Internet of Things 17 (2022) 100488

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Tutorial

A mesh network case study for digital audio signal processing in Smart Farm

João Paulo Lemos Escola [a,b,*], Uender Barbosa de Souza [c,d], Rodrigo Capobianco Guido [e], Ivan Nunes da Silva [b], Jovander da Silva Freitas [a], Lucas de Araújo Oliveira [a]

[a] *Instituto Federal de São Paulo, Av. C-1, 250, 14781-502 Barretos, SP, Brazil*
[b] *Universidade de São Paulo, Av. Trabalhador São-carlense, 400, 13566-590 São Carlos, SP, Brazil*
[c] *Instituto Federal de Goiás, Rua 75, 46, 74055-110 Goiânia, GO, Brazil*
[d] *Universidade Federal de Goiás, Av. Universitária, 1488, 74605-010 Goiânia, GO, Brazil*
[e] *Instituto de Biociências, Letras e Ciências Exatas, Unesp – Univ Estadual Paulista (São Paulo State University), Rua Cristóvão Colombo, 2265, 15054-000 São José do Rio Preto, SP, Brazil*

ARTICLE INFO

Keywords:
Internet of Things
Digital audio signal processing
Smart Farm
Mesh network
WSN

ABSTRACT

The Internet of Things (IoT) is increasingly present in people's daily lives and in many projects involving Smart Farm monitoring. Digital processing of audio signals enables detection and monitoring of species that emit sounds in crop fields. This paper aims to show a case study of a mesh network format cicada monitoring system in coffee plantations. The system, manages the sending, receiving, controlling and caching of data traveling between nodes deployed in the field. Laboratory tests have shown promising results for the intended application.

1. Introduction

The Internet of Things (IoT) is one of the most promising topics for industry and research in the field of communications engineering today [1]. With the growth of the semiconductor industry and the availability of wireless communication, this technology is increasingly available to consumers, allowing engineers to implement ever more daring projects such as Smart Cities and Smart Farms enabled to use mini unmanned aerial vehicles (UAVs) or drones intended for monitoring and security [2,3].

The mesh network topology allows data exchange between all connected devices, named "nodes", favoring the inclusion of new components in its coverage area for possible expansions without additional configurations, providing greater coverage without the need for a central device with large range has to cover all nodes, such as networks with star topology. The concept of wireless mesh networks has been discussed for decades [4], but it has not yet been implemented on a considerable scale. This scalability can be achieved if deployed in conjunction with available IoT technology [5], and thus the most attractive alternative to traditional centralized or tree-based network topologies, however, in the case of mesh networking, the research community and industry are currently seeking to develop common standards and comprehensive research into the options available for effective deployment [6].

Digital signal processing (DSP) enables a wide range of digital applications, including audio processing [7,8], which consists of performing computational calculations on acoustic data coming from various sources, such as a digital file or a microphone. There are many tools used in science for audio DSP, such as the Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT). In addition to these, the Discrete Wavelet Transform (DWT) is one of the most efficient means of time–frequency analysis in an audio signal. In addition to enabling filtering of audio signals, DWT reveals the time support of each frequency in the signal efficiently [9].

* Corresponding author at: Instituto Federal de São Paulo, Av. C-1, 250, 14781-502 Barretos, SP, Brazil.
E-mail address: jpescola@ifsp.edu.br (J.P.L. Escola).

https://doi.org/10.1016/j.iot.2021.100488
Received 17 July 2021; Received in revised form 3 December 2021; Accepted 9 December 2021
Available online 4 January 2022
2542-6605/

The concepts of Smart Farm, e-farm, digital agriculture, among other derivations, refer to the use of smart devices in agricultural work, usually employing IoT. This is a very promising area, with several studies recently published [10–14].

In a recently published paper [15], we suggest developing a Smart Farm system to detect and help manage cicada species *Quesada gigas* by capturing and processing the audio signal emitted by the male. This species is considered a key pest in coffee plantations in the state of Minas Gerais and the northeast region of the state of São Paulo, Brazil, a common problem in monocultures as a result of the decrease in biodiversity in that region and needs to be resolved so that the productivity of the plantations is not harmed, and the management is most often done by chemical pesticides that can impact the environment and human health. Through digital audio signal processing, it is possible to automate cultivation processes that involve characteristic audio signals, such as species and pest insect monitoring and classification [16–18].

A mesh network, one of the most widely used topologies in WSNs (Wireless Sensor Networks) today, is the ideal solution for an IoT project like the one mentioned, because in cases of unwanted events, such as natural disasters, its robustness guarantees the continuity of communication, even if part of the network is destroyed. Data travels from one node to another according to traffic availability, ensuring system continuity until faulty nodes are fixed [1,19].

One of the main disadvantages in implementing a mesh network is the need to implement a communication protocol specific to the application for which the network is intended [20]. In this sense, the objective of this paper is to present an IoT communication algorithm for mesh networks developed for systems that implement digital audio signal processing, as described.

The paper is organized as follows: in Section 2, works related to the proposed theme are presented; Section 3 is focused no the methodology used, followed by the analysis of results and conclusions with proposals for future work.

2. Related works

A set of technologies and devices to act in the rural environment was presented by Zheleva et al. [21]. After exposing a survey on multiple technologies used in the field to increase smallholders productivity, the authors propose a system to provide integrated monitoring and control of farming. Details about the technologies employed in the communication of the modules, integrated sensors, and processing the collected data are discussed in detail in the paper.

Pascale et al. [5] presented a software framework to integrate IoT mesh networks with artificial neural networks. In this work, the mesh network was transformed into an artificial neural network, where each IoT module was considered a neuron in the network. The main objective was to test the hypothesis of processing data directly in the network, even before sending it to the cloud, and mitigating the power hole effect by optimizing power consumption between the network nodes and increasing its lifetime.

