

Protótipo de Solução de Deteção e Dispersão de Aves para Proteção de Colheitas

Leonardo G.P.S. Lourenço
Instituto Politécnico de Castelo Branco
Branco
Av. Pedro Álvares Cabral
nº 12, 6000-084 Castelo Branco,
Portugal
leonardo.lourenco@ipcbrcampus.pt

Eduardo B. Micaelo
Instituto Politécnico de Castelo Branco
Branco
Av. Pedro Álvares Cabral
nº 12, 6000-084 Castelo Branco,
Portugal
eduardo.micaelo@ipcbrcampus.pt

Pedro D. Gaspar
C-MAST
Universidade da Beira Interior
Rua Marquês d'Ávila e Bolama,
6201-001 Covilhã, Portugal
dinis@ubi.pt

João M. L. P. Caldeira
Instituto Politécnico de Castelo Branco
Instituto de Telecomunicações
Portugal
jcaldeira@ipcb.pt

Vasco N. G. J. Soares
Instituto Politécnico de Castelo Branco
Instituto de Telecomunicações
Portugal
vasco.g.soares@ipcb.pt

RESUMO

Os animais são uma das pragas que constituem uma ameaça para a produtividade das culturas destinadas ao consumo humano. As perdas de colheitas causadas por aves representam um desafio sério e dispendioso para os agricultores. O trabalho apresentado neste artigo parte de um estudo prévio dos mesmos autores que concluiu sobre a ineficiência das técnicas dissuasoras comumente utilizadas para dispersão de aves nas colheitas. Assim, num contexto de agricultura inteligente, propõe o uso de drones, inteligência artificial, sensores e IoT, para tentar resolver este problema. Apresenta uma proposta de arquitetura e descreve o processo de implementação de um protótipo, incluindo os seus componentes de hardware e software. Descreve um conjunto de testes realizados para validação do conceito e identifica pontos de trabalho em aberto.

Palavras-Chave

Dissuasão de aves, Proteção de colheitas, Agricultura Inteligente, Internet das Coisas, Drone, Rede Mesh, Sensores, Protótipo.

ABSTRACT

Animals are one of the pests that pose a threat to the productivity of crops intended for human consumption. Crop losses caused by birds represent a serious and costly challenge for farmers. The work presented in this article builds on a previous study by the same authors that concluded on the inefficiency of the deterrent techniques commonly used to disperse birds on crops. Thus, in the context of smart agriculture, it proposes the use of drones, artificial intelligence, sensors and IoT to try to solve this problem. It presents a proposed architecture and describes the process of implementing a prototype, including its hardware and software components. It describes a set of tests carried out to validate the concept and identifies open points of work.

Keywords

Bird Deterrence, Crop Protection, Smart Agriculture, Internet of Things, Drone, Mesh Network, Sensors, Prototype.

1. INTRODUÇÃO

As pragas animais constituem uma grande ameaça para a produtividade das culturas de consumo humano. Os prejuízos causados pelas aves em culturas frutícolas constituem um problema significativo e dispendioso para os agricultores. As estimativas das perdas causadas por diferentes espécies de aves foram apresentadas num estudo realizado na Suécia entre 2000 e 2015 [1]. Nesse período registaram-se 2194 queixas de danos nas culturas, correspondendo a uma perda aproximada de 34 500 toneladas.

Outro estudo [2] teve como objetivo descobrir que espécies de aves estavam diretamente relacionadas com os danos causados às culturas. Foram recolhidos dados em 60 plantas selecionadas aleatoriamente. O estudo centrou-se em quatro culturas diferentes de 81 locais de campo: 23 de cereja doce; 12 de mirtilo; 24 de maçã e 22 de vinha. Os danos causados pelas aves foram estimados em 2,3% nos campos de maçãs, 3,6% nas culturas de vinhas, 22% nos mirtilos e 26,8% em pomares de cerejas doces. Foram também realizados inquéritos aos agricultores responsáveis por essas culturas em Nova Iorque, Michigan, Washington, Oregon e Califórnia. Só em Nova Iorque indicaram que, tiveram perdas de cerca de 6,6 milhões de dólares por ano e que 65,6% dos agricultores inquiridos estariam a tomar medidas para combater a presença das aves nos seus campos. Metade desses agricultores confirmou ainda que as aves são o principal fator de perda de colheitas.

Um estudo realizado na Polónia [3] concluiu que, em 1974 e 1980, 22% e 16%, respetivamente, dos pomares de cerejas foram destruídos por aves. O mesmo estudo realizou outro inquérito em quatro distritos da Polónia. Nesses quatro distritos as perdas de milho variaram entre 22% e 32%, as perdas de trigo entre 10% e 13%, as perdas de cevada entre 3 e 18% e as perdas de aveia entre 8 e 15%.

Estes resultados demonstram que as aves são grandes causadoras de perdas nas culturas e que o uso de técnicas de dissuasão de aves pode contribuir de forma significativa para a diminuição dessas perdas.

