Desafios na Utilização de Redes em Malha Sem Fios para a Deteção de Aves no Contexto da Agricultura Inteligente

Fábio Cardoso
Polytechnic University of Castelo Branco, Av. Pedro Álvares Cabral, nº 12, 6000-084 Castelo Branco, Portugal
c.fabio@ipcbcampus.pt

Daniel Carvalho
Polytechnic University of Castelo Branco, Av. Pedro Álvares Cabral, nº 12, 6000-084 Castelo Branco, Portugal
daniel.carvalho@ipcbcampus.pt

Pedro D. Gaspar
Department of Electromechanical Engineering, University of Beira Interior; Centre for Mechanical and Aerospace Science and Technologies (C-MAST), Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal
dinis@ubi.pt

Vasco N. G. J. Soares
Polytechnic University of Castelo Branco, Av. Pedro Álvares Cabral, nº 12, 6000-084 Castelo Branco, Portugal
Instituto de Telecomunicações, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal
vasco.g.soares@ipcb.pt

João M. L. P. Caldeira
Polytechnic University of Castelo Branco, Av. Pedro Álvares Cabral, nº 12, 6000-084 Castelo Branco, Portugal
Instituto de Telecomunicações, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal
jcaldeira@ipcb.pt

RESUMO

Os ataques das aves continuam a representar um dos maiores desafios enfrentados pelos agricultores, com implicação direta sobre a rentabilidade e sustentabilidade das explorações agrícolas. A aplicação de métodos tradicionais de dispersão de aves é ineficiente. A agricultura inteligente, por meio de tecnologias tais como Internet das Coisas, big data, inteligência artifical, entre outras, pode ser solução para este desafio. Contudo, um dos principais obstáculos nesse contexto é a disponibilidade de uma rede de comunicação de dados, devido à possível localização remota das explorações e à extensa cobertura requerida para monitorizar as áreas agrícolas. Um recurso para superar essas dificuldades é a utilização de redes em malha sem fios, que permitem a comunicação entre sensores distribuídos. Essas redes podem ser projetadas para funcionar de forma robusta em ambientes rurais, oferecendo a flexibilidade necessária para se adaptar às particularidades do terreno. No entanto, estas redes também apresentam desafios próprios, tais como escalabilidade, congestionamento e latência, que requerem uma análise atenta. Pretende-se com este trabalho analisar, propor e avaliar soluções que otimizem o desempenho destas redes, garantindo o envio bem-sucedido de mensagens necessárias num contexto de uma solução inovadora para dispersão de aves.

Palavras-Chave

Dispersão de Aves; Proteção de Colheitas; Agricultura Inteligente; Internet das Coisas; Redes em Malha Sem Fios; Sensores; Otimização de Rede; Avaliação de Desempenho.

ABSTRACT

Bird attacks continue to be one of the biggest challenges facing farmers, with direct implications for the profitability and sustainability of agricultural holdings. Traditional methods of bird dispersal are inefficient. Smart agriculture, through technologies such as the Internet of Things, big data, artificial intelligence, among others, could be the solution to this challenge. However, one of the main obstacles in this context is the availability of a data communication network, due to the possible remote location of farms and the extensive coverage required to monitor agricultural areas. One resource for overcoming these difficulties is the use of wireless mesh networks, which allow communication between distributed sensors. These networks can be designed to work robustly in rural environments, offering the necessary flexibility to adapt to the particularities of the terrain. However, these networks also present their own challenges, such as scalability, congestion and latency, which require careful analysis. The aim of this work is to analyse, propose and evaluate solutions that optimise the performance of these networks, guaranteeing the successful delivery of the necessary messages in the context of an innovative solution for bird dispersal.

Keywords

Bird dispersal; Crop Protection; Smart Agriculture; Internet of Things; Wireless Mesh Network; Sensors; Network Optimization; Performance Evaluation.

1. INTRODUÇÃO

A dispersão de aves nas colheitas é fundamental para a rentabilidade e sustentabilidade das explorações agrícolas. Os métodos tradicionais utilizados para afastar aves são pouco eficazes [1], levando à necessidade de soluções inovadoras e tecnologicamente avançadas. Nos últimos anos, têm surgido abordagens que incorporam tecnologias de monitorização e comunicação, prometendo melhorar a eficácia na gestão das populações de pássaros que atacam as culturas [2], [3], [4], [5]. Contudo, a implementação destas tecnologias em ambientes rurais apresenta desafios consideráveis. Em zonas remotas, onde a conectividade é limitada, a adoção da agricultura inteligente, que envolve o uso de sensores e dispositivos conectados para monitorizar o solo, as culturas e o ambiente, exige uma infraestrutura de rede de comunicação de dados que assegure a transmissão eficaz de dados e de preferência em tempo real. No entanto, as características geográficas e a extensão das áreas agrícolas, dificultam a utilização de tecnologias de redes convencionais, tornando a comunicação entre dispositivos um dos principais obstáculos à sua implementação.

O trabalho apresentado neste artigo, é uma das etapas que tem por objetivo final desenvolver a solução tecnológica ilustrada na Figura 1. Nas árvores de um pomar de grande dimensão e remoto, são detetados sons com recurso a sensores de aquisição de dados. O áudio recolhido, bem como a informação geográfica, é processado localmente ou encaminhado para um nó central que executa essa tarefa. Se for detetada a presença de aves nesse áudio, é notificado um *drone* que realiza, em modo autónomo, um

voo até ao local, onde efetua um conjunto de manobras para dispersão das aves. Este trabalho parte de um estudo prévio dos mesmos autores [6] que avaliou soluções para a deteção da presença e classificação das espécies de aves com base na aquisição de sons.

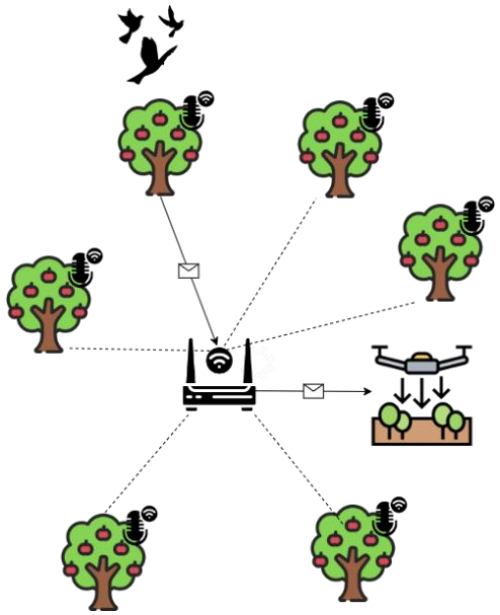

Figura 1. Cenário de estudo.

No cenário descrito, há uma dificuldade no que diz respeito à implementação de uma rede de comunicação de dados, que é necessária para envio da informação recolhida pelos sensores existentes nas árvores, para um nó central. Para ultrapassar a mesma, neste artigo considera-se o uso de uma rede em malha sem fios (do inglês, Wireless Mesh Network (WMN)) [7]. As WMNs são caracterizadas pela sua capacidade de auto-organização e distribuição dinâmica de dados entre vários nós. Estas redes permitem a criação de uma infraestrutura descentralizada, onde cada nó da rede pode atuar como um ponto de retransmissão, garantindo que os dados percorrem distâncias longas.

Apesar do potencial das WMNs para solucionar este problema, estas redes enfrentam desafios tais como: o congestionamento causado pelo tráfego de dados entre os nós, e a latência que afeta o tempo de resposta da rede. Assim, é objetivo geral do trabalho apresentado neste artigo analisar e avaliar o desempenho de métodos e protocolos que contribuam para aumentar a eficácia e a eficiência da WMN no cenário descrito. Pretende-se garantir não só a entrega dos dados, mas também tempo de envio reduzido.

Este artigo está organizado da seguinte forma. A secção 2 descreve o conceito de rede WMN e discute a sua aplicação no cenário considerado. A secção 3 descreve duas abordagens para a implementação da rede WMN, incluindo a configuração, a integração e os testes preliminares. A secção 4 é dedicada aos testes, validação e avaliação de desempenho. Finalmente, a secção 5 sintetiza as conclusões deste estudo.

2. REDES EM MALHA SEM FIOS

Uma rede em malha sem fios é uma topologia de rede na qual cada dispositivo, ou nó, atua como um ponto de retransmissão para os dados, criando uma estrutura de comunicação interconectada [7]. Ao invés de depender de um único ponto central, cada nó na rede comunica diretamente com outros nós, formando uma rede descentralizada [8]. Nestas redes, os nós podem comunicar diretamente entre si ou através de intermediários, o que resulta numa comunicação auto-organizada e com capacidade de se adaptar [9]. Quando um nó falha ou se torna inativo, os dados podem ser redirecionados por outros caminhos disponíveis na malha, garantindo a continuidade da transmissão e minimizando a interrupção do serviço [9]. Além disso, a rede é projetada para se ajustar dinamicamente a mudanças na topologia da rede, como a adição ou remoção de nós [10]. Esta abordagem descentralizada é particularmente benéfica em ambientes extensos e de difícil acesso, onde a cobertura e a robustez da rede são essenciais para garantir a transmissão eficaz de dados [10]. A capacidade de criar múltiplos caminhos de comunicação contribui para uma maior fiabilidade e resiliência da rede, proporcionando uma solução eficaz para a transmissão de dados em tempo real [9].