Kodali, Yerroju and Sahu [10] developed a monitoring system in Smart Farm using LoRa (Long Range) devices [22]. When considering the use of mesh topology, the authors preferred the star topology, claiming that the ZigBee protocol in mesh network, "fails terribly" over long distance communications (more than 1 km), despite considerable performance over medium and short distances.

Another example of practical application of IoT in agriculture was presented by Keerthana, Karpagavalli and Posonia [23], who developed a system, composed of sensors, such as temperature, humidity, water level and soil moisture, integrated with a Raspberry Pi mini-PC and a smartphone app, in order to collect data from sensors scattered around the field. In this case, however, each device needs to have individual Internet access to communicate.

A low-cost IoT-based ecological monitoring system using audio sensor, microphone, battery and a Zigbee-based wireless mesh network was presented by Sheng et al. [24]. The focus of the paper, since it deals with an offline battery-powered system, was on saving energy by processing audio signals employing FFT and sending the features to a server using mesh network in conjunction with the Edge Computing framework. During the tests, the nodes were positioned at 25 meters and 110 meters in different range tests and power saving analysis.

In the paper by Ciani et al. [19], a low-cost mesh network IoT implementation for environmental monitoring in farming was presented. The prototype includes sensors for air temperature, air humidity, soil temperature, soil moisture, and solar radiation. The tests performed were focused on analyzing the behavior of the sensors through customized temperature step stress tests, based on various international standards for electronic devices. The arrangement of nodes in a mesh network was done using the WiFi protocol, and no tests of network range are described.

A prototype of a hybrid mesh network for IoT devices was developed by Jiang et al. [25], who used LoRa technologies combined with the ANT messaging protocol [26]. To this end, the authors have developed a communication protocol based on the TDMA (Time Division Multiple Access) scheduling algorithm. Tests with 20 devices have proven improvements in stability, node power consumption, and network reach.

Alves et al. [11] explores the issue of using bluetooth interfaces in mesh networks to monitor crops via IoT devices. In this case, the proprietary CSRmesh protocol, developed by the company Qualcomm, was used for communication between the nodes. The tests were carried out in a 1 hectare area, with nine sensors separated every 30 m.

Vijay et al. [27] also presented an IoT solution that employs mesh networks, where the focus was on monitoring cargo terminals at airports. Employing a junction of Bluetooth Low Energy (BLE) and Power Line Communications (PLC), the authors have developed a system that enables intelligent air cargo monitoring and continuous tracking, helping to prevent loss and theft.

In their work, Alonso et al. [12] implemented a Smart Farm to monitor a farm using Edge Computing, however, the communication of devices deployed in the crop was done directly with the edge through long range modules deploying LoRa.

Lee and Ke [28] presented a case study of deploying a mesh network with LoRa modules on a university campus. Nineteen stations were distributed over an 800 × 600 meter area, collecting data at a 60-second interval and comparing the performance of the mesh and star network architectures.

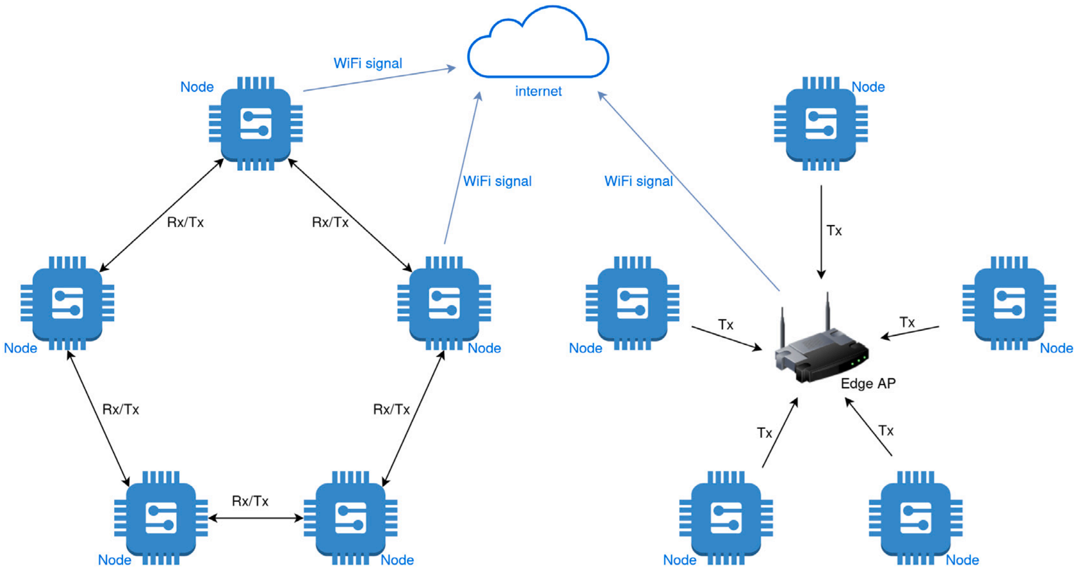

Fig. 1. Mesh network topology (left) and star topology (right).

Presenting a comprehensive survey that addresses various wireless communication technologies that support mesh networks, Cilfone et al. [6] focused on the implementation of mesh networks for IoT solutions. According to the authors, the heterogeneity of the applications (agriculture, industry or cities, for example) and the available devices makes the requirements specific for each implementation. In addition, currently developed technologies are also considered heterogeneous in terms of protocols, performance, cost, and coverage. Therefore, the interest of academia and industry today is to develop systems capable of supporting Wireless Mesh Networks (WMNs).

Jain et al. [29] reinforce the importance of mesh networks for a wide variety of IoT applications that seek long range and area coverage. In the paper, the authors present a parallelization solution, using the OpenMP standard, for processing the data received by the parent node in mesh networks for IoT applications, seeking to reduce the overall response time of the system.