Estas técnicas podem ser classificadas em seis grandes grupos: visuais, auditivas químicas, exclusão, modificação de habitat e remoção. As técnicas visuais provocam um estímulo visual nas aves que pode resultar em medo ou curiosidade. O sentimento de perigo pode ser provocado por um predador real ou simulado. Pode também ser utilizado algo com que as aves não estão familiarizadas, como espantalhos, tintas com cores vibrantes, luzes, fita refletora, papagaios de papel, balões ou outros. Alguns destes repelentes visuais podem também incorporar dissuasores sonoros [4]–[7]

As técnicas auditivas funcionam através de estímulos auditivos. Para aplicar esta técnica pode ser utilizada pirotecnia, canhões de gás, sons de predadores entre outros. Algumas das técnicas auditivas apresentam também uma vertente visual [7]

As técnicas químicas utilizam químicos com o objetivo de causar desconforto, desorientar ou até mesmo tirar a vida as aves. Uma destas técnicas químicas faz uso de uma substância naturalmente presente nas uvas, que inibe as aves de se alimentarem de alimento com esta substância, pois estas não gostam do sabor [7]

As técnicas de exclusão consistem em impedir fisicamente que as aves possam aceder a uma determinada área [8]. Nas técnicas de modificação do habitat, é em alterado o meio que as aves gostam de habitar, de modo que sejam obrigadas a procurar outro local [9], [10]. As técnicas de remoção consistem em retirar à força as aves de um determinado ambiente, através de armadilhas ou da caça [11]–[14].

O trabalho apresentado neste artigo parte de um estudo prévio dos mesmos autores [8], que relata o estado da arte sobre soluções de dissuasão de aves para proteção de colheitas, identifica os principais desafios e as lacunas atuais, e propõe orientações para investigação futura. Assim, apresenta-se neste artigo, um protótipo de solução de deteção e dispersão de aves para proteção de colheita. Este protótipo utiliza uma rede de sensores aplicados em diversas árvores de um pomar, que permitem a deteção de presença de aves pelo som. Esta informação é recolhida e utilizada para enviar um drone para essa localização, com vista a dispersar as aves. Este processo é gerido de forma autónoma não sendo necessária intervenção humana.

Este artigo encontra-se estruturado da seguinte forma. A secção 2 apresenta o problema, descreve a solução proposta e a arquitetura do protótipo. Na secção 3, é descrito o processo de implementação do protótipo e as escolhas de hardware e software. A secção 4, apresenta os testes de validação do protótipo. Por fim a secção 5 conclui o artigo e apresenta o trabalho futuro.

2. PROBLEMA

As diferentes técnicas comumente utilizadas para dispersão de aves nas colheitas, têm em comum o fator de habituação nas aves. Ou seja, o estímulo torna-se habitual para a ave e consequentemente deixa de ser eficaz. Para combater a habituação, no caso dos estímulos visuais e auditivos, é necessário mudar frequentemente o local onde estão instalados os dispositivos. Outro problema é o fator humano, ou seja, para que seja possível utilizar técnicas tais como a falcoaria ou a pirotecnia, são necessárias pessoas qualificadas. Algumas técnicas tais como a falcoaria, a captura e os químicos, podem trazer consequências à vida da ave ou causar problemas ao ecossistema local.

2.1 Proposta

Para dar resposta ao problema identificado, este trabalho apresenta uma solução para dispersão de aves, baseada na utilização de drones autónomos. A deteção de pássaros, na solução apresentada, é feita através da utilização de microfones colocados junto dos pontos da colheita a proteger. Assumindo como cenário um pomar, com várias árvores de fruto, os microfones seriam colocados junto de cada uma dessas árvores. Ao ser detetada a presença de aves, através da captação do seu som, esta informação seria comunicada a uma estação base que daria ordem a um drone para se deslocar ao local. No local o drone realizaria um conjunto de ações, na tentativa de dispersão das aves. O trabalho apresentado assenta no desenvolvimento de uma infraestrutura de suporte à recolha de sons em pontos georreferenciados, o encaminhamento dessa deteção para uma estação base, o envio de ordens de deslocação de um drone aos pontos georreferenciados e a deslocação autónoma desse drone aos pontos identificados. As manobras de dispersão das aves, que possam vir a se realizadas pelo drone, estão fora do âmbito deste trabalho.

Ao contrário de outras soluções estudadas, esta proposta não traz, consequências físicas para as aves, opera de forma autónoma (não necessitando de qualquer intervenção humana) e com a possibilidade de criar variações nas manobras de dispersão realizadas pelo drone, é esperado que também não cause habituação das aves.

2.2 Arquitetura

A solução proposta envolve a construção de vários componentes e a sua interligação. Nesta secção será descrita a organização da solução proposta, em termos da sua arquitetura e dos vários componentes que a integram. Para captação do som das aves, como referido, são usados pequenos controladores, denominados nós sensores, instalados junto das áreas a monitorar (árvores). Para a captura do som, estes nós sensores integram um microfone. Numa área alargada serão usados vários destes nós sensores. Ao ser detetado som, por parte de um nó sensor, essa informação deve ser encaminhada para um outro controlador central, denominado de nó gateway.

Para facilitar a implantação destes dispositivos nos cenários de aplicação, a interligação entre os nós sensores e o nó gateway é feita através de comunicação sem fios. Por essa razão, tanto os nós sensores como a estação base integram módulos de comunicação sem fios. O nó gateway é responsável por fazer a interligação entre os nós sensores e o drone.