2.1 Problemas

Tendo em consideração o cenário de aplicação desta solução tecnológica (um pomar remoto de grande dimensão), um dos principais desafios é assegurar que todas as mensagens enviadas cheguem ao destino pretendido. É por isso preciso evitar a perda de mensagens que pode ocorrer devido a falhas nos nós da rede, interferências ambientais ou problemas de transmissão [11]. Para minimizar esse risco, é crucial adotar mecanismos de retransmissão e protocolos de confirmação de receção das mensagens [12]. Estes métodos garantem que cada mensagem seja, não apenas enviada, mas também recebida e reconhecida pelo nó central, assegurando a integridade da comunicação [13].

Outro problema é a receção tardia das mensagens. Isto é, as mensagens chegam ao nó central fora de um intervalo temporal aceitável. Um tempo de envio longo compromete a eficácia do sistema em tempo real. Para enfrentar este desafio, é necessário otimizar os protocolos de encaminhamento [14] e adotar técnicas de priorização de tráfego de maneira a garantir que as mensagens chegam num intervalo de tempo aceitável [15].

A ocorrência de falhas na transmissão de mensagens é um desafio significativo, que pode surgir devido a erros ou interferências no processo de comunicação [16]. Esses problemas podem comprometer a integridade dos dados, tornando fundamental a implementação de soluções que consigam identificar e corrigir tais incidentes [16]. Para garantir a precisão das informações transmitidas, é fundamental utilizar métodos de verificação, como somas de verificação e códigos de correção de erros [17], que asseguram que qualquer mensagem afetada seja detetada e reparada antes de ser processada, evitando assim que sejam processados dados corrompidos [17].

3. IMPLEMENTAÇÃO

Nesta secção, descrevem-se duas abordagens distintas para a implementação da rede WMN, incluindo a configuração, a integração e os testes preliminares, com o objetivo de identificar a solução que melhor atende às necessidades do cenário considerado neste estudo.

3.1 Aquisição de Áudio

Neste estudo, o foco principal é a otimização do desempenho da rede WMN, que será utilizada para propagar um áudio capturado por sensores, que é depois analisado, para deteção e identificação da espécie de aves correspondente. Assim, optou-se, por usar um áudio pré-gravado. Entre as espécies mais conhecidas por atacar

plantações agrícolas, visíveis na Figura 2, optou-se pela *Pica pica*, popularmente chamada de pega, por ser muito comum na região de Castelo Branco, Portugal, como descrito no website *BioDiversityForAll* [18], apresentado na Figura 3.

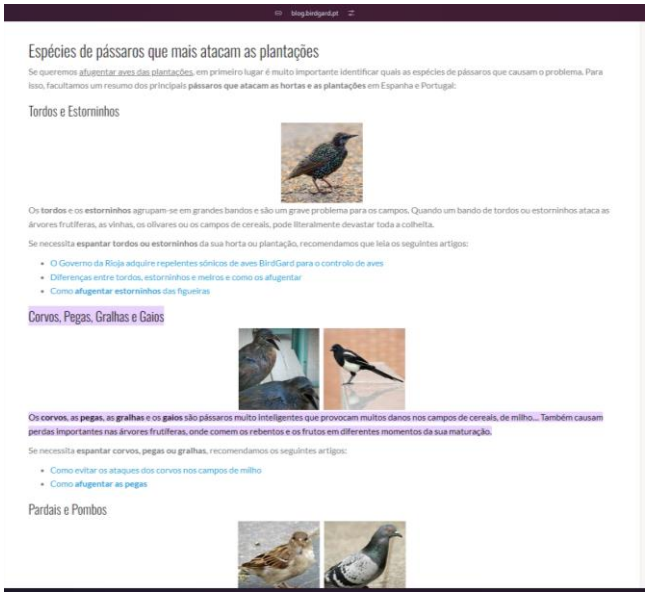

Figura 2. Espécies de pássaros que mais atacam as plantações. Fonte: [19]

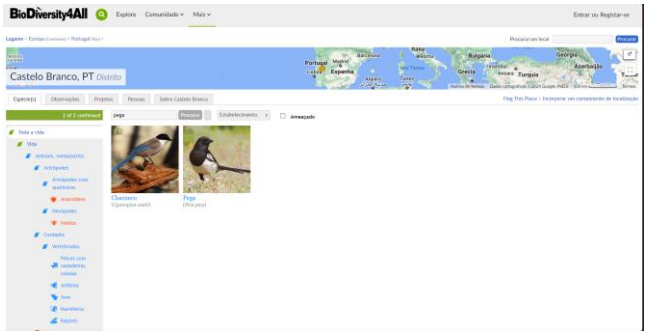

Figura 3. Presença da espécie Pega no distrito de Castelo Branco. Fonte: [18]

Utilizando a base de dados do *eBird* [20], que é a fonte de dados utilizada pelo software *BirdNET* [21], realizámos uma pesquisa sobre na espécie *Pica pica*, apresentada na Figura 4. Através do filtro geográfico, visível na Figura 5, disponível no *eBird* limitamos a nossa pesquisa para a região de Castelo Branco, o que nos possibilitou confirmar a presença da espécie nesta área especifica.

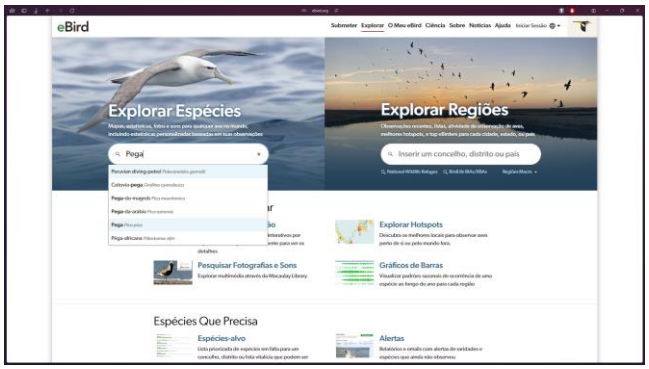

Figura 4. Procura no eBird pela espécie Pega. Fonte: [20]

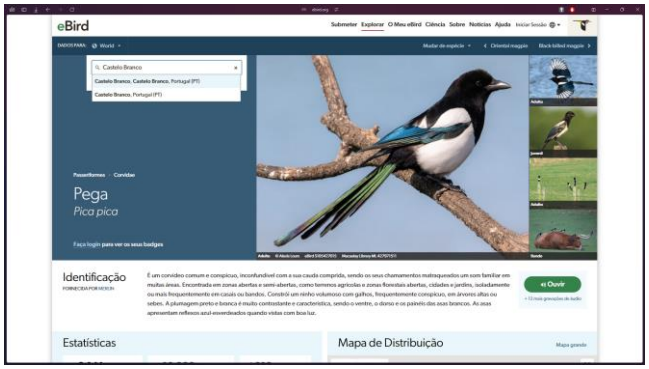

Figura 5. Filtro pela região geográfica. Fonte: [22]

Através da análise dos dados obtidos, verificámos que existem mais de 2000 observações da espécie Pega em Castelo Branco, acompanhadas de algumas imagens, como ilustrado na Figura 6. No entanto, apenas um áudio estava disponível para análise.

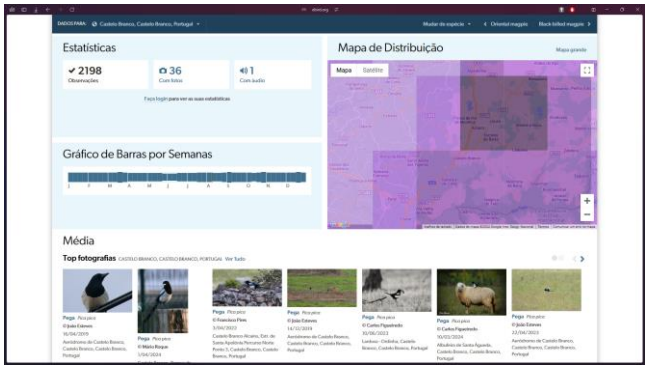

Figura 6. Dados sobre a deteção da espécie Pega em Castelo Branco. Fonte: [22]

Este áudio, apresentado na Figura 7, foi registado no dia 29 de abril de 2021 e recebeu uma classificação de 4 estrelas, atribuída com base em três avaliações de utilizadores. Apesar de algumas interferências, como o vento, o som da espécie é claramente audível.

Ao analisar os detalhes do ficheiro podemos observar que o mesmo é um ficheiro MP3 com uma duração de 16 segundos e com tamanho de 501 KB com uma velocidade de transmissão (bitrate) de 256 kbps.

