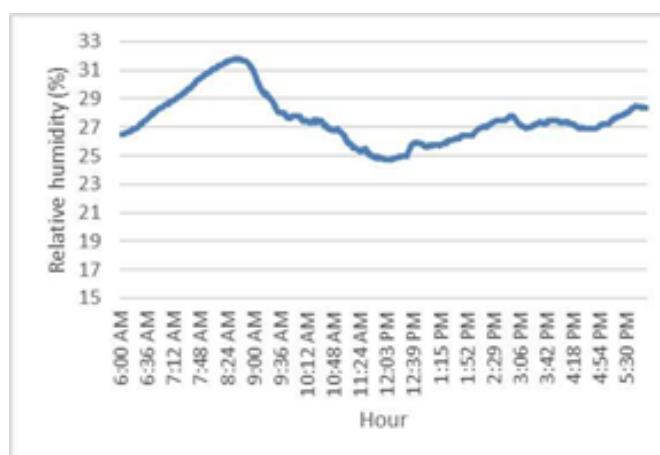**Fig 15.** Temperature(ºC) levels at hall 51:530

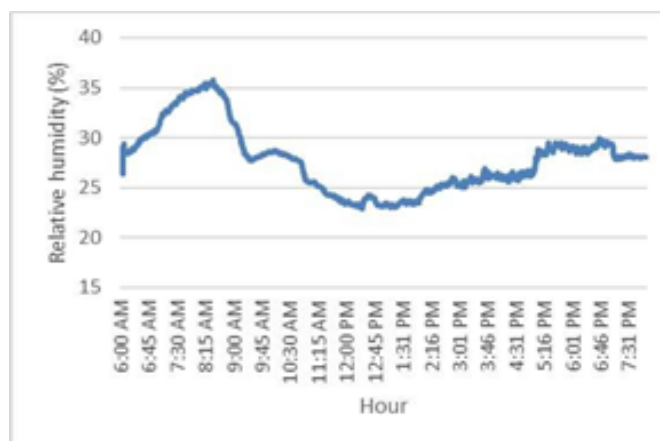**Fig 16.** Relative humidity (%) levels at hall 99:132

Fig 17. Relative humidity (%) levels at hall 31:217

4 Discussion

Different devices were configured on this work to be applied in similar IoT applications for WSNs. This approach was followed by finding the trade-off between robustness and flexibility. From the connection between the WSN and the Internet, MQTT, and SSH services configured through Link It!. These services allow the user or user's applications to be connected to the GW remotely, the SSH services used for security reasons. TRs from the repeaters configured is set on the 868 MHz license-free ISM frequency band. It provides a higher effective radiated power, therefore, the higher range in this direction with the appropriate orientation of the sensors and transceivers. Moreover, the LP mode and the sleep mode are also enabled after complete the three requests by programming the control system. With that, lower consumption of the nodes and a more extended lifetime period reached. To check the performance of the WSN the "Analysis 1" tab is used for the live data dashboard.

From experimental results, it shows that the percentage of the messages received for the samples analyzed was higher than 90%. These results in applications of data collecting about temperature, RH, and CO_2 concentration levels are quite good, and increasing the number of the transmissions, which affects negativity on energy consumption, is an option. This result on the performance strengthens the followed method where we have created the analysis 1 tab to check the performance and provides the user with visual information about the percentage of messages received by the control system. Nevertheless, the graphs created through the data collected provides the quality to analyze the values measured by the sensors. The deployment of the FRC method to collect data reduced the number of transmissions and the time of data collecting in an approximately 80 %, in the experiment carried out, which means that the energy consumption was decreased significantly, where the method to choose this mechanism based on manuals provided with devices. Eventhough the average time to collect data is about 7.6seconds when the method deployed to calculate this time indicates 3.9seconds. The differences could be produced by not consider the SPI communication between the coordinator and the control system and the time to copy the data in the buffer of the GW. However, making an overview of the data saved in the CSV file, the time in most occasions has been 4seconds, which is close to the time predicted.