A case study implementing the ESP-WIFI-MESH protocol has recently been presented by Waechi and Teerapabkajorndet [30], who have created a library that easily implements mesh networks on ESP32 devices using the WiFi interface [31]. Its main limitation is in the maximum nominal range of 200 meters without obstacles, not allowing the inclusion of external communication modules. The authors analyzed network performance using three ESP32 devices arranged in different topologies.

In most of the works obtained in literature searches, it is possible to find implementations of mesh networks using IoT, however, in some cases proprietary software is used and in others the application is not directed to the final objective proposed in this paper. Therefore, below will present the details of the algorithm developed and implemented for testing on the Arduino platform [32], using ESP32 modules, in addition to the practical tests performed in the laboratory.

3. Methodology

In this section the proposed system will be presented along with the details involved in the development the practical experiments of mesh network communication between the nodes, and then the tests to validate the proposal will be presented.

Applications like the ones mentioned involve large distances to be covered by the network, so its topology directly influences power consumption, communication quality, and scalability [33]. Two examples of network topology are illustrated in Fig. 1, mesh and star-shaped networks. In the first example, nodes communicate with each other, exchanging data through common communication interfaces (black arrows), promoting data exchange between nodes, aiming to reach the edge of the network, allowing access to the Internet (blue arrows) where the data is processed and stored permanently. In the second case, a concentrator is needed, represented in the image by a router, and only it has access to the Internet (blue arrow) and the nodes only have secondary interfaces (black arrows) for communication between the nodes.

As mentioned by Cilfone et al. [6], the organization of a WMN is usually done by defining a routing policy shared among all nodes, allowing them to discover and choose the best routes based on different metrics such as link quality, number of hops, and throughput for data to traverse the coverage area until they can reach the edge to be routed to their destination.

As exposed, one of the main disadvantages in implementing a mesh network is the need to implement a communication protocol specific to the application for which the network is intended. Thus, this paper presents a prototype algorithm for mesh networks using IoT for digital processing of audio signals, composed of routines such as capture (CP), receive (RX), cache (CH), send (TX) and others, detailed below and whose practical tests and results are presented in the next section.

Table 1
Data pack header.

B1	B2	B3	B4	B5	B6	B7	B8	B9
!10AE	@10AE	#1	$0	%0	”0	&0	*0	(0

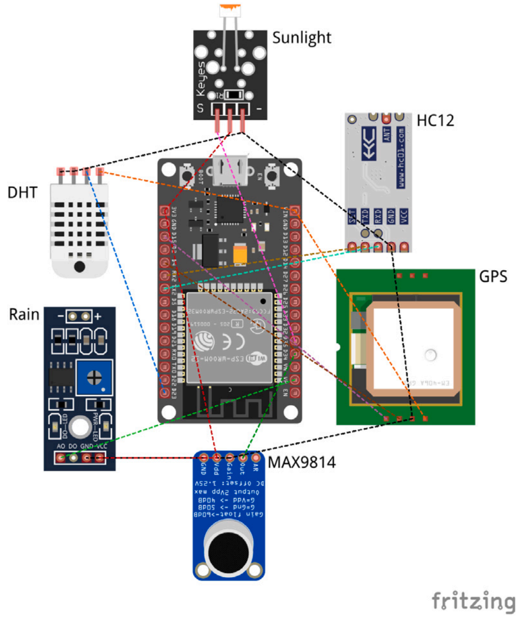

Fig. 2. IoT node prototype.

The Table 1 shows the header data encapsulated by the proposed algorithm. The header is divided into nine blocks whose values, in ASCII format, are delimited by special characters that mark the beginning of each block: Source node name (B1), Replicator node name (B2), Hops counter (B3), Temperature (B4), Humidity (B5), GPS Latitude (B6), GPS Longitude (B7), Raindrops level (B8), and Sunlight level (B9).

Blocks B1 and B2 store the name of the current node, obtained by a routine that reads the MAC address from the ESP32's built-in network interface. This routine can be adjusted by another routine at the developer's choice, generating a random named code, for example. As will be seen later, the B2 block can be replaced if data is relayed through another node before reaching the edge.

The hop counter, controlled in block B3, allows one to know how many nodes were traversed until data is sent to the edge. With this information it is possible for the administrator to measure the cost of hops needed in the network, and can choose to reposition the nodes in the network.

Blocks B4 to B9 were implemented from the reading of external sensors, coupled to ESP32 from its GPIO ports, as shown in Fig. 2, where the prototype of the proposed system is illustrated, with the ESP32 microcontroller (in the center) and the set of sensors that allow the acquisition of data to compose the header, except by the HC12 module that is responsible for communication and the MAX9814 module, being a sound signal acquisition module, which allows the capture of the audio signal for internal processing. These are optional blocks, and additional blocks can be implemented at the developer's discretion according to the needs of the intended application.

Also according to Liu et al. [1], communication failures in mesh networks can be painful, which can be mitigated by including network packet error reporting system, despite the possibility of slowing down the network. Thus, the proposed algorithm includes the error checking routine (ER), detailed below, which is responsible for filtering incoming data before it is cached, ensuring the integrity of packets traveling over the network.

In Fig. 3 the architecture of the proposed algorithm is illustrated, which consists of 9 steps. The sub-steps were described for reference purposes, and can be adapted according to the equipment used, besides allowing substitution of the DSP algorithm and the feature extraction technique.

