DEPARTAMENTO DE CIÊNCIAS E TECNOLOGIAS

MESTRADO EM ENGENHARIA INFORMÁTICA E DE TELECOMUNICAÇÕES

UNIVERSIDADE AUTÓNOMA DE LISBOA
"LUÍS DE CAMÕES"

Sistema de suporte à Previsão e Deteção de Fogos florestais

Trabalho de Projeto para a obtenção do grau de Mestre em Engenharia Informática e Telecomunicações

Autor: Bruno Alexandre Carvalho Martins
Orientador: Professor Dr. António Manuel Caldeira
Número do candidato: 20140954

Fevereiro de 2020
Lisboa

1 Agradecimentos

Em primeiro lugar agradeço à minha família pelo apoio e proporcionar a possibilidade de frequentar a Licenciatura e Mestrado de Engenharia Informática. Em segundo lugar agradecer ao meu Orientador que sempre se mostrou disponível para qualquer dúvida e por autorrecriação foi enviando artigos e tutoriais com o intuito de ajudar a desenvolver o projeto.

2 Resumo

Neste projeto aborda-se o tema de como se detetar preventivamente fogos florestais, usando para tal, sistemas de deteção compostos por dispositivos de *IoT* e *Drones*, que permitirão com um investimento substancial inicial, impedir que fogos florestais ganhem proporções catastróficas com a intervenção preventiva dos meios competentes. Durante este projeto serão abordados os problemas, arquitetura, *middleware*, segurança, protocolos, simulação e análise do desempenho do sistema.

Palavras-chave

Internet of Things; Fogos florestais; Motes; IEEE 802.15.4

3 Abstract

On this project we address the issue of how to preventively detect forest fires, using for that a detection system constituted for IoT and Drone devices, which allow, with an initial investment to prevent forest fires from catastrophic proportions with the preventive intervention of the fire brigades. This project will approach the problems, architecture, middleware, security, protocols, simulation, and analysis of system performance.

Keywords

Internet of Things; Forest Fire; Motes; IEEE 802.15.4

4 Índice

5 Lista de tabelas

6 Lista de Ilustrações

7 Lista de Abreviaturas

MAC	Media Access Control
LC	Logical link Control
PPM	Parts-per-million
LRWPAN	Low Rate Wireless Personal Area Network
LOS	Line of Sight
FFD	Full function device
RFD	Reduced function device
LIDAR	Light detection and ranging systems
QoS	Quality of Service
WSN	Wireless Sensor Network
IoT	Internet of Things
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UAV	Unmanned Aerial Vehicle
AC	Alternating Current
LVDT	Linear Variable Differential Transformer
P2P	Peer-to-Peer
CSMA/CA	Carrier sense multiple access with collision avoidance
WirelessHART	Highway Addressable Remote Transducer Protocol
6LoWPAN	IPv6-based Low power Wireless Personal Area Networks
ISA	International Society of Automation
S_nO_2	Dióxido de estanho
GFSK	Gaussian Frequency Shift Keying
BER	Bit Error Rate
RSSI	Received Signal Strength Indication

NESC	National Electrical Safety Code
OCARI	Open Communication Protocol for Ad Hoc Reliable Industrial Instrumentation
CRC	Cyclic Redundancy Check
PEAS	Probing Environment and Adaptive Sleeping
CLD	Controlled Layer Protocol
MTE	Minimum Transmission Energy
LEACH	The Low-Energy Adaptive Clustering Hierarchy
STCP	Sensor Transmission Control Protocol
PORT	Price-Oriented Reliable Transport Protocol
PSFQ	Pump-slowly, fetch-quickly
XML	Extensible Markup Language
EXI	Efficient XML Interchange
HTTP	Hypertext Transfer Protocol
W3C	World Wide Web Consortium
RPC	Remote procedure-calls
WSDL	Web services description language
WADL	Web application description language
DODAG	Destination Oriented Directed Acyclic Graphs
OF	Objective Function
DIO	DODAG Information Object
ID	Identificador único
DIS	DODAG Information Solicitation
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
RPC	Remote Procedure Call
MVC	Model-View-Controller

8 Glossário

Internet of Things	Descreve um conjunto de dispositivos, que trabalham em redes de grandes dimensões, composta por sensores e outros dispositivos inteligentes. A sua função é recolher os dados operacionais de sensores remotos e permitir a sua utilização em aplicações. (O que é Internet das Coisas?, 2019)
Fresnel Zone	Em ligações ponto a ponto, podem existir obstáculos entre os mesmos, não sendo possível a linha de vista entre ambos. Estes obstáculos podem ser o próprio terreno, vegetação ou prédios, o que resulta em perdas de sinal. Por isso torna-se importante existir uma zona elíptica entre o transmissor e recetor, livre de obstruções para que permita o funcionamento do sistema. A *Fresnel Zone* é as elipses formadas entre os dois pontos, que são determinadas pela frequência de operação e distancia que os separa. O raio da *Fresnel Zone* pode ser calculada através da seguinte fórmula: $Raio(m) = 17{,}31 * \sqrt{\frac{distancia\ (Km)}{4 * frequência\ (GHz)}}$ (What is the Fresnel Zone?, 2019) (Fresnel Zones – What are they and why are they so important?, 2019)
Ad hoc	São redes onde os dispositivos são capazes de se ligar um ao outro sem a necessidade de um ponto de acesso, pois ambos trabalham como se de routers se tratassem, o que torna a ligação rápida e flexível devido a não existir necessidade de utilização de uma infraestrutura fixa. (O que são redes Ad Hoc e quais as suas vantagens?, 2019)
Agilla	É uma solução *middleware*, que adota um paradigma de agente móvel, onde os programas são compostos por agentes, que podem ser migrados entre os nós que compõem uma rede. Cada um dos agentes é semelhante a uma máquina virtual, com instruções e

	memória dedicada. Tal permite aos mesmos, executar instruções especiais que interagem com o ambiente, mover ou clonar de um nó para outro nó. (Fok, Roman, & Lu, 2006) (Pugliese, Pomante, & Santucci, 2009)
Broadcast	É o envio de um pacote em simultâneo para todos os dispositivos pertencentes a um dado domínio ou nós que se encontrem ao seu alcance. (Diferença entre Unicast, multicast e broadcast, 2011)
Multicast	Envio de um pacote de uma interface para todas as interfaces pertencentes a um dado grupo de dispositivos. (Diferença entre Unicast, multicast e broadcast, 2011) (Alkhatib & Baicher, 2012)
Data aggregation	É descrito como uma série de métodos automatizados para combinar dados, provenientes de múltiplos nós de uma *WSN*, transformando a informação em dados que façam sentido, bem como eliminando a existência de duplicações. (Abed Alhameed Alkhatib & Singh Baicher, 2012)
Data fusion	É descrito quando os nós de uma *WSN*, efetuam novo processamento nos dados já agregados para produzir informação ainda mais precisa. (Abed Alhameed Alkhatib & Singh Baicher, 2012)
Event-driven	É um paradigma da programação, onde as operações da aplicação são determinadas por eventos. Estes eventos podem ser valores medidos em sensores e mensagens recebidas. (Rodriguez, 2018)
Multithreading	Permite que uma unidade central de processamento, tenha múltiplas *threads* de execução independentes, contudo, utilizam os mesmos recursos. As *threads* por sua vez, mantém a informação relevante para a sua execução como prioridade, controladores de exceções, registos de CPU e estado. (Multithreading, s.d.) (Multithreading (computer architecture), s.d.)

Multipath fading	É um efeito que ocorre sobre a propagação de dado sinal através de meio não guiado, tendo como consequência a diminuição da potencia do sinal até ao recetor. Chegando o sinal com uma potência de sinal mais baixa e com distorção. (Tranter, Taylor, Ziemer, Maxemchuk, & Mark, 2007)
Shadowing	É um efeito num dado sinal, onde o mesmo é refletido devido à superfície de objetos que se encontram na linha de transmissão entre transmissor e recetor. (Linnartz, 1995)
TCP	É protocolo de controlo de transmissão, orientado à ligação, o que representa necessitar de pré-estabelecer uma ligação entre o emissor e recetor. Disponibiliza mecanismos para verificação de erros, correção de erros e qualidade de serviço. Utiliza *Three Way Handshake*, como forma de estabelecer a ligação para a posterior transmissão de dados. (Marques da Silva, 2016)
UDP	É um protocolo simples, não orientado à ligação, que significa não necessitar de estabelecer ligação prévia para iniciar a transmissão de dados. Como não estabelece uma ligação prévia e não tem mecanismos de controlo, não é garantido que a informação enviada tenha sido recebida. (Marques da Silva, 2016)
Overhead	Nos termos referentes a redes, o excesso é a informação extra necessária para a transmissão de um pacote, que é derivada do aumento do tamanho dos cabeçalhos do pacote.
REST	É um estilo arquitetural de *software* que define a forma como deverão ser criados *web services*, que abrange a forma de implementação do cliente e do servidor. (What is REST?, s.d.) (Shelby & Bormann, 2009)
SOAP	É um protocolo de mensagens, que permite a troca de informação estruturada. Por *web service* permite que

	elementos distribuídos de determinada aplicação possam comunicar. Pode utilizar protocolos de baixo nível para a comunicação como HTTP. Normalmente, a estrutura da informação tem estrutura XML. (Shelby & Bormann, 2009) (Rouse, 2019)
Service-based	São web services que utilizam chamadas remotas para comunicação entre cliente e servidor. (Shelby & Bormann, 2009)
Resource-based	São web services que disponibilizam recursos através da invocação do URL. (Shelby & Bormann, 2009)
RPC	A chamada de procedimento remoto é um mecanismo de chamada de procedimentos, que visa permitir a transferência de controlo e dados entre diferentes espaços de endereçamento de memória. Utiliza o modelo cliente / servidor e consiste em três fases a ligação entre ambos. Na primeira fase o servidor, publicita os seus serviços, na segunda fase o cliente localiza o serviço no servidor de nomes e por último na terceira fase, estabelece a ligação do canal de transporte. (Caldeira, 2017)
Sink	É uma estação base numa rede *WSN*, que tem como objetivo coletar e processar dados. (Xu, Sun, & Xiao, 2018)
Cooja	É um simulador quem tem como fim disponibilizar um ambiente de simulação de dispositivos *IoT*, com diversos tipos de comunicação, tendo por base o sistema operativo *Contiki* pertencente à empresa. (The Contiki Operating System, s.d.)
Zolertia	É uma empresa especializada no desenvolvimento de dispositivos de baixo consumo para múltiplos propósitos. (Zolertia, s.d.)
Z1	É um tipo de dispositivo existente no mercado e no simulador *cooja*, pertencente à empresa *Zolertia*. (Lignan, 2015)