The data processing implemented through Node-Red is scalable in the number of nodes and services and has allowed storing data with all the information required but it might be preferable to store this data in a cloud service or upon a server to increase the scalability of the application on that case. The GW deployed has not enough capacity to perform as a server for several users, where it would be better to externalize data processing, storage, and visualization. Cloud Services would be allowed to increase the scalability but small applications can be better this structure, which ease the implementation and the deployment of the system. It was seen that a tool to make the recovery of the GW is needed due to that the GW has not this option for security reasons. Failures on the OS means that the application running over the GW can be lost. It should be noticed that the method followed in this paper is in concordance with target 9.4 of the 2030 Agenda for sustainable development[28].

5 Conclusion

This work has provided a set of configured HW to deploy IoT applications for WSNs. The configuration reduces the amount of time required to implement these devices by a developer and ease its implementation through a flexible and robust system. Sensors, repeaters and the GW were configured to work with each other and be accessed through the internet. The energy consumed on the experiment carried out has been reduced by more than 80% choosing LP modes and enable sleep algorithms over the Node's OS where the time was also reduced around 80% in comparison with the available options. By set MQTT and SSH services, this application is scalable to be integrated with cloud services and be accessed by remote computers. Therefore, this HW could be implemented in several in-building and other environments to be analyzed through data accessed via the internet providing a real IoT application.

Acknowledgement

References

1) Lazarescu TM. Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. 2013;3(1):45–54. Available from: https://dx.doi.org/10.1109/jetcas.2013.2243032.
2) Karl H, Willig A. Protocols and Architectures for Wireless Sensor Networks. Chichester, UK. John Wiley & Sons. 2005. Available from: http://doi.wiley.com/10.1002/0470095121.
3) Cionca V, Newe T, Dadârlat V. TDMA Protocol Requirements for Wireless Sensor Networks. In: and others, editor. 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008). IEEE. 2008;p. 30–35. Available from: http://doi.org/10.1109/SENSORCOMM.2008.69.

4) Haseeb K. Secret Sharing-Based Energy-Aware and Multi-Hop Routing Protocol for IoT Based WSNs. *IEEE Access*. 2019;7:79980–79988. Available from: http://doi.org/10.1109/ACCESS.2019.2922971.

5) Haseeb K, Islam N, Almogren A, Din IU. Intrusion Prevention Framework for Secure Routing in WSN-Based Mobile Internet of Things. *IEEE Access*. 2019;7:185496–185505. Available from: https://dx.doi.org/10.1109/access.2019.2960633.

6) Haseeb K, Almogren A, Islam N, Din IU, Jan Z. An Energy-Efficient and Secure Routing Protocol for Intrusion Avoidance in IoT-Based WSN. *Energies*. 2019;12(21). Available from: https://dx.doi.org/10.3390/en12214174.

7) Saba T, Haseeb K, Din IU, Almogren A, Altameem A, Fati SM. EGCIR: Energy-Aware Graph Clustering and Intelligent Routing Using Supervised System in Wireless Sensor Networks. *Energies*. 2020;13(16). Available from: https://dx.doi.org/10.3390/en13164072.

8) Abella CS, Bonina S, Cucuccio A, D'Angelo S, Giustolisi G, Grasso AD, et al. Autonomous Energy-Efficient Wireless Sensor Network Platform for Home/Office Automation. *IEEE Sensors Journal*. 2019;19(9):3501–3512. Available from: https://dx.doi.org/10.1109/jsen.2019.2892604.

9) BenSaleh MS, Saida R, Kacem YH, Abid M. Wireless Sensor Network Design Methodologies: A Survey. *Journal of Sensors*. 2020;2020:1–13. Available from: https://dx.doi.org/10.1155/2020/9592836.

10) Haseeb K, Din IU, Almogren A. sensors An Energy Efficient and Secure IoT-Based WSN Framework : An Application to Smart Agriculture. *Sensors*. 2020;20:2081–2095. Available from: http://doi.org/10.3390/s20072081.

11) Li X, Zhu L, Chu X, Fu H. Edge Computing-Enabled Wireless Sensor Networks for Multiple Data Collection Tasks in Smart Agriculture. *Journal of Sensors*. 2020;2020:1–9. Available from: http://doi.org/10.1155/2020/4398061.