- **Start**;
- **Step 1: Audio Signal Capture (DT).** The audio signal is captured and processed, obtaining the N energies of the discrete Wavelet transform [34]. This step can be replaced by any other type of data processing, such as discrete Fourier transform [7, 35] or the Hilbert–Huang transform [36,37], or the data acquisition process, in case this algorithm is implemented for other purposes;

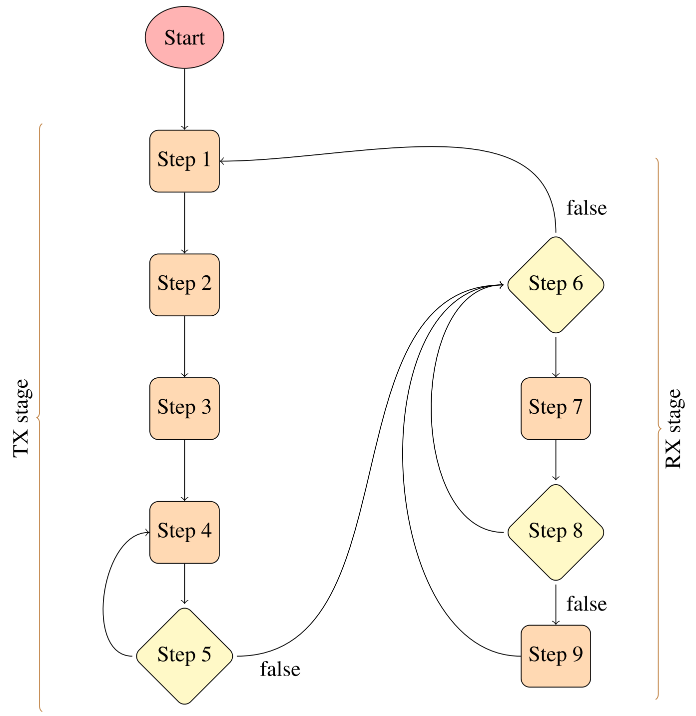

Fig. 3. Diagram of the proposed algorithm..

- **Step 1.1:** Start capturing the audio signal by the MAX9814 module, storing it in the temporary array;
- **Step 1.2:** Normalize the original signal and perform the Wavelet Packet transform on the signal, at maximum decomposition level, stored in the temporary array;
- **Step 1.3:** Convert the transformed signal into an array of subband frequency energies of the signal;
- **Step 1.4:** Generate a String S with the concatenation of each energy value with its respective representative letter, where the letter "A" represents the first feature;
- **Step 1.5:** Return the string of energies S.

• **Step 2: Header Assembly (HE).** The data captured by the other sensors are organized in the format described in Table 1, composing the first part of the data array that will be transmitted later. The header can be extended to allow inclusion of data from more sensors or shortened for simplification;

- **Step 2.1:** Get the microcontroller ID by reading the last 4 characters of the MAC network address;
- **Step 2.2:** Create the X header String with the device ID obtained in the previous step;
- **Step 2.3:** Take a Reading of the DHT sensor, obtaining the temperature and humidity, concatenating in X with their respective control characters;
- **Step 2.4:** Get the GPS positioning and concatenate on X with its control characters;
- **Step 2.5:** Read the raindrops and sunlight sensors, concatenating in X with their respective control characters;
- **Step 2.6:** Terminate the header assembly and return the String X as the result.

• **Step 3: Local Cache (LC).** The last result of the DT routine is stored in cache position 0. This guarantees that at least one of the data acquisitions to be transmitted belongs to the current node, but this does not guarantee receipt by the recipient;

- **Step 3.1:** Take the result of DT (Step 1) and HE (Step 2) as a parameter, concatenating the two data sets;
- **Step 3.2:** Store the resulting packet in cache memory position 0.

• **Step 4: Transmission (TX).** The cached packet stack, consisting of header and data features, is transmitted in queue. By default, at least one packet (obtained in Step 3) will be available in cache before transmission;

- **Step 4.1:** Transmit the current cache memory position packet via the HC12 communication module;
- **Step 4.2:** Move to the next cache memory position.

• **Step 5: Transmission check (TC).** Check if there is still data to be transmitted, returning to the previous step or going to the next step if all the data in the cache has been transmitted;

- **Step 5.1:** Check if there is any cached data left to be transmitted, returning to the previous step if so. Exit the routine if not.

• **Step 6: Receive Check (RC).** Check for packets to receive, going to the next step to start receiving, or restart the iteration set if there are not;

- **Step 6.1:** Verify the data to be received by checking the "available" function on the HC12 module, restart the process (Step 1) if not.

• **Step 7: Receiving (RX).** The data is received and stored in a temporary memory, starting at position 1, keeping position 0 in read-only mode (DT routine of the current node);

- **Step 7.1:** Having gone through the previous step, which means that there is data to be received, store the data in the temporary memory.

• **Step 8: Error Checking (ER).** The data in the temporary memory is checked and discarded in case of failure;

- **Step 8.1:** Check Package Header Integrity:

  * **Step 8.1.1:** Load the special characters and their respective positions in the header into memory to check their positions and content;
  * **Step 8.1.2:** Check each defined position in the special character array, verifying that the special character exists and that the contents are valid. Stop the process if it fails;
  * **Step 8.1.3:** Check for duplicate incidence of the already loaded character, which means that the signal is corrupted and halting the process.

- **Step 8.2:** Check Package Body Integrity:

  * **Step 8.2.1:** After checking the integrity of the header, verify the integrity of the package body by checking in all control characters and their contents.

• **Step 9: Secondary Cache (SC).** The packet resulting from the previous step is stored in cache memory;

- **Step 9.1:** Replace in the header the sender ID (B1), with the current device ID, while keeping the source device ID (B2);
- **Step 9.2:** The cache memory position is incremented. If the memory limit is reached, reset the position counter to 1, keeping position 0 to ensure at least one signal to the current device;
- **Step 9.3:** The data obtained from the temporary memory is stored in the available cache memory position;
- **Step 9.4:** Continue to Step 6 (RC routine).