Java	É uma plataforma que permite desenvolver aplicações, sendo uma das principais linguagens de programação existentes hoje em dia. Permite que as aplicações desenvolvidas sejam executadas em ambiente virtual ao contrário de outros tipos de linguagem. (What is Java technology and why do I need it?, s.d.)
Maven	É um gestor de conteúdos que pode ser incorporado em projetos *Java*, permitindo que seja possível gerir as dependências e módulos presentes no projeto. (Introduction, s.d.)
Framework	É uma camada de estrutura do projeto que permite servir como suporte ou abstração de código para estender o conteúdo do projeto. Desta forma oferece funcionalidades genéricas como acontece com bibliotecas, no entanto difere no sentido de a mesma influenciar a estrutura do projeto. (Framework, 2019)
Mysql	É um sistema de gestão de base de dados relacionais. Faz uso da linguagem *SQL* como interface de operações de consulta, alteração ou eliminação de dados. (MySQL, 2019)
Hibernate	É um *framework* que permite o mapeamento de relação-objeto, que utiliza a linguagem Java, permitindo diminuir e abstrair a complexidade de implementação de gestão de operações de base de dados. (Hibernate, 2019)
Primefaces	É uma *framework* composta por componentes gráficos baseada em *JavaServer Faces*. (Primefaces, 2019)

Introdução

O projeto escolhido para investigar e apresentar um protótipo, tem como tema "Sistema de suporte à Previsão e Deteção de Fogos florestais".

8.1 Motivação

Com a realização deste projeto pretendo dar resposta ao problema existente nas florestas, com recurso à utilização de tecnologia para benefício da natureza e população que vive junto às áreas florestais. Todos os anos, os incêndios fustigam as florestas por todo o mundo, sendo que Portugal faz parte deste paradigma, onde se perde uma fonte de oxigénio, fauna e habitat de espécies de plantas e árvores únicas. Fontes como mão criminosa, utilização de queimadas na agricultura e descuidos são fontes de incêndios de grandes proporções, é necessário que exista uma deteção e resposta rápida na sua fase inicial, para que se evite consequências mais graves. Com a ajuda de sistemas de deteção de incêndio, como dispositivos de *internet of things*, abre a possibilidade de serem recolhidos dados com um intervalo mínimo de atraso entre a leitura e posterior monitorização e deteção. Este tipo de dispositivos, são bastante flexíveis em termos de compatibilidade e versatilidade com diferentes tipos de sensores, permitindo colocar sensores de temperatura, humidade, localização, concentração de gases, luminosidade, bem como outros diversos que se mostrem interessantes do ponto de vista do âmbito em que se encontram. Os seus custos são reduzidos, da perspetiva do custo unitário de cada dispositivo. Observando o paradigma verifica-se que já vários países adotaram esta estratégia ou a efetuar estudos no sentido da aplicação do sistema de *WSN* para deteção de fogo, torna-se uma motivação de que é um dos caminhos a seguir, por exemplo em Portugal. A apresentação de um projeto de exemplo para a aplicação em determinadas zonas de reserva natural, poderá ajudar a que mostre como sendo um caminho a seguir para os governantes.

8.2 Contribuições

O projeto tem como objetivo, apresentar soluções que ajudem no campo da prevenção e deteção de fogo nas florestas, visto tal, ser um dos problemas que mais tem assolado o país, e não só, dando como exemplo os Estados Unidos da América e Austrália recentemente, onde milhares perderam as suas casas. Quando chegam as épocas de temperaturas elevadas, normalmente devido em grande parte a mão criminosa ou a

descuidos do homem, desperdiça-se com os fogos florestais uma grande quantidade de recursos no seu combate, bens, matéria-prima e principalmente grande parte dos seres vivos que compõe as florestas, que são perdas irrecuperáveis.

8.3 Problemas

Um sistema constituído por múltiplos dispositivos, implica uma grande complexidade na sua conceção e construção, principalmente devido à irregularidade dos terrenos, onde podemos ter áreas planas e de grande concentração de árvores como declives de elevado grau de inclinação. Acrescenta-se ainda, as dificuldades criadas pelas condições climatéricas, que pode ter um cenário de grande intensidade de precipitação e vento, como alturas de elevada temperatura e seca. Estas variações implicam que o sistema de comunicação dos dispositivos se adapte conforme a necessidade em tempo real, necessitando de um sistema que suporte a comunicação em qualquer destas circunstâncias sem que ocorra indisponibilidade do sistema, ou que na eventualidade de existir, seja por pequenos períodos.

Como a comunicação será efetuada sem fios, sem ser num meio guiado, significa que está vulnerável a grandes perdas e fenómenos de distorção na transmissão do sinal, que tem consequências na comunicação com os outros dispositivos circundantes, implicando assim, que estas comunicações sejam feitas utilizando protocolos que sejam compatíveis com a comunicação não orientada à ligação, visto que podem estar ligados e a qualquer momento a ligação ser interrompida.

Para além disso é necessário ainda ter-se em conta, que as comunicações não tenham um custo elevado a nível de processamento dos dados que consequentemente provoca um elevado gasto de bateria, visto tratar-se de dispositivos que tem por base a alimentação através de baterias de pequenas proporções. Tal dificuldade, implica que os dispositivos apenas transfiram os dados exclusivamente necessários, que é onde se coloca a necessidade de um *middleware*, ou por outras palavras de um sistema que dê suporte à rede, que irá ser explicado mais à frente, mas contudo também implica escolher quais os dados a recolher pelos dispositivos e que permitam que através dos mesmos seja possível detetar que existe fogo ou grande probabilidade de ocorrer, permitindo assim ajustar os meios de prevenção com maior facilidade.

Outra dificuldade, é como obter energia para além da que se encontra na bateria dos dispositivos, que nos caso dos *drones* torna-se fácil através da criação de uma base

de carregamento onde os mesmos quando chegam a determinada percentagem de bateria, retornam à estação de carregamento e substituídos por outros, já no caso de dispositivos *IoT*, torna-se mais difícil, pois são dispositivos que não são móveis.

9 Conceitos

9.1 Webservices

Os *web services* são definidos pelo *W3C* como um software desenhado para comunicações pela rede, entre cliente e servidor. Utilizam o protocolo *HTTP*, por forma a comunicar de forma mais cómoda, no entanto podem utilizar *HTTPS* para ligações seguras. Existem dois tipos de *web services,* os *service-based* chamado *SOAP* e *resource-based* chamado *REST*.

Os *web services service-based* utilizam o formato *XML e SOAP* para disponibilizar chamadas de procedimento remotas, mais conhecidas por *RPC,* entre o cliente e servidor *SOAP*. As mensagens e sequências *SOAP* trocadas entre ambos, são descritas através da linguagem *WSDL*. A interface *SOAP* é designada através de *URL HTTP*, que implementa inúmeros métodos *RPC*, como por exemplo http://mote12.brm.com/dados?wsdl .

O paradigma *REST* ao invés do *SOAP*, mapeia objetos em recursos *HTTP*, em que cada *URL*, permite o acesso aos métodos disponibilizados. Estas interfaces podem ser descritas através de *WADL*, no entanto após a disponibilização do *WSDL 2.0*, interfaces baseadas em *REST*, podem ser definidas da mesma forma como as *SOAP*. O conteúdo das mensagens *REST HTTP* podem ser qualquer conteúdo *MIME,* apesar de o formato *XML* ser o mais habitual nas aplicações máquina a máquina. No entanto, devido ao tamanho que o formato *XML* ocupa, não é apropriado para utilização em *LoWPAN*, para isso foi criado um formato mais eficiente chamado *EXI*. Segundo o formato de implementação *REST*, os objetos estão acessíveis, através dos métodos padrão *HTTP GET, POST, PUT* e *DELETE*. O *GET* é utilizado para solicitar determinado(s) valore(s) e o *POST* para enviar determinado valor como parâmetro. O *URL* de exemplo para a interface *REST* é http://mote12.brm.com/sensors/temperature .

Para *LoWPAN*, que façam uso do protocolo *6LoWPAN* existem duas formas de utilização de *web services* que é a compressão ponto-a-ponto ou a compressão por meio de um *proxy*. Desta forma, permite que o tamanho exigido na troca de mensagens com recurso a *web services* seja suportado pelo tamanho máximo dos pacotes do protocolo *6LoWPAN*. Na imagem a seguir é possível a visualização do fluxo nos modelos *IP* e *6LoWPAN*. (Figueroa, Pérez, & Amezcua, 2017) (Shelby & Bormann, 2009)

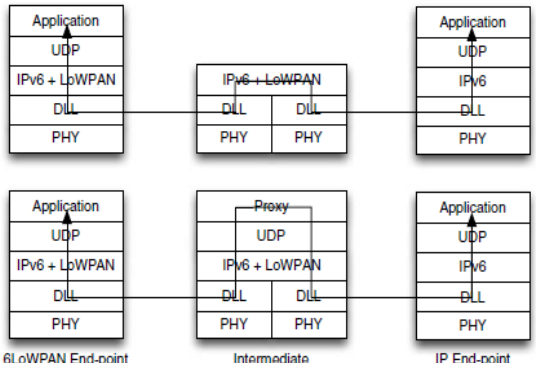

Figura 1 – Duas formas de compressão do pacote baseado em REST. (Figueroa, Pérez, & Amezcua, 2017)

9.2 Wireless sensor network

Uma *WSN* é uma rede constituída por dispositivos, chamados nós, podendo estes estarem estacionários ou a mover-se, são compostos por sensores, que com recurso aos mesmos permite a recolha de dados do ambiente onde se encontram. Posteriormente, essa informação pode ser comunicada para fora da zona, através das ligações entre os dispositivos que a constituem. Isto é conseguido, através de múltiplos saltos até um *sink*, que é nada mais, nada menos do que uma *gateway* que divide a rede de sensores e outras redes externas. Os nós que formam a rede, podem ser homogéneos ou heterogéneos conforme a necessidade e ainda podem ter conhecimento da sua localização ou não, se capacitados com um sensor para tal.