Para garantir a interligação entre todos os nós sensores e o nó gateway, será usada uma topologia de rede em *mesh*. O uso desta tecnologia permite que o envio de dados entre um nó e um ponto de destino na rede possa ser feito sem que haja uma ligação direta entre os dois. Para isso são usados os próprios nós que integram a rede para o encaminhamento de mensagens até que estas possam chegar ao seu destino. Esta facilidade permite estender a área de cobertura da conectividade de uma rede sem fios de forma a garantir que os dados possam chegar ao seu destino.

No caso da arquitetura da solução proposta, esta tecnologia será usada para interligar todos os nós sensores, espalhados numa área que pode ser bastante alargada, ao nó gateway. Quando percecionado um som num dos nós sensores, esta informação é encaminhada através da rede *mesh* até ao nó gateway. O nó gateway ao receber esta informação comunica ao controlador do drone para que este se desloque ao local onde o som foi percecionado. A Figura 1 apresenta a arquitetura da solução proposta mostrando todos os seus componentes e a interligação entre eles. Para teste, validação e demonstração desta proposta foi desenvolvido um protótipo cujo a construção é detalhada nas secções seguintes.

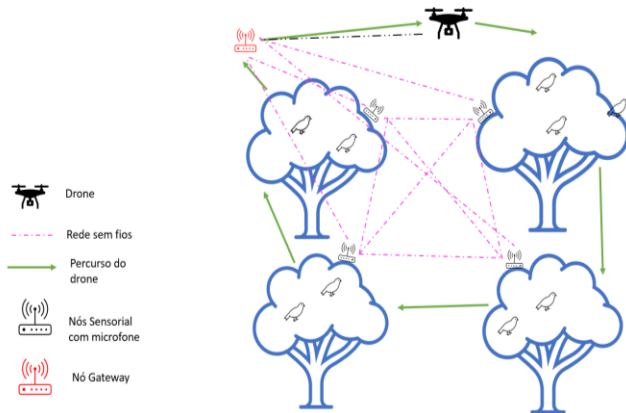


Figura 1. Diagrama com a arquitetura do protótipo.

3. PROTÓTIPO

Esta secção apresenta os componentes de hardware e software que compõem a arquitetura do protótipo. Está dividida em duas partes, na primeira parte são apresentados os nós equipados com microfones que capturam os ruídos das aves e que por sua vez formam a rede mesh, é também descrito o funcionamento desta. Na segunda parte, são descritos todos os componentes de gestão de dados recebidos da rede mesh, e também o funcionamento do controlador do drone responsável pela criação do plano de voo para o drone levar a cabo.

3.1 Deteção de Aves

Neste protótipo foram utilizadas mini placas wifi Wemos D1 Mini Pro [15], [16] visíveis na Figura 2 a), para a deteção de aves e formarem rede mesh. Estes circuitos são amplamente utilizados para o desenvolvimento de soluções de IOT devido á sua versatilidade e baixo custo [17]. Possuem módulos WiFi, 16 MB de armazenamento flash e são modulares, ou seja, têm 11 pins digitais aos quais é possível ligar outros dispositivos [15], [16]. Neste caso foi utilizado sensor auditivo, um microfone Sound Sensor V2, Figura 2 b), este é um microfone digital, logo apenas capta variações no som, ou seja, permite saber se há ruído ou não, sendo impossível distinguir o ruído [18] Estes circuitos são programáveis através do IDE do Arduino, para isso é necessário instalar o Driver CP210 [19], para que o Wemos seja detetado pelo IDE. Neste IDE, foi desenvolvido o código referente à utilização do microfone, para que fosse possível capturar e tratar do sinal capturado pelo microfone e foi também desenvolvido o código para a formação da rede mesh.

Neste protótipo foi utilizada uma rede mesh, pois é eficaz no que toca a cobrir grandes áreas geográficas, devido á sua redundância, facilidade de implementação e eficiência energética [20]. Para que seja possível criar redes mesh tem de ser incluída a biblioteca *PainlessMesh* utilizando o código `"#include <PainlessMesh.h>"`. Esta é uma biblioteca compatível com as placas Wemos que permite criar redes mesh através de conexões WiFi. A grande vantagem desta tipologia de rede é a gestão dinâmica de rotas de encaminhamento dentro da rede, isto é, se algum dos dispositivos que formam a rede falhar ou entrar um novo dispositivo na rede o encaminhamento de dados é ajustado automaticamente, tornando a rede tolerante a falhas e mais eficiente [21], [22].

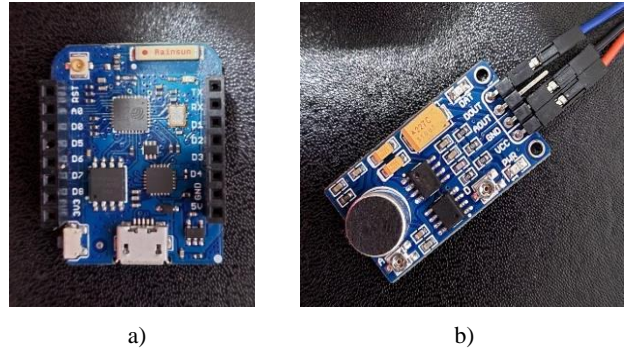


Figura 2. a) Wemos D1MiniPro; b) Sound SensorV2.