The final step (End) has been omitted because it is a monitoring algorithm, which should run indefinitely, keeping the process of capturing, transmitting and receiving packets continuously as long as there is power supply, and it is also possible to include an intermittent hibernation routine in order to save energy and extend its operation time by fractioning it during the 24 h of the day.

In the next section, the results and discussions about the implementation of the mentioned algorithm and the tests performed in the laboratory will be presented.

4. Results

To perform the tests, the proposed algorithm was implemented under the Arduino platform using four ESP32 microcontrollers. The implementation was developed using a microcomputer with Intel(R) Core(TM) i7-4800MQ CPU @ 2.70 GHz, 32 GB RAM and 1TB SSHD hard disk in a Linux environment, using the IDE PlatformIO [38].

The HC12 transmitter module was used for data transmission between nodes. It is a half-duplex wireless transceiver operating in the frequency band between 413 and 473 MHz and a nominal range of 1 km [39,40]. The Fig. 2 contains the illustration of this module connected to the ESP32 microcontroller.

Three sets of tests were performed in the laboratory in two stages, totaling 140 tests in the first stage and 30 tests in the second stage. The details of each testing step are explained below. For the purposes of understanding the tests performed, the term "packet" represents each processed data string, containing header and body, as described earlier. Similarly, "healthy packets" is the term used to represent packets that have been delivered intact to their destination.

4.1. First stage of tests

During the first stage, the T1 and T2 batteries of tests were performed using two nodes, as shown in Fig. 4. In T1 the nodes were placed at a distance of 20 meters in an open area (red border), and in T2 the nodes were allocated at a distance of 30 meters in a non-open area (blue border), i.e., with the incidence of multiple obstacles between the nodes comprising the mesh network. The T3 battery of tests, on the gray border, is described later.

In each battery of tests in the current stage, 70 tests were performed, divided into 5 sets, varying the amount of features of the digital audio processing routine between 5 and 25, making a total of 14 tests in each set. These features constitute the body of the

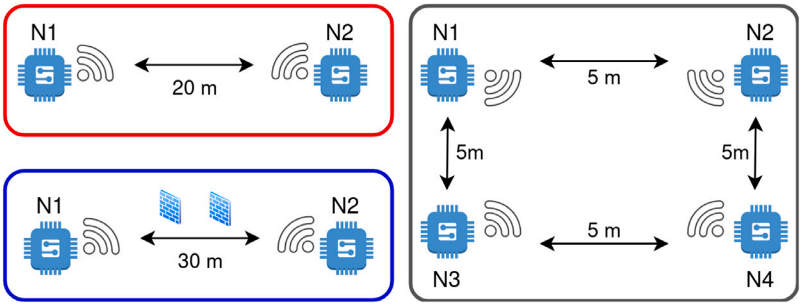

Fig. 4. Organization diagram of tests T1 (left above), T2 (left below) and T3 (right).

Table 2
First stage of tests.

Battery	Nodes	Distance (m)	Tests	Features	Packets/test	Min	Median	Max
T1	2	20	14/70	5	100	66	70	78
T1	2	20	14/70	10	100	28	51	73
T1	2	20	14/70	15	100	29	39	59
T1	2	20	14/70	20	100	38	48	53
T1	2	20	14/70	25	100	43	45	59
T2	2	30	14/70	5	100	37	55	87
T2	2	30	14/70	10	100	58	64	71
T2	2	30	14/70	15	100	38	53	66
T2	2	30	14/70	20	100	26	46	59
T2	2	30	14/70	25	100	31	41	49

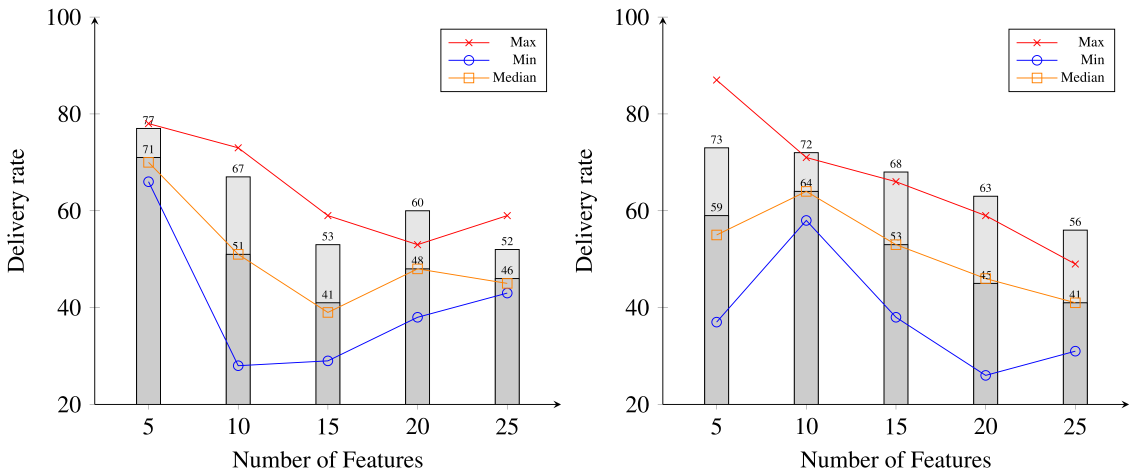

Fig. 5. Results in T1 (left) and T2 (right).

packet to be transmitted between network nodes. In each test, 100 packets were fired between the nodes that are part of the mesh network and the average delivery rate was measured. The test setup and its respective results are organized in Table 2.

The nodes in the network alternate between sending (TX) and receiving (RX) packets in an iteration limited to 100 transmissions. The transmission process is measured from the number of packets received by each node. Thus, at the end of the process, an average yield of 51.4% was obtained in T1 and 54.4% in T2. Fig. 5 presents the results obtained on test sets T1 and T2.