A forma tradicional de uma rede *WSN* é composta por múltiplos dispositivos e uma *Sink*, o que cria problemas de escalabilidade devido a ter apenas um *Sink*. Atualmente, o cenário mais realista é ser composto por múltiplos dispositivos e múltiplos *Sink*, dado a maior densidade da rede, a probabilidade de existirem problemas de propagação de informação devido à falta de alcance do sinal devido à falha de dispositivos, é substancialmente menor ou inexistente. Ao contrário do modelo tradicional, o atual permite que a escalabilidade da rede seja bastante elevada, visto que os dispositivos podem selecionar um dos *Sink* que melhor se adeque conforme as métricas utilizadas pelo protocolo utilizado para o cálculo do encaminhamento. Isto significa, que pode ser possível atingir a máxima largura de banda e o mínimo de saltos de encaminhamento, no entanto pode trazer desvantagens como a necessidade de implementação de protocolos mais complexos para albergar a quantidade de dispositivos.

As redes *WSN* podem ser utilizadas para múltiplas aplicações, como monitorização de determinado ambiente e logística em armazéns. Em função dos requisitos da aplicação é necessário efetuar, a escolha correta da tecnologia que melhor se enquadra. Para tal, é necessário que se conheça as várias tecnologias de transmissão sem fios. Existem dois tipos de aplicações de acordo com o tipo de dados a recolher. Podem ser classificadas como deteção de eventos e processo de estimação espacial.

9.2.1 Deteção de eventos

Neste tipo de aplicações os dispositivos são distribuídos para detetar determinado evento, como por exemplo a ocorrência de fogo numa floresta, que é o âmbito do projeto. Os dados são processados e se passa determinado valor definido a informação segue para os dispositivos *Sink*. O processo de deteção pode ser efetuado de forma distribuída, isto é, tanto o *Sink* como os nós, coordenam-se de forma a identificar o evento.

9.2.2 Processo de estimação espacial

A aplicação consiste em estimar determinado evento físico, como por exemplo temperatura, que podem ser modeladas como um processo aleatório bidimensional. Mas para tal, existe o problema em obter a estimação de todo o processo espacial baseado em amostras recolhidas pelos nós em locais aleatórios. As medições recolhidas são posteriormente processadas, que será efetuado de forma distribuída nos dispositivos ou centralizada no *Sink*. O erro da estimação deve-se apenas à densidade dos nós, que quantos mais melhor, assim como à variância espacial do processo.

(Buratti, Conti, Dardari, & Verdone, 2009)

9.3 RPL

É um protocolo de encaminhamento desenhado para redes *6LoWPAN*, que tem como objetivo otimizar as operações, através da disponibilização de encaminhamento multiponto-a-ponto de nós *LoWPAN* para um nó de controlo central, denominado *Sink*, ou vice-versa caso a ligação seja bidirecional. O protocolo efetua o cálculo da melhor rota para a topologia utilizada que para a qual gera um gráfico, com base na unidade distância-vector que adapta as rotas para um destino, a isto chama-se *DODAG*. O gráfico é criado através do uso do *OF*, que define a forma como a métrica do encaminhamento é calculada e quais nós utilizar. Os nós *Sink* ou também apelidados de *DAG root*, são aqueles responsáveis por criar o gráfico topológico da rede. É conseguido através do *broadcast*

do seu *ID*, nível e outras informações importantes através do *DIO*, assim que os nós intermédios recebam a mensagem *DIO*, respondem para o nó *Sink*, que após efetuar o cálculo e atualização do seu nível de parentesco para com os restantes, responde aos nós vizinhos com as informações dos níveis que deverão corresponder. Este cálculo e topologia acabará por ser semelhante à de uma folha, posteriormente esses nós vão efetuar o mesmo processo com os vizinhos e assim sucessivamente até que alcance o último nível e não exista mais.

Caso um nó receba mais do que uma mensagem *DIO* de vizinhos, terá que escolher qual o seu nó de parentesco, nível entre outras métricas. Outra situação possível, é caso um nó não receba mensagem *DIO*, durante o tempo disponível para integrar, efetua a solicitação aos nós vizinhos, através do envio de uma mensagem *DIS*. (Glissa & Meddeb, 2018)

9.4 6LoWPAN

O *6LoWPAN* é um padrão de código aberto, que é uma rede simples de baixo custo que proporciona conectividade através de wireless em aplicações com limitada fonte de energia e requerimentos de largura de banda flexíveis. Consiste na utilização do *IPv6* utilizando o protocolo *TCP/IP* em redes *WSN*, sobre o padrão *IEEE 802.15.4* que é caracterizado pelo baixo alcance, baixa largura de banda, baixo consumo, baixa utilização de memória e baixo custo, utiliza a faixa de frequência dos 2.4 GHz adaptada e outras como a *1 GHz low-power*, *RF*, *Bluetooth Smart*, *PLC e low-power WI-FI*. Por forma a permitir ter capacidades de comunicação IP, faz utilização de uma camada de adaptação para a fragmentação e remontagem do pacote. (Olsson, 2014) (Ee, Ng, Noordin, & Ali, 2010)

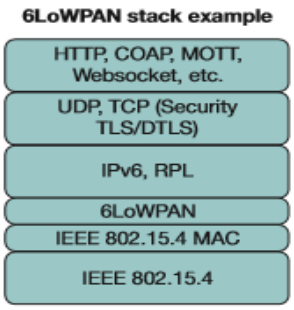

Figura 2 - Pilha de camdas 6LoWPAN (Olsson, 2014)

Quando um dispositivo de baixa capacidade de processamento, chamado *RFD*, necessita de enviar um pacote de dados para um dispositivo *IP* fora da rede *6LoWPAN*, inicialmente envia o pacote para o dispositivo da rede com maiores capacidades de processamento, chamado de *FFD*. O *FFD* irá agir como se de um router se tratasse na rede *6LoWPAN*, recebendo o pacote e encaminhando até à *gateway* da rede usando o protocolo *IPv6*. Posteriormente, o pacote é enviado até ao destino por *IP*, usando o protocolo que melhor se adequa.

10 Investigação

10.1 Estudos e projetos realizados

Desde que surgiu o paradigma dos dispositivos de IoT, tem surgido vários projetos para a deteção de fogos florestais, no entanto, todos tem tido como problema o investimento necessário e o retorno do mesmo, contudo dever-se-ia ter realmente em conta as vidas que se podem ajudar a salvar e a fauna que nela habitam.

10.1.1 IoT for detecting wildfires

Uma notícia com o título "IoT for detecting wildfires", descreve na primeira pessoa, que o fogo propaga-se de forma volátil, não oferecendo chance de chamar os meios responsáveis pelo combate e alertar as proximidades, sendo a deteção antecipada um ponto fulcral para se poder ativar todos os meios necessários e impedir que o seu resultado seja mais catastrófico. Um sistema como o retratado neste projeto, tem como principal preocupação antecipar o envio de meios quando está prestes a iniciar-se ou se já iniciou o incêndio. Para a sua deteção, explica que se deverá ter em conta os seguintes fatores: leitura de valores de CO2, que quando estes aumentam de forma rápida indicia que deverá haver fogo; através do uso de infravermelhos ou ultravioleta que permite identificar focos de temperatura; através dos valores de temperatura, que onde o seu aumento rápido poderá significar que existe o indicio de incêndio ou que este poderá surgir a qualquer momento. Acrescenta ainda, que no seu projeto/investigação concluíram que necessitavam de dispositivos compostos por sensores infravermelhos, ultravioletas, temperatura e CO2 a cada 50 m2. (Martínez, 2017)

10.1.2 Detecting Forest Fires using Wireless Sensor Networks

Num artigo com o título "Detecting Forest Fires using Wireless Sensor Networks", fala sobre um projeto que se encontra aplicado no terreno, mais precisamente no Norte de Espanha, chamado "SISVIA - Vigilancia y Seguimiento Ambiental", que através de dispositivos compostos por sensores de temperatura, humidade, monóxido de carbono e dióxido de carbono conseguem detetar a presença de incêndios. Necessitaram de 90 dispositivos, proprietários de uma empresa, para cobrir uma área de 210 hectares, a arquitetura que compõe este sistema é, uma rede de sem fios de sensores (*WSN*), uma rede de comunicações e uma central de receção. O sistema contém uma aplicação que atua como uma espécie de *Middleware*, onde permite a gestão dos vários sensores

individual ou global, apresenta os valores dos sensores pelo mapa das áreas onde se encontram e caso os valores despoletem o sistema de alarme este comunica com os bombeiros para os encaminhar para o local, através das coordenadas de *GPS*. Cada sensor é alimentado através de baterias recarregáveis e de painéis solares. (Solobera, 2010)

Um outro artigo fala sobre o potencial de um *Middleware* numa rede destas, permitindo este tratar de tudo relacionado com o encaminhamento e troca de informação entre dispositivos heterogéneos, que é o caso abordado, pois teremos dispositivos *IoT* e *Drones*. Desta forma, é responsável por assegurar uma maior eficiência e efetividade de todos os dispositivos, permitindo a escolha do melhor protocolo de rede, facilidade na configuração das dezenas ou mais de dispositivos, na gestão da bateria através da implementação e gestão de rotinas de funcionamento e controlo e gestão de tráfego na rede assegurando o melhor *QoS*. (Wang, Cao, Li, & K. Dasi, 2008)

10.1.3 Early forest fire detection and verification using optical smoke, gas and microwave sensors

Um *paper* realizado em 2012, sobre deteção antecipada de fogos florestais, abordou a utilização de sensores de fumo, gás e micro-ondas como forma de prossuposta vigilância. Na sua opinião, existe dois níveis de vigilância, que depende do tamanho e presença humana. Caso o risco seja elevado, deverá ser vigiada por vigilantes, já áreas de grandes proporções e com pouco risco deverão ser vigiadas por satélites e meios aéreos. Esta forma de vigilância, é realizada na parte ocidental da Alemanha, sendo composto por torres equipadas com sistemas compostos por camaras, em que as imagens são transmitidas para um centro de controlo e analisadas por um software especializado, que caso detete é inicializado um alerta para os meios de combate. Os falsos alarmes, são um problema que requerem medidas adicionais de verificação, um exemplo que provoca falsos alarmes é o nevoeiro e o pólen. Para identificar a autenticidade do alarme de deteção de fogo, é enviado para o local um *UAV*, equipado com sensores de gás e camara de infravermelhos controlado remotamente, com o intuito de se obter imagens detalhadas e mais dados. Com base nestes novos dados, são posteriormente despoletados os meios necessários ou confirmado de que se trata de um falso alarme. Para se obter uma boa taxa de confiança na deteção de incêndios, é utilizada a combinação de camaras de infravermelhos, medidores de micro-ondas e sensores adicionais de partículas de fumo montados em *UAV* ou balões de ar.