Esta rede é formada por três destes dispositivos Wemos, e estes são denominados por nós. Neste protótipo, há dois tipos de nós, os nós sensoriais e o nó gateway. Os nós sensoriais são aqueles que estão equipados com o microfone e estão responsáveis pela deteção de sons e sinalização das coordenadas nas quais foram captados sons. Para configurar o microfone na placa Wemos foi necessário incluir a linha de código `"pinMode(D4, INPUT)"`. Desconstruindo esta linha de código obtemos `"pinMode"` que é utilizado para configurar os pins da placa Wemos, o `"D4"` refere-se ao pin em uso, e o `"INPUT"` configura o modo de como o pin está a ser utilizado, neste caso como entrada de informação [23].

Os nós sensoriais, após detetarem ruídos, enviam as suas coordenadas para o nó seguinte através da rede mesh utilizando a linha de código `"mesh.sendSingle(msg)"`. O fluxograma da Figura 3, descreve o funcionamento deste nó, nele podemos ver o ponto S, que representa o dispositivo com o sensor. Inicialmente, este dispositivo conecta-se á rede mesh e, uma vez estabelecida a conexão aguarda a deteção ruídos de aves e/ou receber coordenadas de outros nós. Quando deteta ruídos ou recebe dados de outros nós, encaminha as coordenadas para o nó seguinte na rede, até que as informações alcancem o nó gateway.

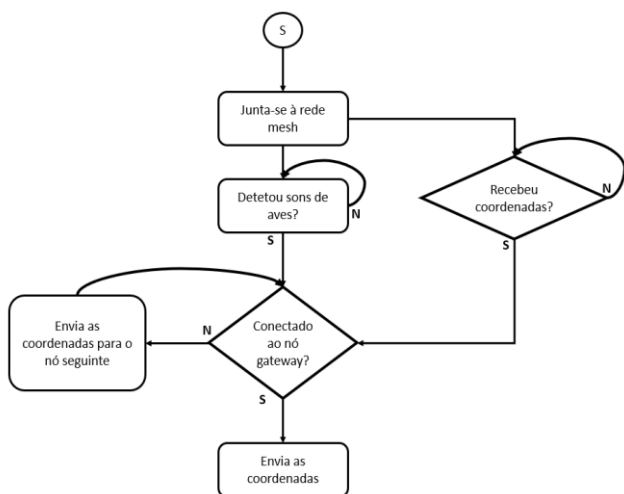


Figura 3. Fluxograma do nó sensorial.

O nó gateway recebe a informação da rede mesh, e envia-a para a base central, descrito na secção seguinte, através da porta Serial. Para habilitar a transmissão de informações através desta porta, foi necessário incluir “*Serial.println()*” [24] no código desenvolvido. Na Figura 4 está descrito o funcionamento deste nó. Este nó está representado pelo ponto G. Inicia-se conectando à rede mesh e fica à espera de receber dados, quando recebe os dados, transmite-os através da porta Serial que está conectada de forma física ao Raspberry Pi (base central) responsável por enviar as informações ao controlador do drone.

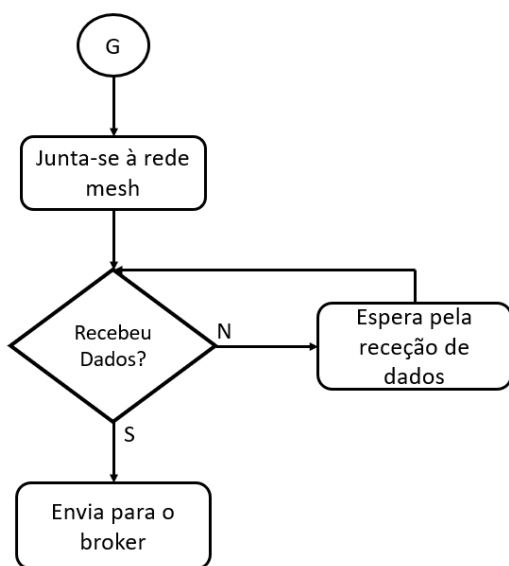


Figura 4. Fluxograma do nó gateway.

3.2 Atuação do Drone

A partir da rede mesh, através do nó gateway, é recebida a informação com as coordenadas de onde foram detetadas aves. Para isto, foi ligada a porta serial do nó gateway a um computador Raspberry Pi 3B+ com o sistema operativo Raspberry Pi OS, apresentado na Figura 5. Estes computadores são versáteis e de

pequenas dimensões. O modelo utilizado está equipado com um processador Broadcom BCM2837 de 64bits a 1,4GHz e 1GB de RAM [25]. Tal como os dispositivos Wemos, este também tem pins, aos quais é possível conectar outros dispositivos. Nesta arquitetura são utilizados dois destes computadores. Um deles está conectado ao nó gateway e denomina-se de base central. Este é o responsável por receber e formatar toda a informação recebida da rede mesh, e enviá-la para o controlador do drone. Para esta transmissão de dados, é utilizado o protocolo de comunicação MQTT[26], [27]. Este protocolo precisa para o seu funcionamento de funciona um broker, neste caso optamos por usar o Mosquitto [28]. De um *publisher* e de um *subscriber*, que na nossa arquitetura serão ambos os Raspberry Pi, visto que ambos enviam e recebem informações do broker [29]. Por último, esta informação precisa de ter tópicos para ser enviada, em que o *publisher* cria os tópicos no broker para o *subscriber* poder subscrever esses mesmos tópicos no broker para receber a informação. Como descrito acima, ambos a base central e o controlador do drone vão ser *publisher* e *subscriber* ao mesmo tempo. Para aplicar este protocolo, foi desenvolvido um script na linguagem Python.