Looking at Fig. 5 in both T1 and T2, it is observed that the network performance is gradually reduced as the amount of features resulting from the DT routine is increased. The drop in performance in T1 with 15 features, followed by an increase in performance in the next test (20 features) may have occurred due to some type of electromagnetic interference, since the aforementioned experiments were performed in the laboratory. According to Fey and Gauer [41], radio signals are conducted through the air and can experience electromagnetic obstructions and interference.

Each test had an average duration of 23 min, so considering that each test sent 100 packets and received an average of 51% to 54%, it can be considered that the system provides an average response time of $\frac{23\ \text{min} * 60\ \text{s}}{54\ \text{packets}} = 25.55$ s/packet, which can be considered acceptable for monitoring systems like the proposed one, since it is not considered a real-time system.

Table 3
Second stage of tests.

Battery	Nodes	Distance (m)	Tests	Features	Packets/test	Min	Median	Max
T3	4	5	6/30	5	100	96	105	117
T3	4	5	6/30	10	100	77	96	115
T3	4	5	6/30	15	100	78	82	90
T3	4	5	6/30	20	100	60	68	79
T3	4	5	6/30	25	100	53	68	81

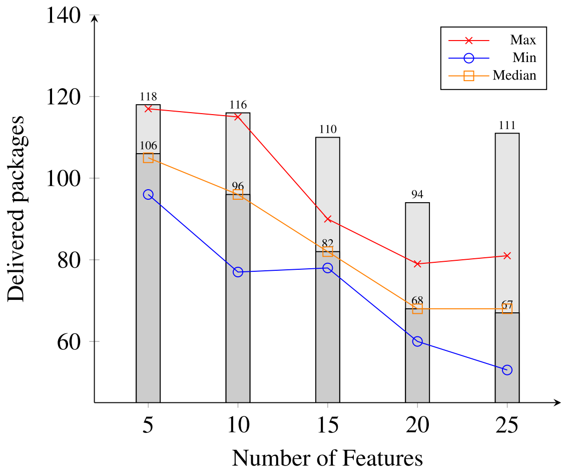

Fig. 6. T3 test results.

```
RX[96/85]:  !309D@C8BD#1$10%99¨&*89(84A16B16C16D16E16
RX[97/86]:  !58FD@C8BD#1$11%99¨&*0(82A17B17C17D17E17
RX[98/87]:  !C8BD@C8BD#1$10%99¨&*72(92A19B19C19D19E19
RX[99/88]:  !58FD@C8BD#1$12%99¨&*0(79A23B23C23D23E23
```

Fig. 7. Packets received during testing using 4 nodes.

4.2. Second stage of tests

In the previous step, in order to effectively control the sending/receiving of packages in the tests, only 2 nodes were used. As is known, in a real mesh network, this number can be much higher. Therefore, in order to analyze the feasibility of the algorithm to act as a mesh network, a T3 battery of tests was included in this work, composed of 4 simultaneously active nodes, simulating, in a minimal way, the expected environment in a field (gray border in Fig. 4).

At the current stage, 30 tests were performed, divided into 5 sets, varying the amount of features of the digital audio processing routine between 5 and 25, making a total of 6 tests in each set. These features constitute the body of the packet to be transmitted between network nodes. In each test, 100 packets were fired between the nodes that are part of the mesh network and the average delivery rate was measured. The configuration of the tests and their respective results are organized in Table 3.

The Fig. 6 shows the results of the aforementioned test, which was performed with nodes allocated in a mesh network at a distance of 5 m. The light bar represents the average amount of packets received and the dark bar shows the average amount of healthy packets, i.e., only the packets received that passed the error checking routine (ER). The minimum, maximum and median values of the total healthy received packets by the 4 nodes were included.

In some cases in T3, the number of packets delivered was more than 100, although the algorithm is limited to sending 100 packets in each test. This happened because for each receiver node present in the network, there are 3 possible senders, different from what happened in tests T1 and T2 where there is only one sender and one receiver.

Regarding the number of features, we can also see in Fig. 6 that increasing the number of features reduced the network performance even for a reduced distance, as used in the test mentioned above.

Finally, Fig. 7 illustrates an example of the monitoring report displayed by one of the nodes in T3. It can be seen in the example that C8BD is the current node that receives packets of 5 features (A to E), whose origins are the nodes "309D", "58FD" and "C8BD" (the same).

5. Conclusions

This paper presented a prototype mesh network algorithm for digital audio signal processing projects employing IoT devices in Smart Farm. The test results showed the feasibility of the technique with satisfactory results for the proposed application.

The present study aims to contribute to the state of the art of IoT in Smart Farm, presenting itself as an alternative to the use of proprietary mesh network protocols [11,24], being limited to the communication process, in contrast to more complex studies that extrapolate the data communication process [5,21,29]. Furthermore, the present case study employs low-cost sensors and microcontrollers, as opposed to using long-range modules [10,12,28] and high-cost mini-pcs [23], which may make the process unfeasible depending on the coverage area to be applied.

As exposed, WSN topologies must be built to ensure efficiency in the data transmission process. Despite focusing on the implementation of mesh and star topologies, referencing modern works available in the scientific literature, other less common topologies can be employed, such as the tree topology. Its use refers to the implementation of hierarchies between nodes, which would reduce the efficiency of the network in case of failures, i.e., once one of the nodes is compromised, all child nodes become inoperable.

The proposed system uses signaling characters to delimit the start/end of blocks to allow undefined block sizes, allowing variation in the size of the data contained in each block and saving data traffic.

The distances chosen in the laboratory tests (20 m in open area, 30 m in non-open area, and 5 m in the final tests) were chosen due to the space limitations in the laboratory. It is desirable to perform tests at greater distances in future work.

The proximity of the nodes in T3 could refer to a star topology, however, comparing blocks B1 and B2 one can confirm the occurrence of multiple hops, characteristic of mesh networks.