Como plataforma de suporte à verificação de incêndio utilizam um dispositivo autónomo com conhecimento da cartografia do terreno, como um *drone*. Desta forma, o *UAV* permite oferecer à brigada de combate a incêndio todas as informações relativas ao mesmo, em todas as condições mesmo à noite quando não existe suporte aéreo. O modelo sugerido é um *AirRobot AR100-B* que é um modelo de baixo custo comparado com o custo de helicóptero, tem um diâmetro de 1 metro e o peso de 1 quilograma e 4 motores de baixo ruído, no entanto apenas tem uma autonomia de aproximadamente 25 min. Atinge velocidades de 36 km/h e suporta ventos até 28km/h. A transmissão de dados e controlo é executado em tempo real através de dispositivos RF.

Para a vigilância de um ponto quente, é proposta a utilização de um dirigível com vários sensores e camara térmica. Devido ao seu tamanho de 9 metros com 2,3 metros de diâmetro, faz com que tenha um peso de 7 quilogramas, incluindo as baterias. Um *embedded-PC* controla o dirigível, assim com o trata da comunicação e transferência de dados para a estação base. A estação base consiste numa interface utilizador (Comm-PC) para transmitir e receber dados de voo através de *WLAN*. Os dados são recebidos através da utilização do modelo *TCP/IP*, em que os dados recebidos são combinados com os dados do *GPS* e instantes de tempo para a visualização e utilização em algoritmos.

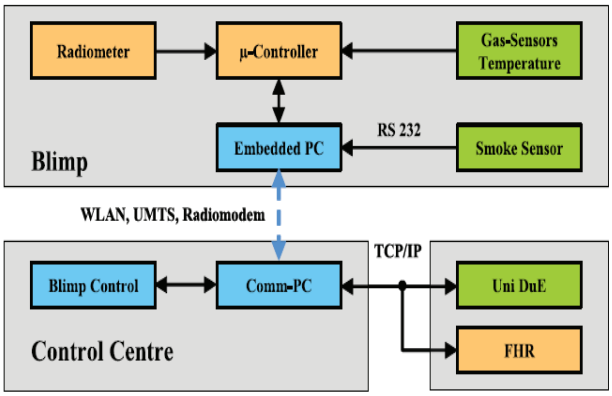

Figura 3- Estrutura de Comunicações de um dirigível (Krull, Tobera, Willms, Essen, & von Wahl, Early forest fire detection and verification using optical smoke, gas and microwave sensors, 2012)

Os sensores adicionais são usados para ajudar em situações ambíguas, para tal é requerido que estes sejam imunes a grande parte de distúrbios causados por nevoeiro, pó e condensação de água. No caso dos detetores de gases, o fumo proveniente dos incêndios é analisado por diferentes semicondutores, protegido por uma capa de metal que protege os sensores contra o pó e outros tipos de distúrbios, por forma a que não afetem as leituras. Este tipo de sensores, estão obrigados a cumprir um leque específico de requerimentos comparados com os aplicados a outro de tipo finalidades. A alta sensibilidade dos sensores, deve permitir que mesmo em pequenas concentrações de fumo, mesmo sendo bastante disperso e sob efeito da turbulência causada pelo vento, o mesmo seja detetado.

Devido à grande ocorrência de libertação de hidrogénio durante o início do fogo, é utilizado um sensor H_2 $[0 - 10ppm]$; um sensor de hidrocarbonetos C_xH_x [0 – 5 ppm], visto serem sensíveis aos gases originados por produtos orgânicos; um sensor de temperatura devido às rápidas variações. Faz ainda uso de um sistema de alta sensibilidade de deteção de fumo, para equipar o dirigível nas alturas de monitorização de fogos extinguidos.

Para a deteção de incêndios, foram estabelecidas três fases: fase de ignição, fase de fogo latente e fogo ativo. A fase de ignição é onde existe apenas fumo invisível e gases, que é detetado por sensores de gás e sistemas de aspiração; na fase de fogo latente é possível observar-se a existência de fumo sendo detetado por sensores de gás e detetores óticos de fumo; na fase de fogo ativo é quando o fumo é bastante espesso e as temperaturas são elevadas. Nos testes realizados no laboratório de *Duisburg Fire*, verificaram que conforme a localização a velocidade de ignição é diferente principalmente devido à composição da floresta, no entanto o objetivo era verificar em cada fase quais os tipos de sensores que melhor se adequavam, que foram os indicados anteriormente. No entanto, também se verificou que as hélices do *UAV* não tem influência nos sensores em relação aos valores medidos.

Nos testes efetuados em campo, a 15 metros do solo, conseguiu identificar o centro do incêndio, dentro de uma coluna de fumo. Verificou-se ainda ser possível através do uso do dirigível monitorizar uma área onde o fogo já foi extinto, pois apesar de extinto, as árvores queimadas vão libertando gases, oriundo do incêndio e vão reacendendo pontos o que torna difícil a sua correta deteção. (Krull, Tobera, Willms, Essen, & von Wahl, Early forest fire detection and verification using optical smoke, gas and microwave sensors, 2012)

10.1.4 Forest Disaster management with Wireless Sensor Network

Este artigo aborda a gestão de uma rede *WSN*, começando por explicar a classificação dos sensores utilizados numa rede destas para detetar fogos florestais.

Como propriedades físicas de leitura temos a temperatura, pressão, aceleração, fluxo, posição e luminosidade. Para cada uma destas, existe um ou mais sensores possíveis de efetuar a medição e, em cada um deles existe uma forma de *output*.

A medição de temperatura pode ser conseguida através de:

- **Termopar**, que traduz a leitura em voltagem, unidade *Volts* (v);
- **Silício**, é um mineral que permite a leitura em voltagem e corrente;
- **Termo resistência**, traduz a leitura em sinal analógico, medido em resistência;
- **Termístor**, traduz a leitura em resistência, unidade *Ohms* (Ω);

A força/pressão pode ser medida através de:

- **Extensómetro**, é um aparelho que traduz a leitura efetuada em valores de resistência;
- **Piezoeletricidade**, que são cristais que permitem a leitura através de voltagem;

Para a medição da aceleração é utilizado:

- **Acelerómetro**, é um dispositivo que permite medir a propriedade indicada, convertendo-a em capacitância, que utiliza a unidade *Farad* (F) para a medir.

No caso da propriedade de medição fluxo, a mesma pode ser obtida através de:

- **Transdutor**, é um dispositivo que permite converter o valor do fluxo em sinal analógico, que neste caso é através de voltagem;
- **Transmissor de pressão**, é um dispositivo que permite a conversão do valor medido em voltagem;

O cálculo da posição é conseguido através de um único sensor, sendo este:

- **Sensor *LVDT***, é um sensor composto por uma bobina principal que induz tensão nas secundárias, originando uma corrente *AC*; (Sensores de deslocamento (MEC113), s.d.)

Por último, a luminosidade pode ser medida através:

- **Sensor Fotodíodo**, permite medir a intensidade de luz, convertendo em corrente como forma de unidade de medição. (Photodiode Working Principle, Characteristics and Applications, s.d.)

Como uma rede de sensores sem fios é focada no baixo consumo de energia, são apresentados pelo escritor do artigo sete possíveis standards que definem as funções e protocolos dos sensores da rede. Os standards abordados são *IEEE 802.15.4*, *ZigBee*, *Wireless HART*, *6LoWPAN*, *IEEE802.15.3*, *ISA100.11a e Wibree*.

O *IEEE 802.15.4*, é focado no baixo custo e no baixo consumo em *LRWPAN*, utiliza uma topologia em estrela e *P2P*, para comunicação entre os dispositivos da rede. Suporta os protocolos da camada física e de ligação de dados. Foi desenhado principalmente para ser aplicado a redes de sensores, onde as comunicações realizadas são feitas a curta distância por forma a reduzir o consumo. A sua camada física suporta frequências entre 868/915 MHz até 2.4 GHz. A camada *MAC* usa *CSMA/CA* para controlar o acesso ao meio de rádio.

O *ZigBee* foi construído sobre o standard *IEEE 802.15.4*, permitindo ter perdas mínimas e usar a tecnologia de gasto de energia reduzido nas comunicações para as aplicações *ZigBee*. Utiliza uma tipologia em malha que permite a ligação entre um grande volume de dispositivos. Nas redes *ZigBee* existem três tipos de dispositivos:

- *ZigBee Coordinator* – Inicializa a formação da rede, armazenando informação na rede e que possibilita a ligação entre redes.
- *ZigBee Router* – Oferece comunicação *multihop* entre os dispositivos.
- *ZigBee end device* – São os dispositivos finais, podendo estes ser sensores ou atuadores que comunicam diretamente com o *ZigBee router* ou *ZigBee Coordinator*.

O padrão *WirelessHART* disponibiliza um protocolo de comunicação de rede sem fios para medição de processos e controlo de aplicações. É seguro, fiável e energeticamente eficiente, desenhado para suportar unicamente ou em simultâneo, tipologias em estrela e em malha. A segurança é garantida através do uso de algoritmos criptográficos, autenticação, verificação e gestão de chaves. Mais uma vez, é baseado no standard *IEEE 802.15.4* e utiliza a banda de frequência de 2.4 GHz para operação.

O standard *6LoWPAN*, utiliza pacotes *IPv6* para comunicação sobre o standard *IEEE 802.15.4*. As principais características são o tamanho reduzido dos pacotes, a pequena largura de banda disponibilizada e o baixo custo. Utiliza tipologias de estrela e malha para a comunicação entre dispositivos. Desta forma permite utilizar a infraestrutura existente e oferecer a possibilidade de utilização de novas técnicas como reconhecimento de localização e disponibilizar mais espaço de endereçamento vital neste tipo de redes.