Figura 5. Raspberry Pi 3B+.

No fluxograma da Figura 6, podemos observar em detalhe o funcionamento da base central. De início, a base central conecta-se ao servidor MQTT e fica à espera de receber informações provenientes da rede mesh. Após receber essa informação verifica se o drone está disponível para efetuar o voo. No caso de não estar disponível, as coordenadas são escritas num buffer se ainda não constarem neste. Quando o drone está disponível, são lhe enviadas as coordenadas. Após receber as coordenadas o controlador de voo confirma a recepção e envia o plano ao drone. Quando o drone finaliza o voo, é feita a verificação se há ou não dados no buffer.

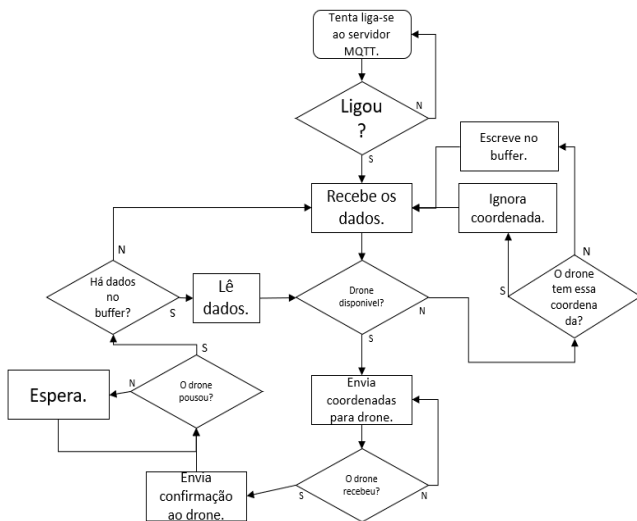


Figura 6. Fluxograma com o envio de dados do nó gateway para a base central.

Como é possível ver na Figura 7, são criados 3 tópicos pelo controlador do drone, os dados que passam neles são confirmações para garantir que as informações vitais foram recebidas. Informações estas tais como: as coordenadas necessárias para construir o plano de voo, se o drone já pousou, e para confirmar se conseguiu conectar-se ao servidor MQTT.

Ao ligar-se ao servidor MQTT, o cliente tem acesso às coordenadas enviadas e pode assim criar o plano de voo para ser executado pelo drone. Este plano de voo precisa apenas da coordenada para ser construído. O controlador do drone vai somar e subtrair 0.000065 graus à latitude e à longitude para criar mais 4 pontos para o plano de voo, para este circular o ponto de onde chegou o pedido. A base central só precisa de receber uma coordenada uma vez para ser imediatamente enviada ao drone ou ficar guardada no buffer. Caso a base envie coordenadas que estejam no buffer para o controlador do drone, este funciona no estilo *first in first out*.

Após o drone executar o plano de voo, envia a confirmação do pouso para o controlador de drone, que envia essa informação para a base central, e após isso fica à espera de novas informações de voo. O diagrama da Figura 8 representa de modo geral o funcionamento a arquitetura descrita anteriormente, desde o envio de dados do nó gateway até ao drone.

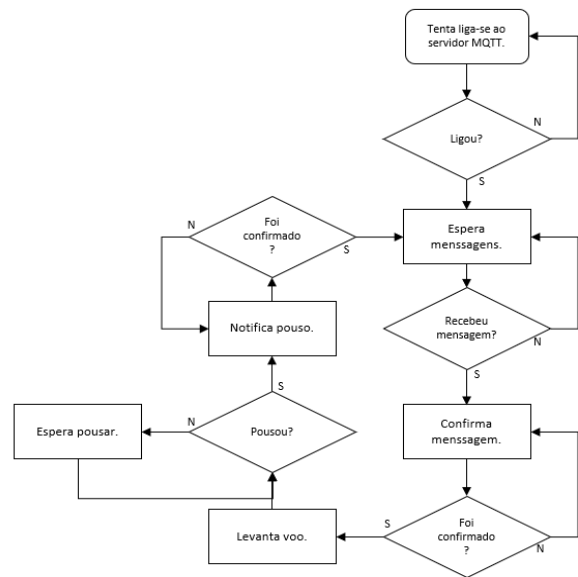


Figura 7. Fluxograma da ligação MQTT no controlador do drone.

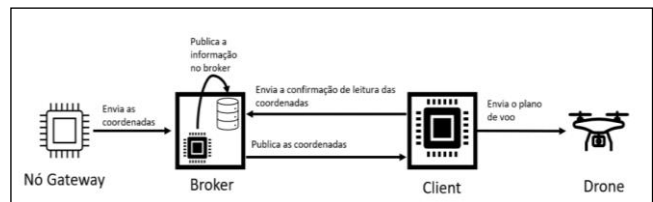


Figura 8. Diagrama de funcionamento da arquitetura.

4. TESTES DE VALIDAÇÃO

Nesta secção são apresentados os testes realizados ao protótipo para prova de conceito e avaliar o comportamento e desempenho.