There is no significant variation in performance between the T1 and T2 batteries of tests, so it can be concluded that at the distances considered in the tests, the results were not directly influenced by the existence or not of obstacles or by the variation in the distance between nodes.

The results showed that increasing the number of features gradually reduces the performance of the network, so it is prudent to look for ways to reduce the need for DT routine features in order to balance the system, seeking the best possible performance in data exchange between network nodes, improving the overall performance of the system.

Considering the two stages of tests performed, it can be considered that the results are statistically significant, as they allow the network utilization to be verified considering the limitations of possible distances in the laboratory.

In future work, it is desirable to perform tests over longer distances, even employing sensors with longer range, trying to keep the low cost, in order to validate the efficiency of the mesh network protocol in real conditions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge the grants provided by the Brazilian agencies "National Council for Scientific and Technological Development (CNPq)" and "The State of São Paulo Research Foundation (FAPESP)", respectively through the processes 306808/2018-8 and 2019/04475-0, in support of this research.

References

[1] Y. Liu, K.-F. Tong, X. Qiu, Y. Liu, X. Ding, Wireless mesh networks in iot networks, in: 2017 International Workshop on Electromagnetics: Applications and Student Innovation Competition, IEEE, 2017, pp. 183–185, http://dx.doi.org/10.1109/iWEM.2017.7968828.

[2] C. Rametta, G. Schembra, Designing a softwarized network deployed on a fleet of drones for rural zone monitoring, Future Internet 9 (1) (2017) http://dx.doi.org/10.3390/fi9010008.

[3] F. Al-Turjman, M. Abujubbeh, A. Malekloo, L. Mostarda, Uavs assessment in software-defined iot networks: An overview, Comput. Commun. 150 (2020) 519–536, http://dx.doi.org/10.1016/j.comcom.2019.12.004.

[4] B. Doshi, C. Funka-Lea, P. Harshavardhana, J. Gong, R. Nagarajan, S. Ravikumar, S. Chen, Y. Wang, Integrated network design tools (indt): a suite of network design tools for current and next generation networking technologies, in: Proceedings Second IEEE Symposium on Computer and Communications, 1997, pp. 332–338, http://dx.doi.org/10.1109/ISCC.1997.616020.

[5] E. Di Pascale, I. Macaluso, A. Nag, M. Kelly, L. Doyle, The network as a computer: A framework for distributed computing over iot mesh networks, IEEE Internet Things J. 5 (3) (2018) 2107–2119, http://dx.doi.org/10.1109/JIOT.2018.2823978.

[6] A. Cilfone, L. Davoli, L. Belli, G. Ferrari, Wireless mesh networking: An iot-oriented perspective survey on relevant technologies, Future Internet 11 (4) (2019) 99, http://dx.doi.org/10.3390/fi11040099.

[7] S.S. Haykin, B. Van Veen, Signals and Systems, John Wiley & Sons, Hoboken, 2007.

[8] S. Gupta, A.T. Patil, M. Purohit, M. Parmar, M. Patel, H.A. Patil, R.C. Guido, Residual neural network precisely quantifies dysarthria severity-level based on short-duration speech segments, Neural Netw. 139 (2021) 105–117, http://dx.doi.org/10.1016/j.neunet.2021.02.008.

[9] R.C. Guido, P.S. Addison, J. Walker, Introducing wavelets and time-frequency analysis [introduction to the special issue], IEEE Eng. Med. Biol. Mag. 28 (5) (2009) 13, http://dx.doi.org/10.1109/MEMB.2009.934243.

[10] R.K. Kodali, S. Yerroju, S. Sahu, Smart farm monitoring using lora enabled iot, in: 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, 2018, pp. 391–394, http://dx.doi.org/10.1109/ICGCIoT.2018.8753086.