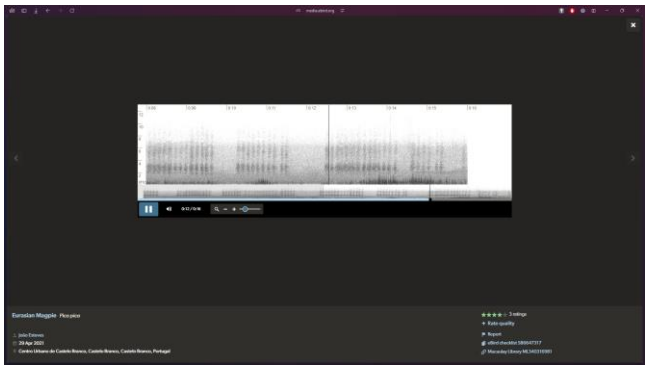

Figura 7. Ficheiro de áudio da espécie Pega em Castelo Branco. Fonte: [23]

Depois, realizámos um teste com este áudio para avaliar a capacidade de deteção da espécie com recurso ao *BirdNET*. Os resultados do teste, visíveis na Figura 8, revelaram a eficácia do sistema na identificação da Pega, mesmo na presença de ruídos ambientais, tendo detetado com uma confiança de entre 97.25 % a 99.49 %.

Selection	View	Channel	Begin File	Begin Time (s)	End Time (s)	Low Freq (Hz)	High Freq (Hz)	Species
1	Spectrogram 1	1	audio.mp3	0	3.0	0	15000	eurmag1 Eurasian Magpie 0.9725	
2	Spectrogram 1	1	audio.mp3	3.0	6.0	0	15000	eurmag1 Eurasian Magpie 0.9885	
3	Spectrogram 1	1	audio.mp3	6.0	9.0	0	15000	eurmag1 Eurasian Magpie 0.9502	
4	Spectrogram 1	1	audio.mp3	9.0	12.0	0	15000	eurmag1 Eurasian Magpie 0.9949	
5	Spectrogram 1	1	audio.mp3	12.0	15.0	0	15000	eurmag1 Eurasian Magpie 0.8750	
6	Spectrogram 1	1	audio.mp3	12.0	15.0	0	15000	bkbmag1 Black-billed Magpie	0.1131
7	Spectrogram 1	1	audio.mp3	15.0	18.0	0	15000	eurmag1 Eurasian Magpie 0.3513	
8	Spectrogram 1	1	audio.mp3	15.0	18.0	0	15000	bkbmag1 Black-billed Magpie	0.1578

Figura 8. Resultados da análise do ficheiro da espécie Pega em Castelo Branco usando o *BirdNET*.

Na próxima etapa iremos preparar o cenário para a implementação e testes subsequentes, seguindo duas abordagens diferentes.

3.2 Abordagem 1

Nesta abordagem, os nós distribuídos são responsáveis pela captura dos ficheiros de áudio no ambiente. Estes ficheiros são transmitidos através da rede WMN para um nó ponte, que faz a ligação entre a rede WMN e o nó central. O nó central, com maior capacidade de processamento, recebe os áudios e realiza a análise para identificar as espécies de aves presentes. Com base nessa análise, são determinadas as ações a tomar, com vista à dispersão de aves. Por exemplo, dar ordem ao *drone* para sobrevoar a zona afetada.

Para o desenvolvimento da nossa solução, optámos por utilizar os módulos *D1 Mini Pro* [24] como nós distribuídos, sendo que um dos módulos foi designado para atuar como nó ponte, conforme ilustrado na Figura 9. O nó ponte desempenha um papel crucial servindo como a ligação entre a rede WMN e nó central que permite a comunicação entre os nós distribuídos e o nó central.

O *D1 Mini Pro*, baseado no microcontrolador *ESP8266*, é adequado para esta função devido às suas capacidades de comunicação de rede de baixa potência e alta eficiência. Com um processador de 32 *bits* operando a 80 MHz, 4 MB de memória *flash* e 80 KB de *SRAM*. Este módulo tem o poder de processamento necessário para executar as funções de nó ponte e nó distribuído sem comprometer o desempenho da rede WMN. Além disso, a sua arquitetura permite a integração fácil com protocolos de rede de comunicação distribuída. Em [25] podemos verificar a capacidade do *ESP8266* para lidar com redes WMN, incluindo a sua resiliência a falhas e a eficiência na transmissão de pacotes, o que justifica a escolha do *D1 Mini Pro* como nó distribuído.

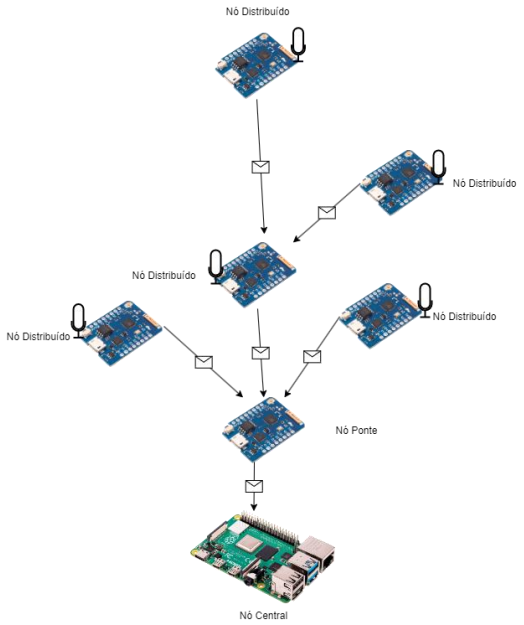

Figura 9. Arquitetura geral da abordagem 1.

A configuração dos diversos *D1 Mini Pro* como nós distribuídos foi feita através do *Arduino Integrated Development Environment (IDE)* versão 2.3.3. Para implementar a rede WMN foi utilizada a biblioteca *PainlessMesh* [26] versão 1.5, representada na Figura 10. Esta biblioteca foi escolhida pela sua capacidade de criar redes WMN auto-organizadas e descentralizadas com facilidade. Uma das principais vantagens da *painlessMesh* é a sua compatibilidade com dispositivos baseados no *ESP8266*, como o *D1 Mini Pro*, o que facilita a configuração e gestão da rede. Além disso, a biblioteca oferece suporte nativo à reconexão automática de nós, garantindo que a rede se reorganize dinamicamente em caso de falhas ou remoção de dispositivos.

Para garantir o funcionamento adequado do sistema, instalámos também as dependências necessárias, apresentadas na Figura 11. Sendo elas a *ArduinoJSON* [27] versão 7.0.4, *TaskScheduler* [28] versão 3.8.5, e *ESPAsyncTCP* [29] versão 1.2.4, que proporcionam o suporte à troca de mensagens entre nós e a gestão eficiente das tarefas assíncronas. Além disso, o pacote *esp8266* [30] versão 3.1.2 mostrado na Figura 12, foi instalado para assegurar a compatibilidade e o reconhecimento do *D1 Mini Pro* pelo *Arduino IDE*.

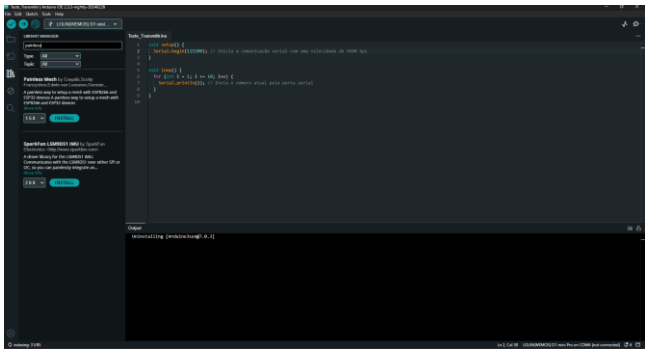

Figura 10. Instalação da biblioteca *painlessMesh*.

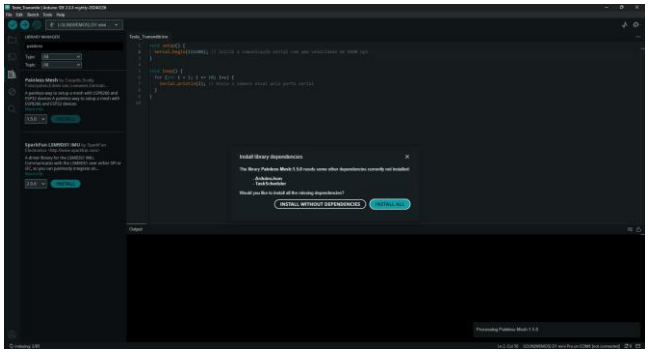

Figura 11. Instalação das dependências necessárias para a *painlessMesh*.

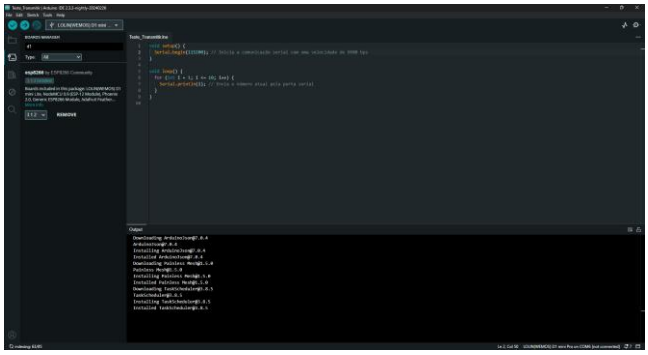

Figura 12. Instalação da biblioteca *esp8266*.