O standard *IEEE 802.15.3* é desenhado para taxas elevadas de dados, suporta a transmissão de vídeo em tempo real e música. Opera na banda de frequência dos 2.4 GHz e tem uma taxa de transmissão entre 11 *Mbps* e 55 *Mbps*, utilizando a técnica de

transmissão *TDMA*, de modo a oferecer *QoS*. Este standard é utilizado em dispositivos como colunas wireless, consolas, impressoras e televisões.

O *ISA100.11a* foi desenvolvido pela *ISA*, criado para ser flexível, suportar múltiplos protocolos, usar *standards* abertos, suportar múltiplas aplicações, fiabilidade, segurança e controlo. Define a pilha de protocolo, gestão de sistema e funções de segurança em dispositivos de baixo consumo e redes de baixa taxa de transmissão. É composto por *field device-Routing, field device-Non Routing, Backbone router, gateway, System Manager e System Security*.

Webree, utiliza a tecnologia *Bluetooth* e opera na faixa de frequência de 2.4 *GHz* e tem como funcionalidade poder saltar rapidamente de bandas dentro dessa mesma faixa de operação, por forma a oferecer segurança e resistência a interferências causadas por outros sinais *Wibree*. O seu desempenho é como o do *Bluetooth*, no entanto utiliza menos energia, permitindo obter-se uma maior duração de bateria dos dispositivos. Os chips do *Webree* são também mais pequenos do que os dos *Bluetooth*.

O modelo proposto no artigo, consiste em recolher a informação dos vários sensores, sendo que se os dados se desviarem dos parâmetros normais, é despoletado um sinal de emergência que é enviado através da estação base e posteriormente *backbone*, até ao utilizador final. Caso a *WSN* esteja fora do campo de comunicação da estação base, esta última é constituída por um microcontrolador para comparar os valores predefinidos e os últimos medidos, permitindo esta também despoletar o alarme.

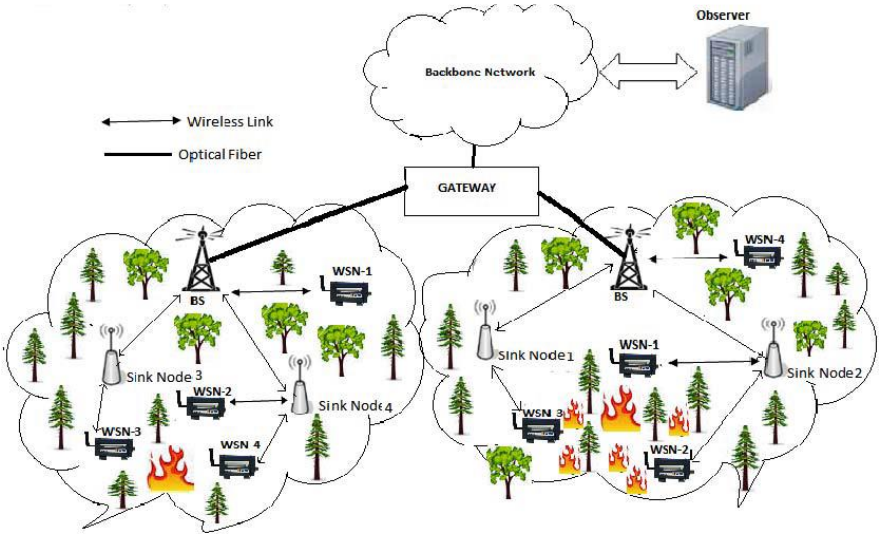

Figura 4- Arquitetura de gestão de desastre numa floresta (Bhosle & M. Gavhane, 2016)

São apresentados dois casos de exemplo:

- No primeiro, quando um incêndio é detetado pela *WSN 3 e WSN 4* na área 1, a informação recolhida é enviada para a estação base mais próxima, que é verificada e caso o volume de dados seja inferior ao predefinido envia um sinal de alerta ao utilizador final;
- No segundo exemplo, quando o incêndio é detetado pela *WSN 1, WSN 2 e WSN 3* na área 2, estes nós posteriormente, enviam os dados recolhidos para a estação base mais próxima, se o volume de dados for acima do normal valor, é enviada uma mensagem de alerta até ao utilizador final.

(Bhosle & M. Gavhane, 2016)

10.1.5 Wireless Sensor Network for Forest Fire Detection

A *WSN* é apresentada como um processo de medição, computação e comunicação que disponibiliza recursos administrativos, de observação e tratamento de eventos e fenómenos a um dispositivo. A utilização de *WSN* é mais eficiente do que usar cabos, portanto a tecnologia das redes *WSN*, oferece condições para que sejam conduzidas experiências no meio ambiente. A título de exemplo, os biologistas necessitam de monitorizar os animais no habitat, os investigadores ambientais necessitam de um sistema de monitorização da poluição, entre outros casos, no entanto todos eles recaem na necessidade de um sistema de monitorização baseado em *WSN*. Os componentes de uma *WSN*, inclui sensores, módulos wireless e computador, todos estes componentes permitem formar um sistema de monitorização capaz de mostrar os dados recolhidos com base nas características dos sensores.

Tipo de sensores	Sensor
Temperatura	Termopar, Termístor
Pressão	Barómetro, Válvula de pressão, Válvula ionizante
Ótico	Foto díodos, Foto transístores, Sensores infravermelhos
Acústico	Microfones, Ressonadores piezoelétricos
Mecânico	Sensores tácteis, Diafragmas capacitivos, Células piezoresistivas
Posição	GPS, Sensores baseados em infravermelhos
Movimento e Vibração	Acelerómetro, Giroscópio, Foto-sensor
Humidade	Sensores capacitativos e resistivos, higrômetros, Sensores de humidade
Radiação	Detetores ionizantes

Tabela 1- Tabela de sensores

Na arquitetura *WSN*, cada nó do sistema *WSN*, normalmente é divido em alguns subsistemas, que são de sensoriamento, processamento, comunicação e energia. Como os combinar é o principal obstáculo, sendo o processador a parte mais importante do sistema *WSN* e que pode afetar drasticamente o desempenho e consumo de energia. Para o processador existem várias opções, que podem incluir a utilização de *Microcontroladores, Processador de sinal digital, Field programmable gate array e Application-specific IC*.

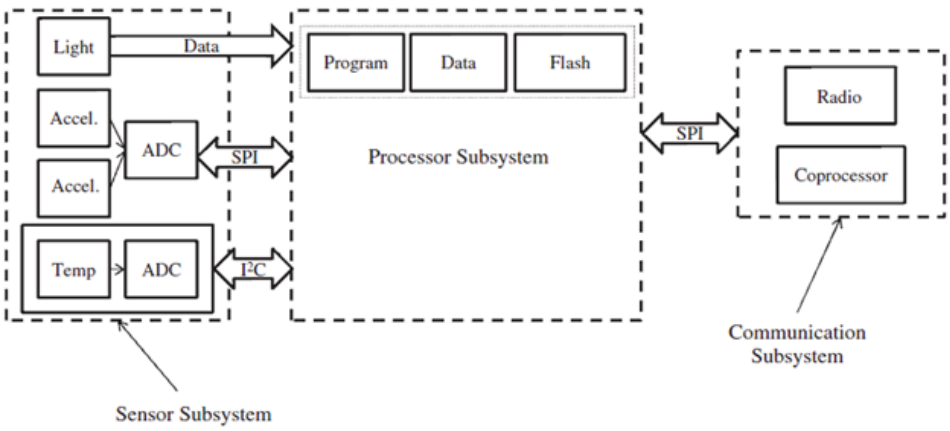

Figura 5- Arquitetura geral de WSN (Hariyawan, Gunawan, & Putra, 2013)

O sistema proposto pelo artigo é composto por:

10.1.5.1 Uvtron Flame Detector

É um sensor que permite detetar a presença de fogo, é capaz de detetar luz ultravioleta entre 185 e 260nm que é o intervalo da luz emitida pelo fogo. Estes sensores não são capazes de detetar o tamanho do fogo, no entanto conseguem detetar a mais de 5 metros um cigarro aceso. O sensor é ativado quando recebe uma voltagem de 350 Vdc, que pode ser obtido de um circuito C3704 no qual está integrado.

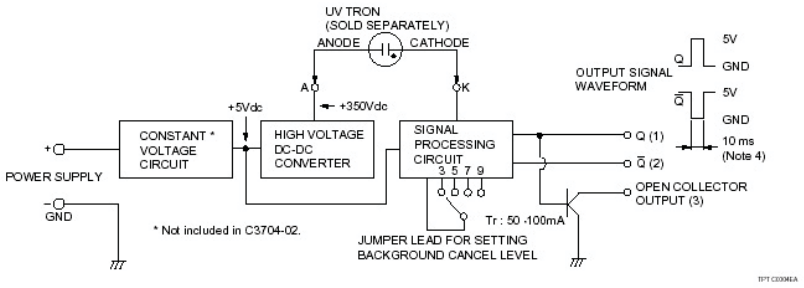

Figura 6- Circuito C3704 (Hariyawan, Gunawan, & Putra, 2013)

10.1.5.2 MQ-2 Gas Sensor

Este sensor é altamente sensível a substâncias de S_nO_2 , que na atmosfera é mau condutor. Quando ocorre a existência das substâncias no ar, são detetadas através da alta condutividade pelo sensor, que quanto maior for a concentração maior será a condutividade. Através do uso de um circuito elétrico, é possível a conversão da condutividade em sinal através da concentração do gás. Para além, da deteção da substância de S_nO_2, é altamente sensível a substâncias como GPL, Propano, Hidrogénio, Metano e vapores inflamáveis. O seu custo é relativamente baixo.

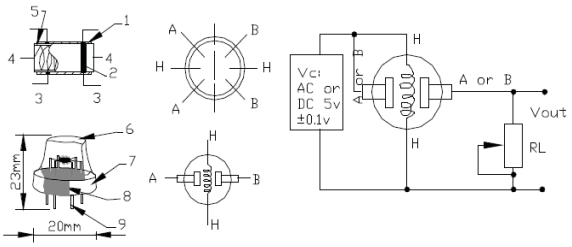

Figura 7- Sensor MQ-2 de gás (Hariyawan, Gunawan, & Putra, 2013)

10.1.5.3 LM35 Temperature Sensor

Este sensor, é um transdutor que tem como função converter temperatura em sinal analógico elétrico, que é medido em voltagem. Tem uma precisão elevada e um design

fácil de integrar comparado com outros semelhantes, para além disso, tem uma baixa impedância de saída e alta linearidade, permitindo que seja conectado a uma série de controlo especial sem grande necessidade de ajuste. A temperatura pode ser medida entre os 55ºC e os 150ºC, o que é bastante amplo.