[11] L. Alves, E. Antunes, R. Ferreira, J.A. Nacif, A mesh sensor network based on bluetooh: Comparing topologies to crop monitoring, in: Anais Estendidos Do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais, SBC, 2019, pp. 125–130, http://dx.doi.org/10.5753/sbesc_estendido.2019.8647.
[12] R.S. Alonso, I. Sittón-Candanedo, O. García, J. Prieto, S. Rodríguez-González, An intelligent edge-iot platform for monitoring livestock and crops in a dairy farming scenario, Ad Hoc Netw. 98 (2020) 102047, http://dx.doi.org/10.1016/j.adhoc.2019.102047.
[13] A. Zgank, Bee swarm activity acoustic classification for an iot-based farm service, Sensors 20 (1) (2020) 21, http://dx.doi.org/10.3390/s20010021.
[14] M. Gupta, M. Abdelsalam, S. Khorsandroo, S. Mittal, Security and privacy in smart farming: Challenges and opportunities, IEEE Access 8 (2020) 34564–34584, http://dx.doi.org/10.1109/ACCESS.2020.2975142.
[15] J.P.L. Escola, R. Guido, I.N. da Silva, D.H.B. Maccagnan, A.d.M. Cardoso, U. Souza, Estado da arte no monitoramento acústico de cicadidae em lavouras de café, Rev. Macambira 5 (1) (2021) http://dx.doi.org/10.35642/rm.v5i1.562.
[16] S.H. Gage, W. Joo, E.P. Kasten, J. Fox, S. Biswas, Acoustic observations in agricultural landscapes, Ecol. Agric. Landsc.: Long-Term Res. Path Sustain. (2015) 360–377.
[17] J.P.L. Escola, R.C. Guido, I.N. da Silva, A.M. Cardoso, D.H.B. Maccagnan, A.K. Dezotti, Automated acoustic detection of a cicadid pest in coffee plantations, Comput. Electron. Agric. 169 (2020) 105215, http://dx.doi.org/10.1016/j.compag.2020.105215.
[18] A. Nugroho, S. Purba, Y. Pratomo, S. Hadi, S. Utami, et al., Development of cloud-based bioacoustics monitoring system for supporting integrated pest management in agriculture production, in: IOP Conference Series: Earth and Environmental Science, Vol. 449, IOP Publishing, 2020, 012032, http://dx.doi.org/10.1088/1755-1315/449/1/012032.
[19] L. Ciani, M. Catelani, A. Bartolini, G. Guidi, G. Patrizi, Influence of raised ambient temperature on a sensor node using step-stress test, IEEE Trans. Instrum. Meas. 69 (12) (2020) 9549–9556, http://dx.doi.org/10.1109/TIM.2020.3006600.
[20] J. Crichigno, M.-Y. Wu, W. Shu, Protocols and architectures for channel assignment in wireless mesh networks, Ad Hoc Netw. 6 (7) (2008) 1051–1077, http://dx.doi.org/10.1016/j.adhoc.2007.10.002.
[21] M. Zheleva, P. Bogdanov, D.-S. Zois, W. Xiong, R. Chandra, M. Kimball, Smallholder agriculture in the information age: Limits and opportunities, in: Proceedings of the 2017 Workshop on Computing Within Limits, 2017, pp. 59–70, http://dx.doi.org/10.1145/3080556.3080563.
[22] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, T. Watteyne, Understanding the limits of lorawan, IEEE Commun. Mag. 55 (9) (2017) 34–40, http://dx.doi.org/10.1109/MCOM.2017.1600613.
[23] K. Keerthana, S. Karpagavalli, A.M. Posonia, Smart system monitoring agricultural land using iot, in: 2018 International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR), IEEE, 2018, pp. 1–7, http://dx.doi.org/10.1109/ICETIETR.2018.8529037.
[24] Z. Sheng, S. Pfersich, A. Eldridge, J. Zhou, D. Tian, V.C. Leung, Wireless acoustic sensor networks and edge computing for rapid acoustic monitoring, IEEE/CAA J. Autom. Sin. 6 (1) (2019) 64–74, http://dx.doi.org/10.1109/JAS.2019.1911324.
[25] X. Jiang, H. Zhang, E.A.B. Yi, N. Raghunathan, C. Mousoulis, S. Chaterji, D. Peroulis, A. Shakouri, S. Bagchi, Hybrid low-power wide-area mesh network for iot applications, IEEE Internet Things J. 8 (2) (2020) 901–915, http://dx.doi.org/10.1109/JIOT.2020.3009228.
[26] C. Harrison, Ant message protocol and usage, Dynastream Innov. Inc 5 (1) (2014) 1–134.
[27] R. Vijay, T. Prabhakar, V. Hegde, V.S. Rao, R.V. Prasad, A heterogeneous plc with ble mesh network for reliable and real-time smart cargo monitoring, in: 2019 IEEE International Symposium on Power Line Communications and Its Applications (ISPLC), IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ISPLC.2019.8693413.
[28] H.-C. Lee, K.-H. Ke, Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: Design and evaluation, IEEE Trans. Instrum. Meas. 67 (9) (2018) 2177–2187, http://dx.doi.org/10.1109/TIM.2018.2814082.
[29] M. Jain, R. Saxena, S. Jaidka, M.K. Jhamb, Parallelization of data buffering and processing mechanism in mesh wireless sensor network for iot applications, in: Smart Computing Paradigms: New Progresses and Challenges, Springer, 2020, pp. 3–12, http://dx.doi.org/10.1007/978-981-13-9680-9_1.
[30] M. Waechi, W. Teerapabkajorndet, Prototype development and performance evaluation of esp-wifi-mesh sensor networks for IoT, in: Asia Pacific Conference on Robot IoT System Development and Platform 2020, 2020.
[31] Espressif, Esp-wifi-mesh scale up iot with wi-fi mesh network, 2021, URL https://www.espressif.com/en/products/sdks/esp-wifi-mesh/overview.
[32] M. Banzi, M. Shiloh, Getting Started with Arduino: The Open Source Electronics Prototyping Platform, Maker Media, Inc., Sebastopol, 2014.
[33] P. Santi, Topology Control in Wireless Ad Hoc and Sensor Networks, Vol. 37, Association for Computing Machinery, New York, NY, USA, 2005, http://dx.doi.org/10.1145/1089733.1089736.
[34] R.C. Guido, Effectively interpreting discrete wavelet transformed signals [lecture notes], IEEE Signal Process. Mag. 34 (3) (2017) 89–100, http://dx.doi.org/10.1109/MSP.2017.2672759.
[35] A.V. Oppenheim, Signals & Systems, Pearson Education, Londres, 2010.
[36] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.-C. Yen, C.C. Tung, H.H. Liu, The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 454 (1971) (1998) 903–995, http://dx.doi.org/10.1098/rspa.1998.0193.
[37] U.B. de Souza, J.P.L. Escola, L. da Cunha Brito, A survey on hilbert-huang transform: Evolution, challenges and solutions, Digit. Signal Process. 120 (2022) 103292, http://dx.doi.org/10.1016/j.dsp.2021.103292.
[38] PlatformIO, Professional collaborative platform for embedded development, 2021, https://platformio.org/ (Accessed: 2021-03-22).
[39] J. Joseph, K. Manju, M. Sajith, S. Nair, V.P. Viay, S. Krishnan, Water management system using iot, Int. Res. J. Eng. Technol. 5 (04) (2018).
[40] Elecrow, Hc-12 wireless serial port communication module, 2021, https://www.elecrow.com/download/HC-12.pdf (Accessed: 2021-07-07).
[41] A.F. Fey, R.R. Gauer, Solucionando Problemas de Comunicação em Redes Lan e Wan, Clube de Autores (managed), Joinville, 2015.