O nó ponte será configurado como o nó raiz da rede WMN e será conectado ao *Raspberry Pi 4B* através dos pinos *general-purpose input/output* (*GPIO)* de 5V, *GND*, 14 e 15 do *Raspberry Pi 4B*, visível na Figura 13, e 5V, *GND*, *RX* e *TX* do *D1 Mini Pro*, visível na Figura 15. Esta configuração permite que o nó central receba as mensagens da rede WMN e que as transmita para o nó central. Para a comunicação entre o nó ponte e o nó central, será utilizada a biblioteca *PySerial* [31] versão 3.5 visível na Figura 13 no *Raspberry Pi 4B*, que possibilita a receção de dados via interface *serial*.

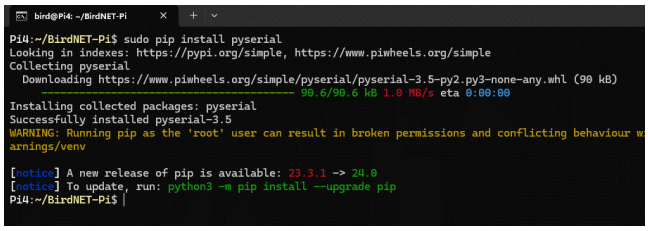

Figura 13. Instalação da biblioteca *PySerial*.

Figura 14. Nó central com ligação *GPIO* para o nó ponte.

Figure 15. Nó ponte com ligação *GPIO* para o nó central.

3.2.1 Software

Após a escolha do hardware e a obtenção de um áudio de referência, o próximo passo seria o desenvolvimento do software para a rede WMN. Este processo foi dividido em várias etapas de maneira a garantir a integração e o funcionamento da mesma.

Inicialmente, foi desenvolvido um *script* para transferir o ficheiro de áudio do computador para os nós distribuídos, utilizando as bibliotecas *FS [32]* e *LittleFS* [33] para gerir o sistema de ficheiros no *D1 Mini Pro*. Nos nós distribuídos, o sistema de ficheiros foi devidamente inicializado, com uma verificação do sucesso dessa operação, para que caso ocorra uma falha seja registada uma mensagem de erro. Após feita esta verificação, procede-se à identificação de uma conexão *serial* ativa. Quando uma conexão é estabelecida, é criado um ficheiro denominado '*audio.mp3*' em modo de escrita, seguido de uma verificação adicional para assegurar que o ficheiro foi criado corretamente. Após esta verificação, os dados são armazenados no ficheiro, conforme representado no fluxograma da Figura 16.

No lado do computador, uma ligação *serial* é estabelecida, abrindo-se o ficheiro de áudio em modo de leitura. Um *loop* é então implementado para monitorizar continuamente a existência de dados a serem lidos. Quando dados estão disponíveis, estes são transmitidos pela conexão *serial*, como representado no fluxograma da Figura 17.

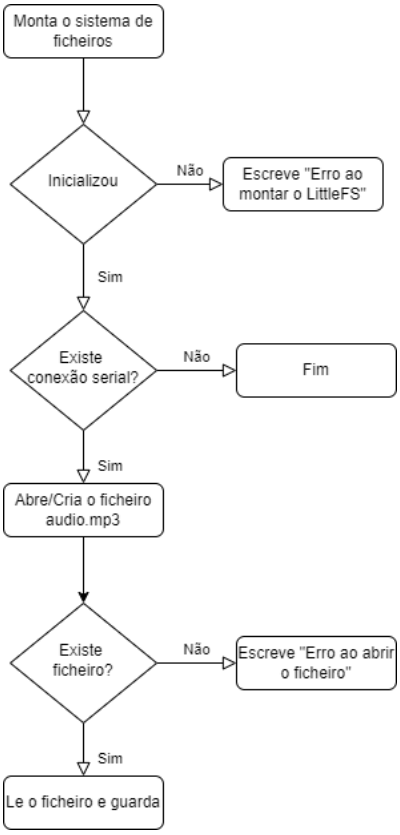

Figura 16. Fluxograma para receção do ficheiro áudio nos nós distribuídos.

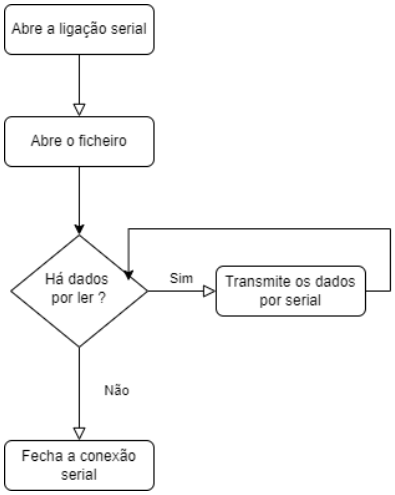

Figura 17. Fluxograma para transmissão do ficheiro para os nós distribuídos.

Em seguida, foi configurada a rede WMN com uma estrutura inicial simples, projetada para testar a conectividade entre os nós. Nessa configuração, cada nó enviava o seu *ID* para toda a rede utilizando *broadcast*, como ilustrado no fluxograma da Figura 18, garantindo que existia comunicação entre todos os nós. Esta configuração inicial, embora básica, permitiu verificar se existia conectividade total entre os dispositivos. Após confirmar o funcionamento correto da comunicação, o *script* foi modificado para direcionar as mensagens exclusivamente ao nó raiz, utilizando o seu *ID*. O fluxograma da Figura 19 demonstra o comportamento do sistema após essa adaptação, com as mensagens sendo enviadas diretamente para o nó central.

Figura 18. Fluxograma do envio de mensagem *broadcast* com ID do nó distribuído.

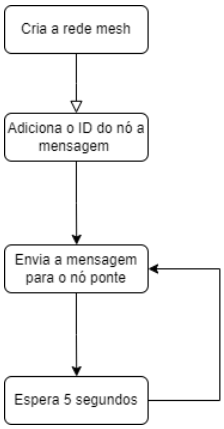

Figura 19. Fluxograma do envio de mensagem dirigida ao nó ponte com o *ID* do nó distribuído.

No nó ponte, o *script* foi criado de maneira a receber exclusivamente mensagens e transmiti-las via comunicação *serial*, permitindo que sejam visualizadas no nó central, conforme representado na Figura 20. A Figura 21 apresenta o fluxograma do *script* que corre no nó central que é responsável por receber as mensagens enviadas pelo nó ponte e exibi-las no terminal, garantindo a monitorização das comunicações entre os nós da rede WMN e o nó central.

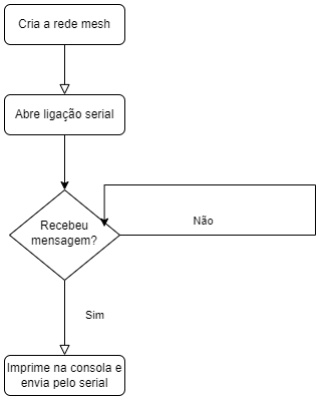

Figura 20. Fluxograma da receção e escrita de dados no nó central.

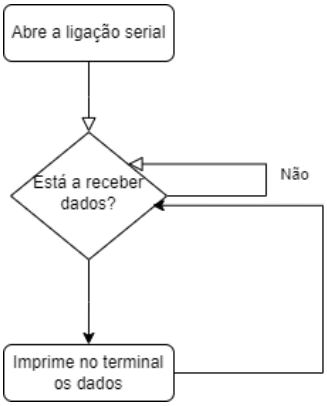

Figura 21. Fluxograma do código nó ponte para receção das mensagens e reencaminhamento para o nó central.

Após a implementação da rede, conseguimos verificar que existia transferência de mensagens entre os nós distribuídos da rede e entre a rede e o nó central. Esta validação foi fundamental para garantir que a comunicação estava a funcionar conforme esperado. Depois, com base nestes resultados, procedemos à alteração dos *scripts* para permitir o envio do ficheiro de áudio (previamente gravado) a partir dos nós distribuídos.

O primeiro passo no envio do áudio consistiu em dividir o ficheiro de áudio em blocos. Este processo envolveu a leitura do ficheiro de áudio e a divisão dos dados em pacotes de um tamanho adequado. Devido à quantidade reduzida de *RAM* dos *D1 Mini Pro*, escolhemos 256 *bytes* e o envio sequencial de cada bloco para o nó central. O fluxograma deste processo, apresentado na Figura 22 ilustra as etapas, desde a leitura do ficheiro até a verificação de recebimento no nó raiz.

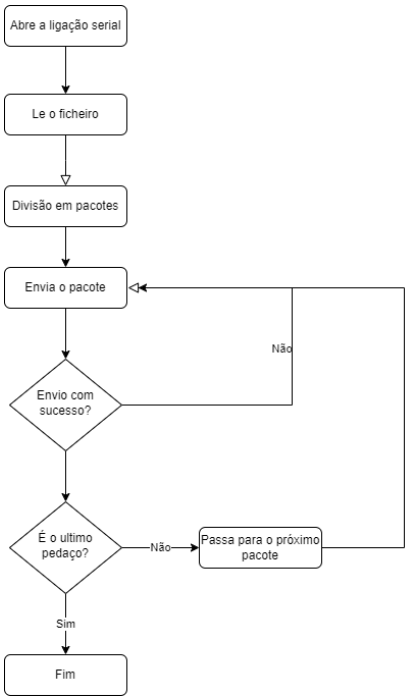

Figura 22. Fluxograma do envio do áudio através de blocos.