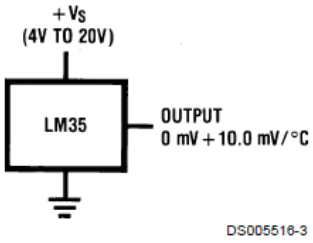

Figura 8- Circuito de um LM35 (Hariyawan, Gunawan, & Putra, 2013)

10.1.5.4 Node Sensor

O sistema de um nó de sensor consiste em 3 peças, sensores, microcontrolador e módulo RF. Para que esteja habilitado a enviar dados através de um meio sem fios, é necessário que o dispositivo tenha a capacidade de conversão de dados digitais em frequência numa portadora para ser enviado através de um meio não guiado. Um dos módulos RF sugeridos é o KYL 500S, que é um transceptor, com capacidade de transmitir e receber dados através de sem fios, com frequências de 433/868/915 MHz e taxa de transmissão 38,400 bps. O uso deste tipo de módulos é prático visto terem um tamanho pequeno, poderem ser ligados diretamente a uma porta RS232 e funcionar com potência entre os 3.3 e os 5 VDC. Tem um alcance entre 100 metros e 5 quilómetros e utiliza modulação GFSK de modo a possibilitar a utilização anti interferência e baixo BER.

Na sua pesquisa, para detetar fogos florestais efetuaram medições de temperatura, chamas, níveis de metano, gasolina, CO e CO_2 nos pontos quentes de um incêndio.

- Medição no simulador espacial sem queimar turfa e medição no simulador espacial queimando a turfa – O sistema de deteção utiliza três sensores para detetar se é o início do fogo ou não. Estes três sensores detetam fogo, fumo e alterações de temperatura. O sistema foi colocado num espaço fechado no simulador e medidos os vários valores das variáveis.

  O valor da temperatura foi de pico 34ºC com turfa e 31ºC sem turfa, enquanto os valores de metano sobem para uma média de 7 ppm com turfa e 1 ppm sem turfa.

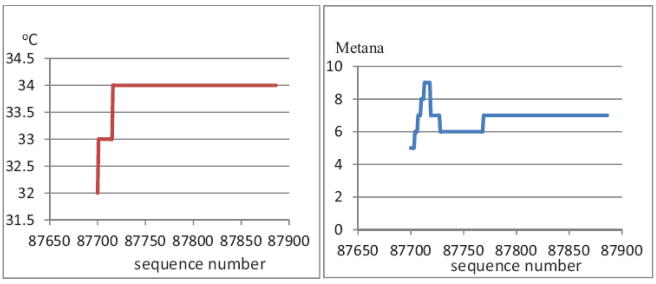

Figura 9- Medição da temperatura e metano (Hariyawan, Gunawan, & Putra, 2013)

Os resultados da medição de hidrocarbonetos e CO_2, concluíram que a concentração de hidrocarbonetos num ponto quente com turfa é de uma média de 95 ppm comparativamente aos 41 ppm sem turfa. Já quanto aos níveis de CO_2 com turfa obteve uma média de 103 ppm e sem turfa 97 ppm.

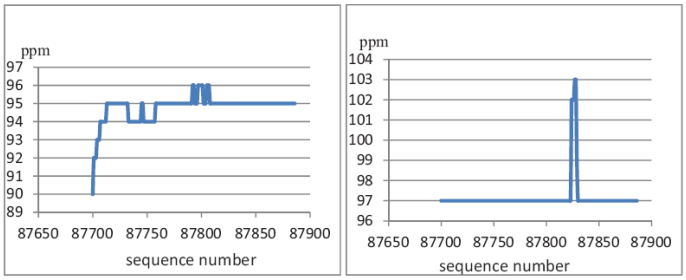

Figura 10- Concentração de hidrocarbonetos (Hariyawan, Gunawan, & Putra, 2013)

- Medição dos níveis de ar numa cidade – Medição da temperatura, níveis de metano, Hidrocarbonetos e CO_2.

  Os testes efetuados na cidade revelam que a temperatura média foi de 34ºC, enquanto que os níveis de metano tiveram um máximo de 4 ppm e mínimo de 1 ppm.

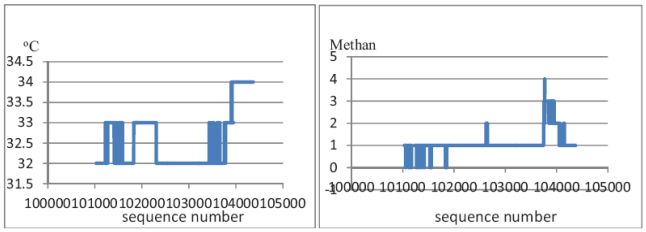

Figura 11- Níveis de temperatura e metano (Hariyawan, Gunawan, & Putra, 2013)

Os resultados das medições na cidade quanto aos níveis de hidrocarbonetos são superiores, tendo sido obtido concentrações de 80 ppm comparativamente com o resultado obtido anteriormente de 41 ppm. Já quanto aos níveis de CO_2, teve uma média de 97 ppm.

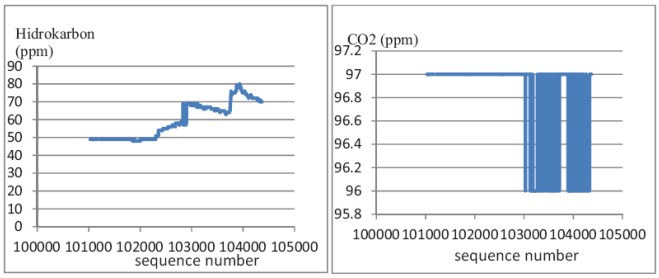

Figura 12- Medição de Hidrocarbonetos e CO2 (Hariyawan, Gunawan, & Putra, 2013)

- Teste do módulo RF – Para determinar o desempenho e propagação de sinal foi feito o teste, concluindo-se que quando maior a distância, menor é a potência recebida por cada nó ou por outro lado a qualidade dos dados recebidos é pior cada vez que a distância aumenta. Os dados recebidos pelo recetor, tem uma taxa de transmissão entre 1200 bps e 19200 bps, o que é baixa, mesmo com os níveis normais de ruído que rondam <80dBm. Apesar de tal nível, o módulo utilizado alcançou 123dBm de potência, permitindo receber dados apesar dos níveis de ruídos. A média de distância máxima de transmissão medida foi de 310 metros para 1200 bps, obtendo-se a 230 metros 9600 bps e a 90 metros 19200 bps. Estas medições permitiram concluir, que comparativamente com os valores anunciados em *LOS*, de alcance até 1000 metros, a realidade é bem diferente quando aplicado em campo.

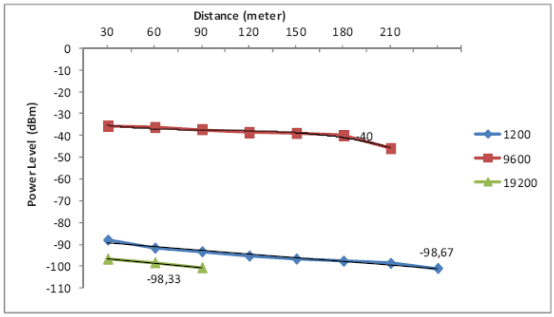

Figura 13- Potência medida em função de distância (Hariyawan, Gunawan, & Putra, 2013)

As medições efetuadas em relação ao *pathloss*, mostram que quanto mais longe se encontrar o recetor do emissor, mais alto será o *pathloss*, estes resultados vão ao encontro com as medições efetuadas aos níveis de potência com a variação da distância que foi apresentado antes.

Inclusive as medições efetuadas ao RSSI, confirmam que aumentando-se a distância entre recetor e emissor o valor irá diminuir, pois é inversamente proporcional ao *pathloss*. Isto significa, que a potência recebida baixa diretamente com a distância e o valor calculado para ser usado pelos sistemas é 9600 bps.

Foram feitas medições quanto à densidade de dispositivos, para se apurar qual o número de dispositivos é necessário para cobrir uma determinada área. Sabe-se que com o aumento da distância entre emissor e recetor, de forma inversamente proporcional, a densidade de potência diminuirá.

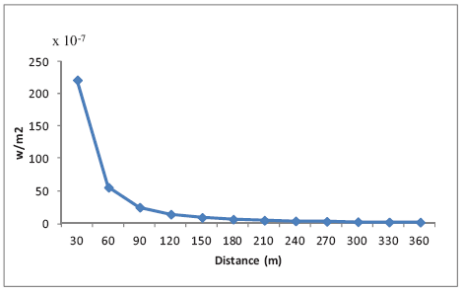

Figura 14- Densidade de Potência em função da distância (Hariyawan, Gunawan, & Putra, 2013)

Os escritores concluíram que com base nos níveis de gás, metano, CO e CO_2 podem ser usados para detetar de forma precoce os fogos florestais e a média de transmissão máxima medida numa área com zonas sombra pode ser observada na tabela a seguir.

Distância Rx / Tx (metros)	Velocidade de transmissão (bps)
310	1200
230	9600
90	19200

Tabela 2 - Velocidade de transmissão em função de distância

(Hariyawan, Gunawan, & Putra, 2013)

10.1.6 Detection and Verification of Potential Peat Fire Using Wireless Sensor Network and UAV

O artigo foi criado como uma solução para ser aplicado nas florestas tropicais da Indonésia. Devido ao grande volume de fogos florestais provocados pela área da agricultura, torna-se importante encontrar uma forma de mitigar o impacto, através da monitorização da floresta utilizando para isso, uma *WSN* composta por uma grande quantidade de sensores, que recolhem informação sobre temperatura, humidade luz e pressão atmosférica, transmitida para um ponto de monitorização ou departamento de proteção civil. Com os avanços neste tipo de redes e tecnologias envolvidas, permite a sua utilização, pois sistemas baseados em satélite são bastante suscetíveis às condições climatéricas e a outra solução que passa por utilizar seres humanos para observar, são bastante limitadas na deteção. É proposta assim a utilização de *WSN* para recolher dados em tempo real e a utilização de objetos aéreos não tripulados para obtenção de imagens de vídeo. As características para o início de um fogo, são ter material inflamável, oxigénio e altas temperaturas.