Para efeitos de teste e validação da solução proposta foi utilizado um simulador de drone. O simulador usado foi o jMAVSim[30], que simula um drone 3D num ambiente virtual, apresentado na Figura 9. Este simulador recebe os dados através do MavLink [31], que é o protocolo de comunicação utilizado entre sistemas de drones e outros dispositivos, como controladores ou outros dispositivos de controlo de voo, em ambientes reais.

Este simulador foi executado num computador com o sistema operativo Linux 20.04 com 16Gb de Ram DDR4 e com um processador AMD geração Zen2 de 3.7Ghz.



Figura 9. Simulador jMAVSim.

Para que fosse possível verificar o funcionamento do drone e criar rotas de voo foi utilizado o QGroundControl[32]. Este software é utilizado em sistemas de veículos aéreos não tripulados. Permite verificar todos os dados de telemetria do drone, tais como altitude, coordenadas atuais e percentagem de bateria, permite também verificar todo o percurso feito pelo drone e planear missões de voo. Estes dados são meramente apresentados no software e em nada influenciam o funcionamento da arquitetura. Foi utilizado com este software o sistema de controlo de voo do PX4 Autopilot [33]. Este sistema é utilizado para que seja possível controlar drones de forma remota, e é também aplicável ao simulador utilizado nos testes de validação.

O processo começou com testes mais simples, para validação do simulador e da rede mesh. Depois partiu-se para um estágio em que se recebiam as mensagens enviadas pela rede mesh no Raspberry Pi. Ao mesmo tempo, executavam-se pedidos para o Raspberry Pi que controlava o drone.

Para validação do protótipo, foram realizados testes considerando dois cenários diferentes. No primeiro cenário, pretende demonstrar o funcionamento de cada sensor, provando que se consegue conectar à rede mesh, captar som e enviar os dados através da rede mesh até ao nó gateway. O segundo cenário, tem como objetivo comprovar o funcionamento da arquitetura, principalmente do algoritmo desenvolvido, para que coordenadas geográficas repetidas não sejam incluídas no plano de voo.

4.1 Cenário 1

Este cenário é composto por um WEMOS D1 Mini Pro, instalado numa coordenada geográfica, que está a uma distância de aproximadamente 5 metros do nó gateway, que por sua vez está ligado na base central, responsável por captar a informação que chega da rede mesh.

Tomando como exemplo a primeira coordenada recebida, a base central irá estabelecer uma comunicação com o Controlador de Drone. Para começarem a comunicação usam um *3-way handshake*, onde na primeira mensagem, a base central envia as coordenadas ao drone e espera 2 segundos para receber uma resposta com o valor booleano verdadeiro. Se não obtiver resposta do controlador do drone ocorre uma retransmissão. Na próxima mensagem, a base central envia uma confirmação para o controlador do drone, com a informação que recebeu a confirmação e irá aguardar por uma mensagem do drone a informar que já pousou.

Do lado do controlador do drone, este recebe as coordenadas e calcula o plano de voo a partir delas. Tenta estabelecer uma ligação com o drone via Wi-Fi. Ao receber a confirmação do drone, envia o plano de voo para o mesmo. Este Raspberry Pi vai ficar também à espera que o drone pouse, para enviar a mensagem correspondente para a base central, para que esse nó depois volte a aceitar coordenadas, para que o ciclo volte ao início novamente.

Resultados

Como se pode ver através do software QGroundControl na Figura 10, o drone chegou com sucesso ao ponto A. O drone executou a rota desejada, que era a que estava no plano de voo, incluindo o ponto central e os outros 4 calculados a partir da coordenada.



Figura 10. Representação da trajetória realizada pelo drone entre o ponto A e B. Sendo A e B os nós sensores auditivos e o drone o elemento a vermelho.

Desconstruindo a arquitetura nas várias etapas para tornar mais visível o que está por detrás dela, na Figura 11 pode-se observar a recolha dos dados do sensor. Essa informação é recebida pelo nó gateway, na Figura 12, e vai ser lida pela base central, para esta tomar as decisões de enviar as coordenadas para o controlador do drone, Figura 13 ou de as descartar, Figura 14. Na Figura 15 é visível o controlador do drone a receber as coordenadas, a estabelecer uma ligação com o drone, a enviar-lhe a informação para levantar voo, e, por último, a informar que está a aguardar que pouse. Podemos ver o que acontece com as coordenadas que chegam enquanto o drone está em missão. Neste caso, a coordenada é sempre igual à coordenada que está no drone, por isso, vai descartá-la como se pode ver na Figura 14.

```
Output Serial Monitor x
Message (Enter to send message to 'LOLIN(WEMOS) D1 mini Pro' on 'COM10')
Som captado em: 39.819149,-7.515305
A enviar para o nó ponte
Som captado em: 39.819149,-7.515305
A enviar para o nó ponte
```

Figura 11. Output gerado pelo nó sensorial.

```
Output Serial Monitor x
Message (Enter to send message to 'LOLIN(WEMOS) D1 mini Pro' on 'COM3')
Mensagem recebida: 39.819149,-7.515305
Mensagem recebida: 39.819149,-7.515305
```

Figura 12. Output do nó gateway.

```
Espero confirmação
False
Waiting for acknowledgment...
b'True'
39.819149,-7.515305
Escrevi no Buffer
39.819149,-7.515305
39.819149,-7.515305
Escrevi no Buffer
39.819149,-7.515305
```

Figura 13. A coordenada recebida foi escrita no buffer.

```
Shell
descartei
39.819149,-7.515305
descartei
39.819149,-7.515305
descartei
39.819149,-7.515305
descartei
39.819149,-7.515305
descartei
```

Figura 13. A coordenada recebida foi descartada.

```
Shell
Conectado. Código de resultado: 0
b'39.819218,-7.514852'
b'ola'
tou aqui
Waiting for drone to connect...
-- Connected to drone!
Waiting for drone to have a global position estimate...
-- Global position estimate OK
-- Uploading Mission
-- Arming
-- Starting mission
-- Waiting for landing
```

Figura 15. Conexão do drone ao Raspberry Pi.