Em seguida, o código no *Raspberry Pi 4B* foi ajustado para receber os dados enviados pelos nós distribuídos. O *script* no *Raspberry Pi* aguardava pela receção dos blocos e, assim que todos os dados foram recebidos, tentava reconstituir o ficheiro original a partir dos blocos. No entanto, apesar da implementação adequada, os dados não foram recebidos com sucesso. Ao verificar o ficheiro gerado, notámos que o mesmo apresentava um tamanho reduzido e, consequentemente, não funcionava como esperado.

Com o intuito de resolver o problema de transmissão, decidimos experimentar o envio dos dados em *raw binary*. Este novo método envolveu a leitura do ficheiro de áudio como dados binários, que seriam enviados divididos em blocos. O fluxograma da Figura 23 demonstra o processo da leitura do ficheiro em formato binário e o envio para o nó central.

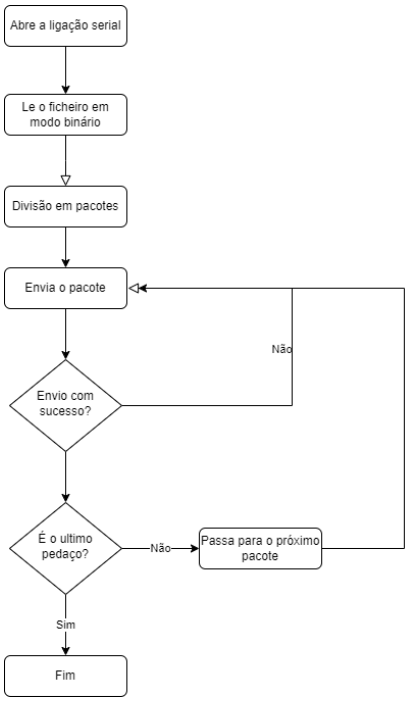

Figura 23. Fluxograma do envio do áudio através de blocos binários.

No lado do *Raspberry Pi*, o código foi novamente adaptado para receber os dados binários e gravá-los num novo ficheiro áudio. Embora esta abordagem tenha permitido que algo fosse recebido, a análise do ficheiro resultante revelou que apenas alguns bytes tinham sido transferidos. O tamanho final do ficheiro gerado era de apenas 1 KB, muito inferior ao tamanho original do áudio de 501KB, e o ficheiro estava danificado.

Face às dificuldades de transferência do áudio pela rede WMN para posterior processamento no nó central, foi necessário encontrar uma alternativa que viabilizasse este objetivo. Assim, decidimos implementar uma segunda abordagem em que o áudio é analisado localmente nos nós distribuídos, enviando apenas os resultados da análise para o nó central.

3.3 Abordagem 2

Nesta abordagem, os nós distribuídos são responsáveis pela captura, análise e processamento local dos áudios capturados. Assim, é no nó distribuído que a espécie de pássaro é identificada e o nível de confiança, e esta informação é enviada para os nós ponte. Estes nós ponte reencaminham as mensagens para o nó raiz, que por sua vez, transmite as informações para o nó central. O nó central utiliza esses dados, recebidos via rede WMN, para decidir sobre as ações necessárias para dispersar as aves (i.e., dar ordem ao *drone* para sobrevoar a zona afetada).

Para esta abordagem, é necessário realizar alterações no *setup* previamente descrito, especialmente em relação aos dispositivos utilizados e à identificação dos mesmos. Como o processamento será feito localmente, é fundamental contar com nós mais robustos e capazes de realizar a análise do áudio, levando à introdução dos *Raspberry Pi 02W* nas extremidades da rede.

Assim, os nós distribuídos são implementados em *Raspberry Pi 02W* e passarão a ser responsáveis pela análise e processamento do áudio. Os *D1 Mini Pro*, que anteriormente desempenhavam essa função, agora atuarão como intermediários na comunicação e serão denominados nós ponte. O *D1 Mini Pro* que estabelece a ligação entre a rede WMN e o nó central será agora designado como nó raiz. Estas alterações na arquitetura da solução podem ser observadas na Figura 24.

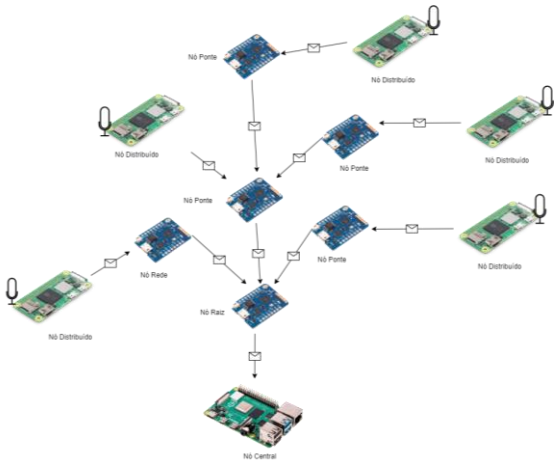

Figura 24. Arquitetura geral da Abordagem 2.

3.3.1 Software

Nesta abordagem foi necessário realizar não apenas alterações nos dispositivos utilizados, mas também na componente de software, face às mudanças no comportamento dos nós da rede e aos novos nós distribuídos.

Para o envio das mensagens, foi preciso definir tanto o formato quanto as informações a serem transmitidas pela rede. Para tal, desenvolveu-se um novo protocolo de mensagens com o seguinte formato:

```
(IdNo: <id>, Especie: <nome>, Confianca: <confiança>, timeStamp: <timeStamp>)
```

Neste protocolo, a informação sobre a espécie detetada (*Especie*), o nível de confiança (*Confianca*) e a data e hora em que foi capturada (*timeStamp*) são enviadas pelo nó distribuído para o nó ponte correspondente e este adiciona o seu identificador (*IdNo*) e encaminha a mensagem para o nó raiz. Deste modo, conseguimos identificar a posição do nó através de uma base de dados com as localizações dos nós, que embora não seja o foco principal deste projeto, permite uma visão espacial da rede. Além disso, incluímos os dados essenciais sobre a espécie de ave detetada, a percentagem de confiança e a data e hora, para tratamento estatístico futuro.

Para assegurar que os dados são entregues corretamente ao nó central e evitar uma sobrecarga desnecessária na rede WMN, é necessário que os nós ponte enviem as mensagens de forma direta e exclusiva para o nó raiz. Para garantir este comportamento, utilizámos o método integrado na *API* da biblioteca *PainlessMesh* denominado *newCallbackConnection* [26], que é invocado sempre que uma nova conexão é estabelecida na rede. Este método fornece o *ID* do novo nó conectado e permite ao nó raiz enviar uma mensagem no formato "*Root: <id>"* para o novo nó. Do lado dos nós da rede, utilizámos o método integrado chamado *newRecievedCallback* [26] que é invocado sempre que um nó recebe uma mensagem da rede. Desta forma, quando um nó recebe uma mensagem da rede WMN, verifica se a mesma começa com "*Root:* " e, se for o caso, o nó extrai o *ID* do nó raiz e armazena-o numa variável que será usada nas comunicações futuras, garantindo que todas as mensagens sejam enviadas diretamente para o nó raiz.

Durante a fase inicial de testes, o *script* criado foi configurado com os dados da espécie e do nível de confiança pré-preenchidos com os valores *Eurasian Magpie* e 0.9949 respetivamente. Estes dados correspondem aos dados retirados da análise do ficheiro de áudio de teste no intervalo entre os segundos 9 e 12, como ilustrado na Figura 8. O fluxograma representado na Figura 25 demonstra o processo que ocorre no nó distribuído para a preparação e envio da mensagem para o nó da rede, via conexão *serial* enquanto a Figura 26 ilustra o processo de receção da mensagem, adição do *ID* do nó e retransmissão da mensagem do nó ponte para o nó raiz.

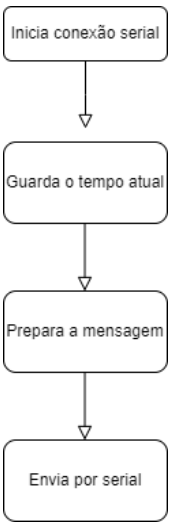

Figura 25. Fluxograma da preparação da mensagem e envio para o nó ponte.

Figura 26. Fluxograma da receção, atualização e encaminhamento da mensagem para o nó ponte.

Após termos implementado com sucesso o protocolo de mensagens na nossa rede, o próximo passo é integrar a análise de ficheiros de áudio e preencher as mensagens com dados reais extraídos dessa análise. Para isso, foi necessário modificar o *script* "*analyze.py*" do *BirdNET-Analyzer* de maneira a permitir que, após a análise do áudio, os resultados sejam retirados diretamente do ficheiro gerado.