O conceito de modelo de arquitetura utilizada consiste numa *WSN* e *UAV*, sendo o sistema composto por três camadas:

- *Sensor network layer*, oferece monitorização do ambiente ao nível do solo.
- *UAV layer*, oferece monitorização de baixa altitude, através de vídeo vigilância.
- *Satellite layer*, oferece monitorização da superfície da terra em diferentes espetros de banda do vídeo, infravermelhos e frequências de radar.

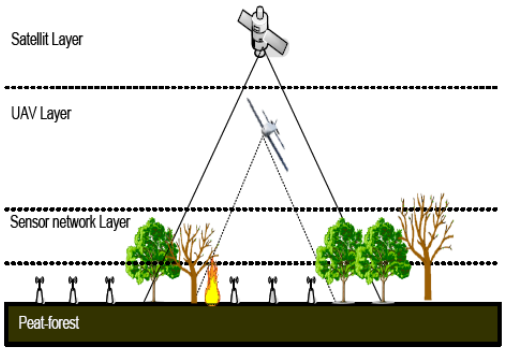

Figura 15- Três camadas de deteção (Teguh, Honma, Usop, Shin, & Igarashi, 2012)

- A *WSN* consiste em *Crossbow IRIS motes* e *MTS400 sensor board*. É composta pelos seguintes componentes:
  1. Temperatura, humidade relativa, pressão barométrica e luminosidade.
  2. *Routing* nodes para a transmissão de dados até estação base.
  3. Estação base ligada a um servidor web.
  4. Servidor web ligado a uma base de dados *PostgreSQL*.
  5. Cliente para receber os dados recebidos acerca de temperatura, e alarme de fogo com internet móvel.
  6. Gestão de eventos de sincronização de tempo, para rotinas de roteamento e gestão de energia.
- Os nós de sensores, composto pelo *MEMSIC ex-Crossbow IRIS mote* é operado usando *TinyOS*, que é desenvolvido para programar pequenos dispositivos com microcontroladores embebidos. Estes nós tem como principais funções a comunicação, processamento de dados e de sensor. Em adição, o *TinyOS* é programado em grande parte no *NesC*, que dá suporte a *component-based* e programação *event-driven* para a criação de aplicações na plataforma *TinyOS*. Os nós incluem, módulo de comunicação e de sensores. O módulo de comunicação utilizado, faz uso do standard IEEE 802.15.4, que taxa de transmissão de 250 kbps. A transmissão no exterior em LoS é 500 metros e 100 metros no interior entre paredes, utilizando uma antena ¼ *wave dipole* e uma potência RF de 3dBm.
- O nó de *gateway* da plataforma consiste num *Stargate NetBridge*, com sistema operativo. A sua principal função é servir um servidor de base de dados e um servidor web, que se encontram disponíveis através de um cliente web-browser. Permite despoletar alarmes de fogo e relatórios de eventos anormais de elevadas temperaturas.

Em relação à disponibilização da *WSN* e *UAV*, é necessário ter em conta o custo, número e localização, bem como a precisão e alcance dos sensores no ambiente em particular. É necessário ter em conta que o alcance de transmissão numa floresta é mais reduzido devido às obstruções como árvores, portanto a estimação da distância ideal é o ponto chave para a ótima propagação do sinal e assim existir a conectividade entre nós apropriada.

Para a disponibilização da camada de rede de sensores, é importante o alcance de transmissão, o que aumentando o requerimento de cobertura, faz com que a precisão dos dados também aumente. A técnica de separar a disponibilização dos sensores pode resultar numa transmissão de maior alcance e maior consumo de energia enquanto que uma disponibilização mais densa, leva a que o alcance de transmissão seja menor e seja usado menor energia. A topologia da rede deve ser desenhada para poupar energia, enquanto que também tem de oferecer ótimo roteamento em multi-salto, fiabilidade nos dados e qualidade na conectividade.

A cobertura de captura de dados e conectividade da rede são os dois pontos que trazem mais problemas quando aplicado numa topologia de disponibilização menos densa. A eficiência da disponibilização dos nós, vai fazer com os custos sejam minimizados e reduzida a computação e comunicação necessária. O recurso mais importante, para além da qualidade dos dados recolhidos, é a energia, a comunicação por multi-salto é explorada através de um *relay* de envio da informação através dos vários nós até à estação base. Como os nós mais próximos da estação base tem uma maior carga de comunicação dos pacotes, estes consomem mais energia. Normalmente quando se fala de canais *wireless*, é sinónimo de problemas com a qualidade de sinal, que depende sobretudo da aplicação, das características do ambiente e frequência do espectro utilizado. Como já foi falado neste tipo de ambientes as árvores e vegetação atuam como obstáculos que causam absorção e espalhamento do sinal. As comunicações por wireless, perdem potência de sinal de propagação com a distância, frequência, ganho de antena e potência. Os pontos chave que afetam o sinal neste tipo de aplicação são a distância ao solo, a distância do sinal, a reflexão no solo, a obstrução e difração causada pela vegetação e padrão de radiação da antena.

A técnica de *free space loss*, é utilizada para a propagação do sinal de forma ideal, sem que exista obstáculos por perto para criar reflexão e difração, o que permite prever a potencia de sinal recebida por determinado nó quando existe *LOS* entre dois nós. No entanto, para caminhos com obstrução não é adequado utilizar-se a técnica apresentada anteriormente, para quando o sinal é perto do solo. Para maximizar a conectividade é necessário utilizar-se uma determinada altura de antena que ultrapasse a *Fresnel Zone*.

Como a *WSN* será disponibilizada numa zona de difícil acesso, a fonte de energia deverá suportar as operações de longa duração dos nós, pois funciona com baterias de capacidade limitada e assim que a mesma acabar, faz com que o dispositivo seja desligado

da rede e afete o desempenho da aplicação de forma significativa. Os dispositivos podem fazer turnos, de hibernação e carga criando um balanço no consumo de energia por forma a maximizar o tempo de vida da *WSN*. A sua operação normal, é ter uma duração de ativo de 3 segundos, seguido de 7 segundos de hibernação e 0.3 segundos de *duty cycle*.

O principal objetivo desta *WSN* é investigar a habilidade do sistema em detetar o fogo e a robustez do hardware nas condições de fogo florestal. Os dados recolhidos são enviados do sensor para a estação base através de roteamento prioritário dos pacotes. A estação base processa os dados e guarda-os numa base de dados. Caso um dos nós detete fogo, os adjacentes alteram o seu *duty cycle*.

Para a recolha remota de dados, podemos utilizar dispositivos como satélites, *UAV* ou *aircraft*. Na deteção de fogo, existe duas técnicas que é a horizontal e vertical, no caso da horizontal é utilizada uma torre de vigia com guarda e auxílio de vídeo, já na vertical é usada a técnica de recolha remota. Neste tipo de recolha vertical, os métodos de recolha são feitos através de fotos, que permitem obter com precisão e atualizadas da situação. Permite ainda identificar, controlar e verificar as condições ambiental das zonas. É definida uma rota de observação, para execução da missão e análise do vídeo de vigilância pelo *UAV*. É utilizada camara infravermelhos e de cor para extrair informação. Posteriormente os dados são comparados com os recebidos da *WSN* para identificar a zona de fogo.

Da experiência realizada em campo, mais precisamente na Indonésia, utilizaram uma *WSN* composta por 6 nós de sensores, um *UAV*, um *gateway* e aplicação *MoteView* para monitorização. A distância a que foram colocados os nós uns dos outros foi de 100 metros a uma altura de 1,5 metros do solo. Este sistema possibilitou detetar fogos próximos com tamanhos de 3 metros, como a vegetação circundante tem altura entre 1 a 3 metros do solo, faz com que a potencia e propagação do sinal seja afetada. Foi efetuado um pequeno fogo para testar a capacidade da rede, os sensores ao verificar a variação de temperatura vai permitir detetar o mesmo e enviar para a estação base essas indicações.

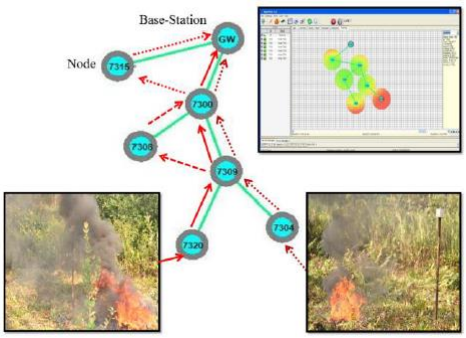

Figura 16 - Figura da rede WSN (Teguh, Honma, Usop, Shin, & Igarashi, 2012)

Do teste realizado, verificou-se uma temperatura mínima de 30ºC e uma máxima de 64ºC, o que resulta numa temperatura média de 46ºC, portanto para efeitos de deteção foi considerado uma temperatura mínima de 45ºC, que acima dessa, significará que se trata de potencial fogo e os sensores entraram em modo de deteção e comunicação ativa. Para além da temperatura em conjunto com os valores de humidade permitiu descobrir qual os nós que estavam a detetar fogo.