12) Ullah U, Khan A, Zareei M, Ali I, Khattak HA, Din IU. Energy-Effective Cooperative and Reliable Delivery Routing Protocols for Underwater Wireless Sensor Networks. *Energies*. 2019;12(13):2630–2630. Available from: https://dx.doi.org/10.3390/en12132630.

13) Khan R, Mian AN. Sustainable IoT Sensing Applications Development through GraphQL-Based Abstraction Layer. *Electronics*. 2020;9(4):564–564. Available from: https://dx.doi.org/10.3390/electronics9040564.

14) Rani S. Dynamic Clustering Approach Based on Wireless Sensor Networks Genetic Algorithm for IoT Applications. *Wireless Networks*. 2019;7:2307–2316. Available from: http://doi.org/10.1007/s11276-019-02083-7.

15) Kocakulak M, Butun I. An overview of Wireless Sensor Networks towards internet of things. In: and others, editor. 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC). ;p. 1–6. Available from: http://doi.org/10.1109/CCWC.2017.7868374.

16) Goyal D, Tripathy MR. Routing Protocols in Wireless Sensor Networks: A Survey. In: and others, editor. 2012 Second International Conference on Advanced Computing & Communication Technologies. 2012;p. 474–480. Available from: http://doi.org/10.1109/ACCT.2012.98.

17) Kopjak J, Sebestyen G. Comparison of data collecting methods in wireless mesh sensor networks. In: and others, editor. IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI). IEEE. 2018;p. 155–60. Available from: http://doi.org/10.1109/SAMI.2018.8324005.

18) Iqrf os. 2013. . Available from: https://static.iqrf.org/User_Guide_IQRF-OS-403D_TR-7xD_191209.pdf.

19) Bali MMEI. Perguruan Tinggi Islam Berbasis Pondok Pesantren. *AL-TANZIM : JURNAL MANAJEMEN PENDIDIKAN ISLAM*. 2017;1:1–14. Available from: https://dx.doi.org/10.33650/al-tanzim.v1i2.109.

20) Tech I. IQRF HW design. 2020. Available from: https://static.iqrf.org/Appl_Note_AN015-IQRF-HW-Design_200526.pdf.

21) Seflova P, Sulc V, Pos J, Spinar R. IQRF wireless technology utilizing IQMESH protocol. *35th International Conference on Telecommunications and Signal Processing*. 2012;p. 101–105. Available from: http://doi.org/10.1109/TSP.2012.6256261.

22) Set Z. The mediating role of narcissism, vulnerable narcissism and self-compassion in the relationship between attachment dimensions and psychopathology. *Anatolian Journal of Psychiatry*. 2020;21. Available from: https://dx.doi.org/10.5455/apd.99551.

23) Guide U. DK-EVAL-04 IQRF development kit DK-EVAL-04. 2014. Available from: https://static.iqrf.org/User_Guide_DK-EVAL-04_140801.pdf.

24) Guide U. CK-USB-04A, IQRF Programmer and Debugger. 2017. Available from: http://static.iqrf.org/User_Guide_CK-USB-04A_171109.pdf.

25) Solutions SC. IQ Home Gateway. 2018. Available from: https://www.iqrfalliance.org/data_files/news/d2-07-iq-home-jozsef-kopjak.pdf.

26) Skovranek J, Pies M, Hajovsky R. Use of the IQRF and Node-RED technology for control and visualization in an IQMESH network. *IFAC-PapersOnLine*. 2018;51:295–300. Available from: https://dx.doi.org/10.1016/j.ifacol.2018.07.169.

27) IQHome Kft. IQ Home Sensor Protocol. 2017. Available from: https://www.iqhome.org/index.php?route=extension/d_blog_module/post&post_id=116.

28) Franco IB, Derbyshire E, Science TS. Actioning the Global Goals for Local Impact. Franco IB, Chatterji T, Derbyshire E, Tracey J, editors;Singapore. Science for Sustainable Societies. . Available from: http://doi.org/10.1007/978-981-32-9927-6_10.