4.2 Cenário 2

No segundo cenário usaram-se os dois microcontroladores WEMOS D1 Mini Pro. Estes estavam numa topologia Daisy Chain, onde os três microcontroladores, incluindo o nó gateway estavam a uma distância de 10 metros entre si. Ou seja, o microcontrolador que capta áudio mais distante, está a uma distância de 20 metros do nó gateway, onde já não é possível estabelecer uma ligação sem que exista um nó entre eles deles para reencaminhar a informação.

Neste cenário, existe também uma ligeira mudança no software. Visto que agora há duas fontes de dados, este teve de ser alterado. Ao chegar uma mensagem, é confirmado se o drone está disponível, e executa o processo tal como no cenário 1. A diferença está quando chega outra coordenada geográfica distinta da que o drone considerou na sua missão atual. Essa nova coordenada é guardada num *buffer*. Quando o drone pouso, a base central vai consultar se existe alguma coordenada antes de ler diretamente das mensagens que estão a chegar da rede mesh. E repete o processo

Resultados

Como se pode observar no software QGroundControl na Figura 10, o drone chegou com sucesso ao ponto A e B. A trajetória foi levantar voo e ir até ao nó A, voltar a pousar, ir até ao nó B, voltar a pousar e ir novamente até ao nó A e pousar.

Desconstruindo a arquitetura nas várias etapas para tornar mais visível o que está por detrás dela, começamos com a Figura 11, onde podemos observar a recolha dos dados nos distintos sensores. Essa informação é recebida pelo nó gateway, Figura 12, e vai ser lida pelo Raspberry Pi que está conectado a ele para tomar as decisões de enviar as coordenadas para o controlador do drone ou de as descartar. É visível na Figura 15 o controlador do drone a receber a coordenada, a estabelecer uma ligação com o drone, a enviar-lhe a informação para levantar voo e por último, a informar que está a aguardar que pouse. De seguida, vemos o que acontece com as coordenadas que chegam enquanto o drone está em missão. Se a coordenada for igual à que está no drone, descarta (Figura 14), caso seja diferente vai guardar num buffer (Figura 13). Este cenário permitiu comprovar que o sistema é capaz de dar resposta à deteção de sons em dois pontos distintos.

O software QgroundControl permitiu ver todo o trajeto efetuado pelo drone. Como se vê na Figura 10, tem-se acesso a esses dados de telemetria e do trajeto levado pelo drone através de uma imagem de satélite. Pode-se observar também, que o drone chega ao ponto onde foi captado o som, Figura 10, e faz um percurso em volta desse ponto central. Este software foi utilizado para permitir visualizar uma rota de voo, e para verificar se estava a ser executada corretamente, visto que com o simulador do drone não é possível analisar o percurso percorrido.

5. CONCLUSÃO

Há diversas pragas que afetam a agricultura, entre elas as aves, que são uma das piores e mais difíceis de combater para os agricultores. As aves são causadoras de elevadas perdas nas colheitas, e têm vindo a ser combatidas com recurso a tecnologias antiquadas e ineficientes. Nesse sentido, neste trabalho apresenta-se uma proposta inovadora de solução de deteção e dispersão de aves, que através de uma rede *mesh*, obtém as coordenadas geográficas onde foram detetadas aves numa cultura, e que depois utiliza um drone que se desloca a esse ponto com o intuito de as dispersar.

Para demonstrar o conceito e validar o protótipo, foram considerados dois cenários. O primeiro cenário visava demonstrar a deteção de som por parte de um nó sensorial equipado com um microfone, a passagem desta informação através da rede *mesh*, a chegada dessa informação ao controlador do drone e o voo levado a cabo pelo mesmo até esse ponto. O segundo cenário, tinha como objetivo validar a capacidade de gestão de coordenadas por parte do controlador do drone, aquando da utilização dos dois nós sensoriais em simultâneo. Com este teste observou-se que, quando são recebidas notificações de duas coordenadas geográficas iguais, uma delas era descartada, para que o drone não vá várias vezes de seguida ao mesmo ponto, e quando era uma coordenada diferente o controlador do drone já a incluía no plano de voo.

O protótipo aqui apresentado apresenta diversas limitações, que serão alvo de trabalho futuro. Pretende-se realizar testes de prova de conceito com recurso a um drone real. Deve investigar-se a utilização de algoritmos para planeamento de rotas de voo, com vista a otimizar a autonomia ao drone. A arquitetura pode também considerar a utilização e gestão de um conjunto de drones, para contribuir para a eficácia e eficiência da dispersão das aves. Pretende-se também utilizar algoritmos de inteligência artificial que permitam distinguir os sons, para que apenas sons de aves sejam tidos em conta no despoletar do voo do drone.