Nos resultados são extraídos os dados referentes à espécie de aves identificada com o maior nível de confiança, juntamente com o *timestamp* obtido no momento em que a análise é iniciada. Estes dados são então formatados de forma a coincidirem com o protocolo de mensagens previamente definido e, finalmente, a mensagem formatada é enviada para a rede WMN seguindo o fluxograma apresentado na Figura 27.

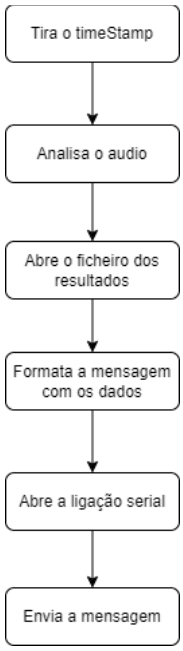

Figura 27. Fluxograma da análise do áudio, preparação e envio de mensagem.

4. TESTES E VALIDAÇÃO

Após implementação da abordagem 2, é essencial testar e validar o desempenho e a eficácia da rede WMN. Esta etapa tem como objetivo avaliar a latência, a confiabilidade e a transmissão de dados em diferentes configurações de nós. Nesta secção, serão apresentados os resultados dos testes realizados e as conclusões obtidas acerca do comportamento da rede em diferentes cenários.

É importante notar que os valores de análise e envio podem variar significativamente, e podem ocorrer *outliers*. Para mitigar a influência de flutuações inesperadas nos resultados, cada teste foi realizado 30 vezes, e utiliza-se a média dos tempos registados como métrica principal. Os tempos que serão monitorizados incluem o tempo de análise, que representa o período necessário para a realização da análise do áudio no nó distribuído; o tempo total, que é a diferença entre o *timestamp* quando o pacote chega e o *timestamp* contido na mensagem; e o tempo de envio, que é calculado subtraindo o tempo de análise do tempo total.

Foi realizada uma série de testes de latência na rede WMN, iniciando com um teste base em que o nó ponte, associado ao nó distribuído, está ligado diretamente ao nó raiz, sem a presença de nós intermédios. Este primeiro teste servirá como referência para avaliar o desempenho da rede. Em seguida, iremos adicionar nós intermédios e alterar a estrutura da rede para observar como estas modificações impactam os tempos de latência.

Na Figura 28, podemos observar alguns dos cenários da rede que pretendemos testar. Nos cenários 1, 2 e 3, o objetivo é verificar o impacto da adição de diversos nós intermédios na latência da rede. O cenário 4 tem duas finalidades: primeiro, determinar qual dos nós seria escolhido para a comunicação e, segundo, testar a capacidade de recuperação da rede em caso de falha de um nó. Por fim, os cenários 5 e 6 serão utilizados para realizar testes de *stress*, onde pretendemos avaliar o impacto na latência quando todas as mensagens são enviadas por um único nó intermédio em comparação com envios realizados por nós intermédios diferentes.

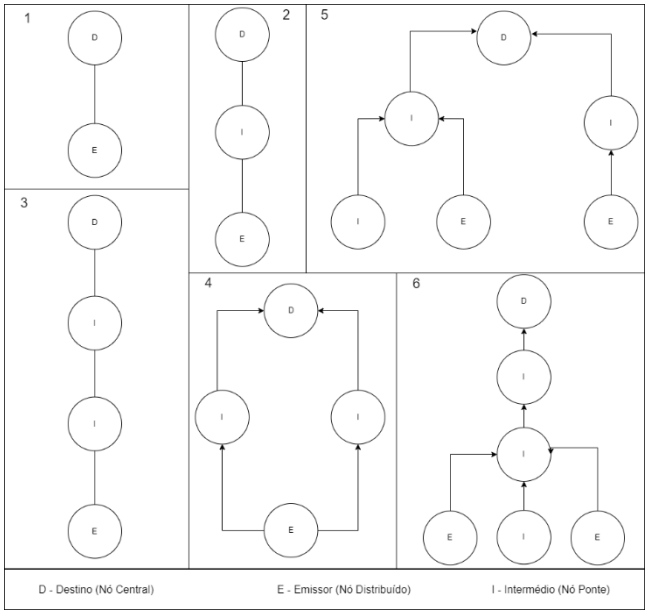

Figura 28. Cenários de teste da rede.

Partindo da referência dos tempos registados no nó diretamente conectado, apresentados na Figura 29, ao observar os valores obtidos nos cenários 1, 2 e 3, podemos concluir que existe uma diferença nos tempos de envio, sendo esta de 0,16 segundos com a introdução de um nó intermédio (Figura 30) e de 0,25 segundos ao introduzir um segundo nó intermédio (Figura 31). No entanto, não se pode concluir definitivamente que a presença de nós intermédios melhora o tempo de envio, pois, ao analisar os 30 tempos que retirámos para calcular a média, verificámos um tempo de quase 6 segundos no cenário 1. Se substituirmos esse tempo por um valor mais próximo dos obtidos, a média dos tempos apresenta uma diferença mínima em relação aos restantes.

Apesar de os resultados mostrarem que não há um grande impacto no tempo de latência, a adição de nós intermédios contribuiu para a melhoria da rede. Em pomares de grandes dimensões, esses nós servem como pontos de retransmissão, encurtando a distância entre os nós distribuídos e o nó central, o que resulta numa melhor qualidade do sinal e adiciona redundância aos caminhos.

Figura 29. Tempos registados no cenário 1.

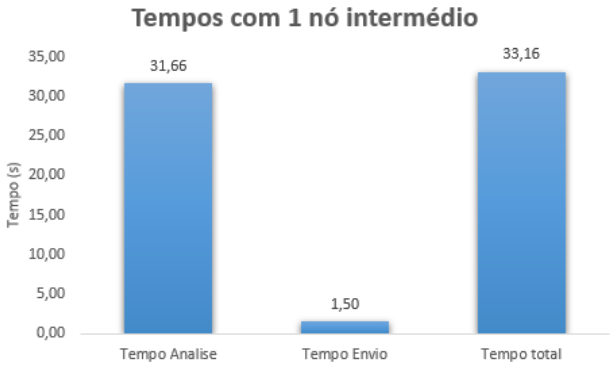

Figura 30. Tempos registados no cenário 2.

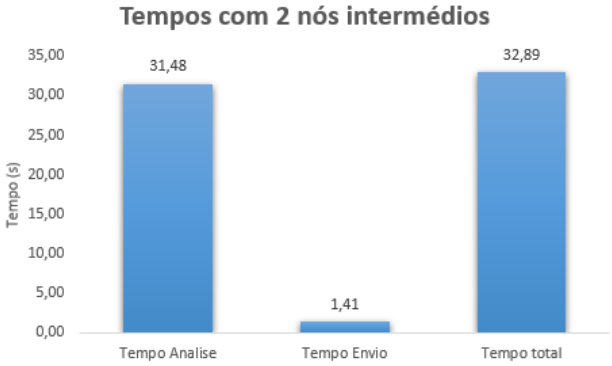

Figura 31. Tempos registados no cenário 3.

Ao tentar implementar o cenário 4, deparámo-nos com um comportamento inesperado da biblioteca *painlessMesh*, que não nos permitiu criar o cenário conforme planeado. Na tentativa de colocar dois nós à mesma distância, ambos ligados ao destino e ao emissor, verificámos que a rede considerava sempre um dos dois intermédios como "filho" do outro, resultando num cenário idêntico ao cenário 3.

Devido a essa limitação, não foi possível testar os cenários 4, 5 e 6 conforme pretendido. Contudo, para avaliar a recuperação da rede e realizar o teste de latência sob condições de sobrecarga, decidimos realizar dois testes alternativos no cenário ilustrado na Figura 32, onde temos um nó identificado como E/I devido à sua dupla funcionalidade.

No primeiro teste, utilizaremos este nó como intermédio para avaliar a recuperação da rede. Durante uma transmissão, o nó será removido para verificar se a rede consegue recuperar, qual o impacto no tempo de envio e se a mensagem chega ao destino ou é perdida. Em seguida, o nó atuará como emissor, permitindo o envio simultâneo de várias mensagens de ambos os emissores. Isso resultará na sobrecarga do nó ponte associado a um dos emissores, que receberá as mensagens de ambos os emissores.

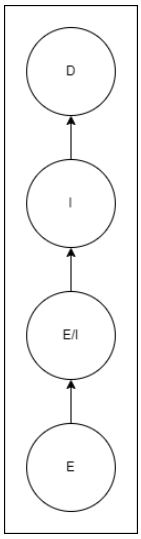

Figura 32. Cenário alternativo para testes.