CONTRIBUIÇÕES DOS AUTORES

Leonardo G.P.S. Lourenço, Eduardo B. Micaelo: investigação, metodologia, análise formal, validação, preparação de rascunho de redação-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] Monrás-Janer T., Knape J., Nilsson L., Tombre I., Pärt T., and Månsson J, “Relating national levels of crop damage to the abundance of large grazing birds: Implications for management,” *Journal of Applied Ecology*, vol. 10, pp. 2286–2297, 2019.
- [2] Henrichs, Heidi M., Boulanger, Jason R., Curtis, and Paul D, “Limiting Bird Damage to Fruit Crops in New York: Damage Assessments and Potential Management Strategies for the Future Wildlife Damage Management,” in *Wildlife Damage Management*, 2013, p. 180.
- [3] Pinowski J. and Zajac Z. R., “Damage to crops caused by bird in Central Europe,” *PWN*, pp. 333–345, 1990.
- [4] Frings H. and M. Frings, “Behavioral manipulation (visual, mechanical, and acoustical),” *Academic Press*, pp. 387–475, 1967.
- [5] I. 5. Hussain, “Trapping, netting and scaring techniques for bird control,” *J.E. Brooks, E. Ahmad, I.*, pp. 187–191, 1990.
- [6] R. L. 20. Bruggers *et al.*, “Responses of pest birds to reflecting tape in agriculture,” *Wildl. Soc. Bull.*, vol. 14, pp. 161–170, 1986.
- [7] Transport Canada, “Bird control, background information, Lester B. Pearson International Airport,” Unpublished report WP TAOG 13-1.
- [8] 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.
- [9] D. F. 93. Mott, “Control of wading bird predation at fish-rearing facilities,” *Wading Birds. National Audubon Society*, pp. 131–132, 1978.
- [10] J. 94. Meyer, “Room for birds and fish. RSPB’s survey of heron damage.,” *Fish Farmer 4*, pp. 23–26, 1981.
- [11] J. A. 103. Hardman, “Bird damage to sugar beet,” *Annals of Applied Biology 76*, pp. 337–341, 1974.
- [12] D. 104. Draulans, “The effectiveness of attempts to reduce predation by fish-eating birds,” *Biol Conserv.*, vol. 41, pp. 219–232, 1987.
- [13] M. J. 107. Harrison, “Municipality of Anchorage sanitary landfill bird hazard analysis and mitigation,” Washington, 1986.
- [14] D. 108. Harke, “Wetting agents and their role in blackbird damage control,” in *Proceedings Bird Control Seminar 4*, 1968, pp. 104–108.
- [15] “D1 mini Pro,” https://www.wemos.cc/en/latest/d1/d1_mini_pro.html.
- [16] “WEMOS D1 MINI PRO ESP8266 DEVELOPMENT BOARD,” <https://hobbycomponents.com/development-boards/864-wemos-d1-mini-pro-esp8266-development-board>.
- [17] “What is Arduino? Uses of Arduino - The Complete Guide For Kids and Parents,” <https://www.create-learn.us/blog/what-is-arduino/>.
- [18] “LM386 Sound Sensor, Sound Detector, Compatible with Arduino,” <https://www.waveshare.com/sound-sensor.htm>.
- [19] Paula Petcu, “Getting started with WeMos D1 mini and WeMos D1 mini PRO,” <https://www.monohelixlabs.com/wemos-d1-mini-pro-esp8266.html>.
- [20] “PainlessMesh Technical Documentation,” <https://gitlab.com/painlessMesh/painlessMesh/-/wikis/home>.
- [21] “Intro to PainlessMesh,” <https://github.com/gmag11/painlessMesh>.
- [22] Chris Pollette and Dave Roos, “How Wireless Mesh Networks Work.”

- [23] “digitalRead(),” <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>.
- [24] “Serial.println(),” <https://www.arduino.cc/reference/en/language/functions/communication/serial/println/>.
- [25] “Microcomputador Raspberry Pi 3 Model A+,” https://mauser.pt/catalog/product_info.php?products_id=096-6700.
- [26] Amazon, “What is MQTT?” Accessed: Sep. 06, 2023. [Online]. Available: <https://aws.amazon.com/pt/what-is/mqtt/>
- [27] Guilherme Santos, “Protocolo MQTT: O Que é, Como Funciona e Vantagens.” Accessed: Sep. 06, 2023. [Online]. Available: <https://www.automacaoindustrial.info/mqtt/>
- [28] R. A. Light, “Mosquitto: server and client implementation of the MQTT protocol,” *The Journal of Open Source Software*, vol. 2, no. 13, p. 265, May 2017, doi: 10.21105/joss.00265.
- [29] P. K. Paul and M. K. Ghose, “Cloud Computing: Possibilities, Challenges and Opportunities with Special Reference to its Emerging Need in the Academic and Working Area of Information Science,” *Procedia Eng*, vol. 38, pp. 2222–2227, 2012, doi: 10.1016/j.proeng.2012.06.267.
- [30] “jMAVSim with SITL,” <https://docs.px4.io/main/en/simulation/jmavsim.html>.
- [31] “MAVLink Developer Guide,” <https://mavlink.io/en/>.
- [32] “QGROUNDCONTROL,” <http://qgroundcontrol.com/>.
- [33] “Open Source Autopilot For Drone Developers,” <https://px4.io/>.