A topologia de rede descrita na Figura 33 ilustra o cenário representado na Figura 32, onde o nó raiz está ligado a um nó intermédio, que, por sua vez, se conecta a um nó híbrido. Este nó híbrido desempenha simultaneamente as funções de emissor e intermédio, mas, neste caso, opera como intermédio e está ligado a um nó exclusivamente emissor. Conforme mostrado na Figura 34, três mensagens consecutivas foram emitidas através deste nó emissor, utilizando o *script* previamente modificado. Após a receção da primeira mensagem, a ligação com o nó intermédio foi interrompida para forçar a rede a reorganizar-se, permitindo a análise do tempo necessário para esta reorganização, bem como a verificação da receção das mensagens seguintes.

```
18:48:40.957 -> Topologia da rede:
18:48:40.989 -> {
18:48:40.989 ->   "nodeId": 2785280611,
18:48:41.021 ->   "root": true,
18:48:41.021 ->   "subs": [
18:48:41.054 ->     {
18:48:41.054 ->       "nodeId": 2785281017,
18:48:41.085 ->       "subs": [
18:48:41.085 ->         {
18:48:41.117 ->           "nodeId": 2785280700,
18:48:41.150 ->           "subs": [
18:48:41.150 ->             {
18:48:41.181 ->               "nodeId": 576125939
18:48:41.213 ->             }
18:48:41.245 ->           ]
18:48:41.245 ->         }
18:48:41.245 ->       ]
18:48:41.288 ->     }
18:48:41.288 ->   ]
18:48:41.288 -> }
```

Figura 33. Topologia da rede inicial.

```
root@raspberrypi:/home/bird/BirdNET-Analyzer# python3 scriptAnaliseEnvio.py --i ../audio_9_a_12_seg.mp3 --o
 res.csv
Species list contains 6522 species
Analyzing ../audio_9_a_12_seg.mp3
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Mensagem enviada: Especie: Eurasian Magpie, Confianca: 0.9965, Timestamp: 2024-10-08 18:49:05
Finished ../audio_9_a_12_seg.mp3 in 22.76 seconds
root@raspberrypi:/home/bird/BirdNET-Analyzer# python3 scriptAnaliseEnvio.py --i ../audio_9_a_12_seg.mp3 --o
 res.csv
Species list contains 6522 species
Analyzing ../audio_9_a_12_seg.mp3
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Mensagem enviada: Especie: Eurasian Magpie, Confianca: 0.9965, Timestamp: 2024-10-08 18:49:36
Finished ../audio_9_a_12_seg.mp3 in 21.86 seconds
root@raspberrypi:/home/bird/BirdNET-Analyzer# python3 scriptAnaliseEnvio.py --i ../audio_9_a_12_seg.mp3 --o
 res.csv
Species list contains 6522 species
Analyzing ../audio_9_a_12_seg.mp3
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Mensagem enviada: Especie: Eurasian Magpie, Confianca: 0.9965, Timestamp: 2024-10-08 18:51:02
Finished ../audio_9_a_12_seg.mp3 in 26.08 seconds
root@raspberrypi:/home/bird/BirdNET-Analyzer# ^C
root@raspberrypi:/home/bird/BirdNET-Analyzer#
```

Figura 34. Envio das mensagens consecutivas.

Na Figura 35, pode-se verificar que após a receção da primeira mensagem, a rede inicialmente detetou a perda do nó intermédio e mostrou uma topologia que continha apenas com o nó raiz e, posteriormente, reconfigurou-se completamente para incluir os restantes nós. O tempo de reorganização total, que corresponde à segunda atualização da topologia, foi de aproximadamente 1 minuto e 26 segundos, calculado a partir da diferença entre o momento da receção da primeira mensagem (18:49:18:585) e o instante em que a nova topologia, incluindo todos os nós, foi estabelecida (18:50:44:871).

Durante este processo, apenas a terceira mensagem foi entregue com sucesso, indicando que a segunda mensagem foi perdida devido à falha momentânea da rede. Para mitigar este tipo de perda, seria necessário implementar mecanismos que assegurem a retransmissão de mensagens não entregues ou que evitem a sua perda em caso de falhas na rede.

```
18:49:18.585 -> IdNo: 576125939, Especie: Eurasian Magpie, Confianca: 0.9965, Timestamp: 2024-10-08 18:49:05
18:49:45.507 -> Topologia da rede:
18:49:45.540 -> {
18:49:45.540 ->   "nodeId": 2785280611,
18:49:45.540 ->   "root": true
18:49:45.582 -> }
18:50:44.871 -> Topologia da rede:
18:50:44.902 -> {
18:50:44.902 ->   "nodeId": 2785280611,
18:50:44.934 ->   "root": true,
18:50:44.967 ->   "subs": [
18:50:44.967 ->     {
18:50:44.967 ->       "nodeId": 576125939,
18:50:44.998 ->       "subs": [
18:50:45.030 ->         {
18:50:45.030 ->           "nodeId": 2785280700
18:50:45.062 ->         }
18:50:45.062 ->       ]
18:50:45.098 ->     }
18:50:45.098 ->   ]
18:50:45.098 -> }
18:51:04.865 -> IdNo: 576125939, Especie: Eurasian Magpie, Confianca: 0.9965, Timestamp: 2024-10-08 18:51:05
```

Figura 35. Receção das mensagens e atualização da topologia.

Após a verificação dos tempos de latência da rede no seu estado normal, com um único nó a enviar mensagens, torna-se necessário avaliar o comportamento da rede quando vários nós transmitem em simultâneo, como ocorreria num pomar de maiores dimensões e extensão. Para tal, inicialmente, foi realizado um teste onde um emissor enviou 30 mensagens e os tempos correspondentes foram registados como base de comparação. Em seguida, as mesmas 30 mensagens foram enviadas quase ao mesmo tempo por dois nós emissores, com o objetivo de criar um cenário de *stress* na rede, especialmente nos nós responsáveis pela ligação à raiz.

A análise dos tempos obtidos revelou um aumento ligeiro de 0,15 segundos na média do tempo de envio, conforme apresentado na Figura 36. Isso indica que, mesmo com apenas dois nós a enviar mensagens através de nós intermédios, já é possível notar uma diferença no desempenho da rede. No gráfico da Figura 37 essa diferença é evidenciada de forma mais clara, através da comparação dos valores mínimos e máximos. Embora o tempo máximo para o envio das mensagens tenha aumentado apenas em 0,8 segundos, representando um acréscimo de 4%, o tempo mínimo para o envio apresentou um aumento de 0,26 segundos, o que equivale a um incremento significativo de 21%. Este aumento é particularmente relevante, correspondendo a quase um quarto de tempo a mais no envio, e salienta a necessidade de implementar protocolos que melhorem a eficiência do envio de mensagens.

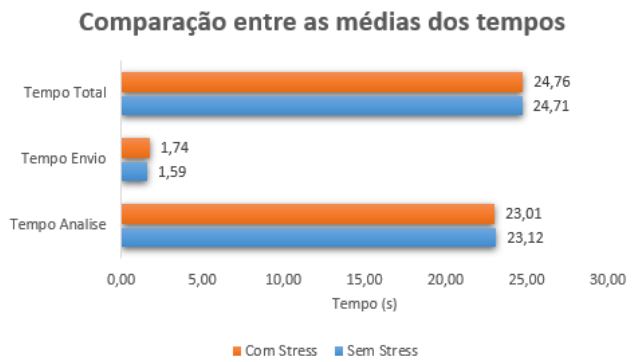

Figura 36. Comparação entre as médias dos tempos.

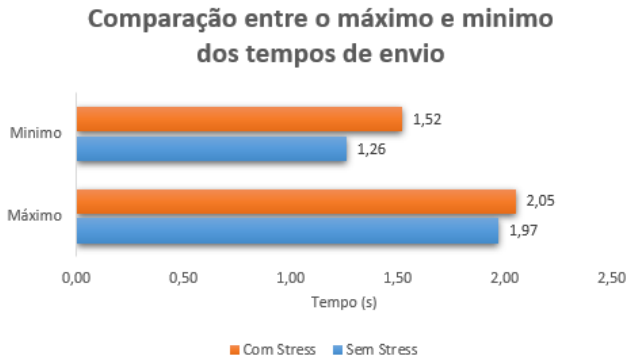

Figura 37. Comparação entre o tempo de envio mínimo e máximo.

Em resumo, apesar de os testes realizados serem simples é possível concluir que, embora a rede esteja funcional e não apresente tempos de latência extraordinariamente elevados, há espaço significativo para melhorias. É aconselhável a utilização de protocolos com vista a melhorar o desempenho da rede. Será fundamental testar a rede numa escala maior para observar de que forma os tempos de latência aumentam com o crescimento do número de nós e de mensagens na rede [25].

5. CONCLUSÃO

É sabido que as aves são responsáveis por elevadas perdas nas colheitas em pomares e que a aplicação de métodos tradicionais de dispersão de aves é ineficiente. Nesse sentido, este trabalho parte de uma proposta de solução para este problema. Considera a utilização de uma rede em malha sem fios, para propagar a informação de deteção de aves, recolhida através de sensores nas árvores de um pomar, para posterior tentativa de dispersão com recurso a um *drone*.

As redes em malha sem fios são uma solução eficaz para permitir a comunicação entre dispositivos em áreas de grande extensão. A sua principal característica é a interconexão distribuída dos nós, permitindo a transmissão de dados por múltiplos caminhos, o que garante uma maior resistência à falha de nós individuais. Assim, a sua utilização pode ser promissora em ambientes rurais, onde a conectividade é frequentemente limitada.

No entanto, neste estudo, identificaram-se diversos problemas associados à implementação de uma rede WMN no cenário considerado, tais como a perda de mensagens, a latência elevada, a corrupção de dados. Para mitigar estes problemas, procurando garantir a entrega rápida e fiável da informação, foram consideradas e avaliadas duas abordagens distintas para a implementação da rede WMN. Testou-se e validou-se o desempenho da abordagem que melhor atende às necessidades do cenário considerado neste estudo. Foram também identificados pontos em aberto para trabalho futuro.

AGRADECIMENTOS

Vasco N. G. J. Soares e João M. L. P. Caldeira reconhecem que este trabalho foi financiado por fundos nacionais através da FCT – Fundação para a Ciência e a Tecnologia, I.P., e, quando elegível, cofinanciado por fundos comunitários no âmbito do projeto/apoio UID/50008/2025 – Instituto de Telecomunicações, com o identificador DOI <https://doi.org/10.54499/UID/50008/2025>

Pedro D. Gaspar agradece o apoio financeiro da Fundação para a Ciência e a Tecnologia (FCT), I.P., através do projeto UIDB/0151/2025, Centro de Ciências e Tecnologias Mecânicas e Aeroespaciais (C-MAST). DOI: https://doi.org/10.54499/UID/00151/2025.

A responsabilidade pelos conteúdos deste artigo é exclusivamente dos autores e não reflete a opinião da FCT.

CONTRIBUIÇÕES DOS AUTORES

Fábio Cardoso, Daniel Carvalho: investigação, metodologia, análise formal, validação, preparação de rascunho de redacção-original.

Pedro D. Gaspar: revisão-escrita e edição, supervisão.

Vasco N. G. J. Soares, João M. L. P. Caldeira: análise formal, validação, revisão-escrita e edição, supervisão.

Os autores declaram não haver conflito de interesses.

REFERÊNCIAS

[1] E. B. Micaelo, L. G. P. S. Lourenço, P. D. Gaspar, J. M. L. P. Caldeira, and V. N. G. J. Soares, “Bird Deterrent Solutions for Crop Protection: Approaches, Challenges, and Opportunities,” *Agriculture*, vol. 13, no. 4, p. 774, Mar. 2023, doi: 10.3390/agriculture13040774.

[2] A. Roihan, M. Hasanudin, and E. Sunandar, “Evaluation Methods of Bird Repellent Devices in Optimizing Crop Production in Agriculture,” *J Phys Conf Ser*, vol. 1477, no. 3, p. 032012, Mar. 2020, doi: 10.1088/1742-6596/1477/3/032012.

[3] A. Muminov, Y. C. Jeon, D. Na, C. Lee, and H. S. Jeon, “Development of a solar powered bird repeller system with effective bird scarer sounds,” in *2017 International Conference on Information Science and Communications Technologies (ICISCT)*, IEEE, Nov. 2017, pp. 1–4. doi: 10.1109/ICISCT.2017.8188587.

[4] K. Nagy, T. Cinkler, C. Simon, and R. Vida, “Internet of Birds (IoB): Song Based Bird Sensing via Machine Learning in the Cloud : How to sense, identify, classify birds based on their songs?,” in *2020 IEEE SENSORS*, IEEE, Oct. 2020, pp. 1–4. doi: 10.1109/SENSORS47125.2020.9278714.

[5] L. G. P. S. Lourenço, E. B. Micaelo, P. D. Gaspar, J. M. L. P. Caldeira, and V. N. G. J. Soares, “PROTÓTIPO DE SOLUÇÃO DE DETEÇÃO E DISPERSÃO DE AVES PARA PROTEÇÃO DE COLHEITAS,” *Revista de*

Sistemas e Computação, vol. 13, no. 3, pp. 34–42, 2023, doi: 10.36558/rsc.v13i3.8490.

[6] F. Cardoso, D. Carvalho, P. D. Gaspar, V. N. G. J. Soares, and J. M. L. P. Caldeira, "A Study on Acoustic Bird Detection in the Context of Smart Agriculture," *Revista de Sistemas e Computação - RSC (aceite para publicação)*.

[7] I. F. Akyildiz and Xudong Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, Sep. 2005, doi: 10.1109/MCOM.2005.1509968.

[8] J. W. R. K. Yaling Yang, "Designing routing metrics for mesh networks," Dec. 2005.

[9] Ren Yueqing and Xu Lixin, "A study on topological characteristics of wireless sensor network based on complex network," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, IEEE, Oct. 2010, pp. V15-486-V15-489. doi: 10.1109/ICCASM.2010.5622543.

[10] A. Esmailpour, N. Nasser, and T. Taleb, "Topological-based architectures for wireless mesh networks," *IEEE Wirel Commun*, vol. 18, no. 1, pp. 74–81, Feb. 2011, doi: 10.1109/MWC.2011.5714028.

[11] H. Yu, D. Wu, and P. Mohapatra, "Experimental Anatomy of Packet Losses in Wireless Mesh Networks," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, IEEE, Jun. 2009, pp. 1–9. doi: 10.1109/SAHCN.2009.5168933.

[12] S. Han, X. Zhu, A. K. Mok, D. Chen, and M. Nixon, "Reliable and Real-Time Communication in Industrial Wireless Mesh Networks," in *2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, IEEE, Apr. 2011, pp. 3–12. doi: 10.1109/RTAS.2011.9.

[13] G. Egeland and P. E. Engelstad, "A Model for the Loss of Hello-Messages in a Wireless Mesh Network," in *2010 IEEE International Conference on Communications*, IEEE, May 2010, pp. 1–6. doi: 10.1109/ICC.2010.5502278.

[14] P. Cappanera, L. Lenzini, A. Lori, G. Stea, and G. Vaglini, "Optimal joint routing and link scheduling for real-time traffic in TDMA Wireless Mesh Networks," *Computer Networks*, vol. 57, no. 11, pp. 2301–2312, Aug. 2013, doi: 10.1016/j.comnet.2012.11.021.

[15] O. T. Valle, A. V. Milack, C. Montez, P. Portugal, and F. Vasques, "Experimental evaluation of multiple retransmission schemes in IEEE 802.15.4 wireless sensor networks," in *2012 9th IEEE International Workshop on Factory Communication Systems*, IEEE, May 2012, pp. 201–210. doi: 10.1109/WFCS.2012.6242568.

[16] J. Singh and J. Singh, "A Comparative Study of Error Detection and Correction Coding Techniques," in *2012 Second International Conference on Advanced Computing & Communication Technologies*, IEEE, Jan. 2012, pp. 187–189. doi: 10.1109/ACCT.2012.2.

[17] U. S. Priyanka Shrivastava, "Error Detection and Correction Using Reed Solomon Codes," 2013.

[18] Biodiversity4All, "Castelo Branco - Pega," Biodiversity4All.

[19] BirdGard, "Como Afugentar Aves das Plantações," BlogBirdGard.

[20] eBird, "Explore."

[21] Cornell Lab of Ornithology, "BirdNET: The Machine Learning Solution for Bird Sound Identification," https://birdnet.cornell.edu/.

[22] eBird, "Common Magpie (Eurasian)," eBird.

[23] Cornell Lab of Ornithology, "Eurasian Magpie Audio," Cornell Lab of Ornithology. Accessed: Aug. 12, 2024. [Online]. Available: https://cdn.download.ams.birds.cornell.edu/api/v2/asset/340316981.mp3

[24] L. Shenzhen WEMOS Electronics Co., "D1 Mini Pro," https://www.wemos.cc/en/latest/d1/d1_mini_pro.html. Accessed: Aug. 12, 2024. [Online]. Available: https://www.wemos.cc/en/latest/d1/d1_mini_pro.html

[25] L. Santos, T. Costa, J. M. L. P. Caldeira, and V. N. G. J. Soares, "Performance Assessment of ESP8266 Wireless Mesh Networks," *Information (Switzerland)*, vol. 13, no. 5, May 2022, doi: 10.3390/info13050210.

[26] G. Mag, "painlessMesh API Documentation," https://github.com/gmag11/painlessMesh/wiki/api.

[27] B. Blanchon, "ArduinoJSON: Efficient and Elegant JSON Serialization in C++ for Embedded Systems," https://www.arduino.cc/reference/en/libraries/arduinojson/.

[28] A. Arkhipenko, "TaskScheduler: Cooperative Multitasking for Arduino," https://www.arduino.cc/reference/en/libraries/taskscheduler/.

[29] dvarrel, "ESPAsyncTCP: Asynchronous TCP Library for ESP8266," https://www.arduino.cc/reference/en/libraries/espasynctcp/.

[30] K. Söderby, "ESP32: A Guide to Using ESP32 with Arduino Cloud," https://docs.arduino.cc/arduino-cloud/guides/esp32/?queryID=935fdd6506525ed3c494e3b64b016338.

[31] C. Liechti, "PySerial: Python Serial Port Access Library," https://pythonhosted.org/pyserial/.

[32] I. Grokhotkov, "Arduino-ESP8266: Filesystem API," https://arduino-esp8266.readthedocs.io/en/latest/filesystem.html#fs-api.

[33] E. Philhower, "Arduino LittleFS Upload," https://github.com/earlephilhower/arduino-littlefs-upload.