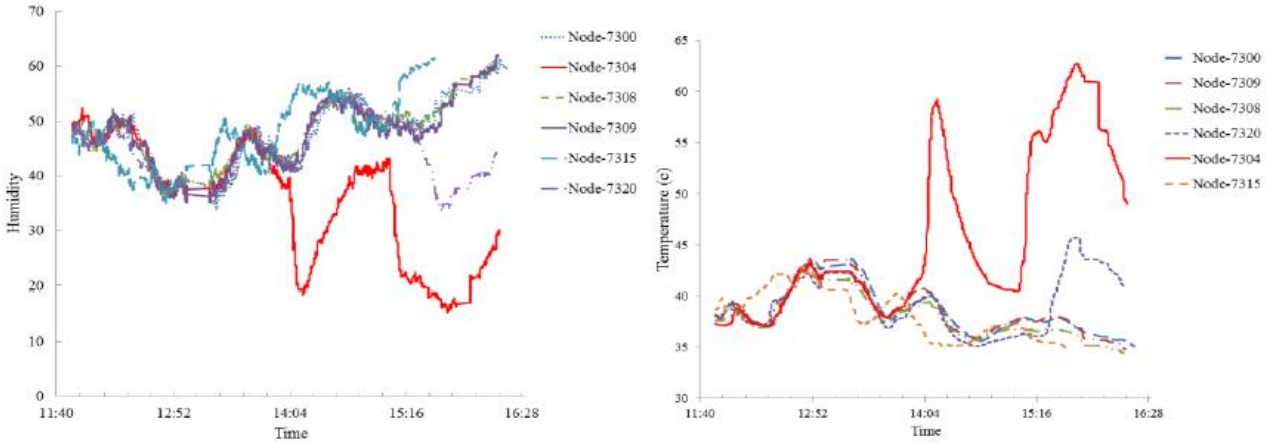

Figura 17- Medição de temperatura e humidade nos nós (Teguh, Honma, Usop, Shin, & Igarashi, 2012)

Com a deteção dos sensores permite posteriormente através de *UAV* verificar as localizações, onde o fogo está a ocorrer, com imagens de vídeo, o que identifica os locais com precisão para as patrulhas se deslocarem para o seu combate.

(Teguh, Honma, Usop, Shin, & Igarashi, 2012)

10.2 Tecnologias emergentes na área

10.2.1 An OCARI-Based Wireless Sensor Network for Heat Measurements during Outdoor Fire Experiments

Neste artigo, aborda que ao contrário do *Wi-Fi* que requer uma infraestrutura para trabalhar, as redes *WSN* são *ad-hoc*. Cada nó pode agir como router emissor-recetor, permitindo que a forma como a informação é transferida entre nós é dinâmica e reconfigurada periodicamente. Esta propriedade permite que a rede seja auto-configurável, em qualquer que seja a circunstância, mesmo que um dos nós deixe de funcionar, o nó vizinho irá reconfigurar uma nova rota para transmitir os dados, no próximo ciclo. O protocolo *IEEE 802.15.4* contém grande parte dos requisitos, eficiência e potência, que é necessário para o desenvolvimento de uma *WSN*. Especifica três topologias de rede, topologia em estrela, topologia em malha / ponto-a-ponto e topologia em árvore. Tanto a tecnologia *OCARI* como o *ZigBee* foram desenhadas para redes pessoais de curto alcance. O *IEEE 802.15.4*, é indicado para redes baseadas em microcontrolador que pode funcionar de dois modos:

- FFD - Os nós funcionam no modo *full function device*, isto é, cada nó é capaz de encaminhar a informação recebida ou recolhida, através dos nós vizinhos;
- RFD – Os nós funcionam como *end-devices*, que são nós finais de recolha de dados e que enviam mensagem com os dados recolhidos para nós *FFD*.

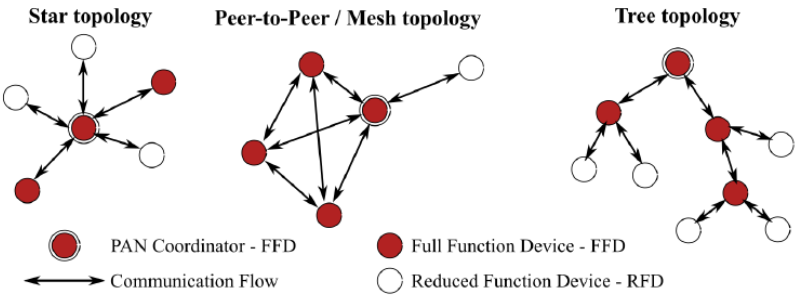

Figura 18 - Topologias e modos de funcionamento no protoloco IEEE 802.15.4 (Carlotti, et al., 2019)

O modo *FFD* tem como ponto negativo, o consumo de energia, devido ao encaminhamento estar constantemente a ser recalculado, sendo estritamente necessário um protocolo deste tipo para que caso um dos nós seja inativado pelo fogo ou outra razão externa, seja recalculada uma nova rota de encaminhamento para os nós vizinhos. Como o protocolo *IEEE 802.15.4*, apenas pode ser configurado com o modo *FFD*, por períodos

muito reduzidos (cerca de um dia) e o Zigbee impõem o modo RFD, faz com que grande parte das *WSN* utilizem o método de o protocolo apenas enviar pacotes com dados, caso os valores medidos ultrapassem um x valor ou seja detetado fogo.

O protocolo *OCARI*, é uma pilha vertical de software que inclui várias funções para os vários níveis e baseia-se nas tecnologias *ZigBee*. O *ZigBee* pode ser considerado como a camada superior do *IEEE 802.15.4*, em que no caso do *OCARI* introduz uma nova camada de controlo de acesso ao meio chamada *MaCARI*, que tem como função melhorar o uso do meio e especificar algoritmo de agendamento chamado *OSERENA*, que permite melhorar a atividade na rede. Na figura a seguir é possível ver a comparação entre o *OCARI* e o *ZigBee* / *IEEE 802.15.4*.

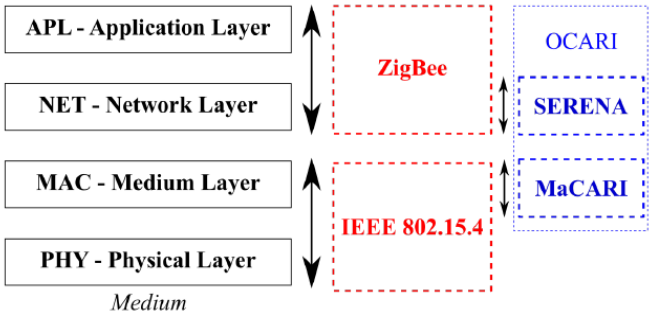

Figura 19 – Pilha de camadas ZigBee e IEEE 802.15.4 e OCARI

O protocolo *OCARI*, oferece meios para disponibilizar facilmente redes em malha de grande escala para ambientes industriais ou de investigação num determinado ambiente. Bem como, oferece melhor desempenho de consumo energético que o *ZigBee*. O limite teórico de módulos operacionais ronda os 400 nós, para criar uma rede *OCARI* é necessário lançar dois tipos de nós:

- Nós CPAN – São controladores de rede;

- Nós PAN – São dispositivos totalmente funcionais, com sensores ou atuadores, existentes nas redes com protocolo *IEEE 802.15.4*.

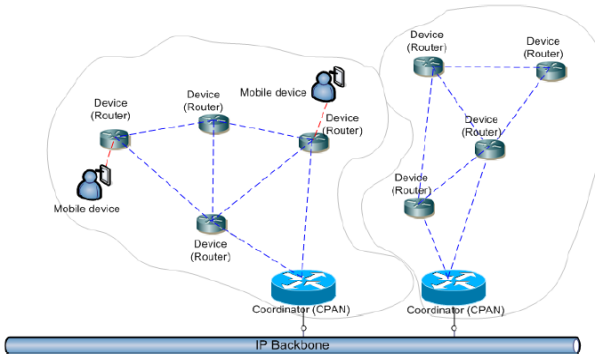

Figura 20 - Modelo de uma rede OCARI (Carlotti, et al., 2019)

Em estudos realizados anteriormente verificou-se que as redes *WSN* comparadas com dispositivos equivalentes ligados por cabo, mostra uma latência mais elevada na comunicação de fogos, no entanto a pilha do protocolo oferece uma qualidade de serviço mais apropriada para deteção de incêndios. O objetivo do estudo é concluir se a pilha *OCARI* quando comparada com a *ZigBee* oferece melhor *QoS*. Com a adição das funcionalidades do *OCARI*, permite que os nós tenham um agendamento cíclico de desligado, ligado, efetuar medições e de transmitir dados. Desta forma, os nós com *OCARI* tem um consumo de 0.3W face aos 1W com *ZigBee*, o que permite colocar baterias de capacidade mais reduzida nos dispositivos, bem como os elementos de uma rede *OCARI* tem dimensões 4 vezes inferiores aos da *ZigBee*, permitindo reduzir o seu tamanho. Traz como grande vantagem preservar os nós do fogo, especialmente se tiverem enterrados ou protegidos com isolamento cerâmico.

Foi desenvolvida uma placa *OCARI*, que garante que no período que os dados analógicos são digitalizados é finalizado antes de começar a criar o pacote de mensagem. Para além disso a pilha *OCARI* como é *Open-Source*, permite que as mensagens trocadas sejam modificadas, tendo sido neste caso alterado para conter o instante da medição, o valor da mesma e ainda mais onze informações relativas à rede, como identificação do nó e intensidade do sinal.

No estudo realizado, à cerca da habilidade das redes *WSN* baseadas em *OCARI*, mostra que conseguem ultrapassar as limitações que englobam as redes *WSN* baseadas

em *ZigBee*, na medição de calor durante fogo, em termos de registo de tempo, sendo perto do valor conseguido por cabo sem um grande custo em termos de energia.

11 Caracterização

11.1 Abordagem do problema

11.1.1 Análise do problema

No projeto abordado, ao se pensar na aplicação do mesmo, surgem logo à partida inúmeras dúvidas/problemas para o desenvolvimento e aplicação do projeto. É necessário que se encontre numa primeira fase, resposta para que tipo de ambiente vamos aplicar o projeto, que neste caso serão terrenos irregulares de floresta o que torna extremamente difícil a sua monitorização e dispersão do sinal.

De seguida, será necessário verificar-se qual o tipo de tecnologia que nos permite implementar um sistema fiável de deteção de incêndio, que neste caso temos ao dispor inúmeras formas, como a utilização de dispositivos *IoT*, torres de vigilância ou *UAV*, tendo-se optado inicialmente por a utilização de uma rede *WSN* de dispositivos *IoT* e *UAV*.

Colocando-se como seguinte questão, a forma de efetuar a comunicação entre os dispositivos, sendo que terá de conseguir comunicação em áreas extensas onde existem inúmeros obstáculos e grandes perdas de sinal devido a estes estarem suscetíveis à variação de condições climatéricas e à irregularidade dos terrenos e vegetação. Tal problema, subdivide-se em outros mais específicos, como qual a densidade necessária de dispositivos, qual o protocolo de rede utilizar para que permita conjugar as perdas de dados e não ter um grande impacto no custo de energia e dados, qual o tipo de arquitetura será necessário os dispositivos terem e como vão interagir com o *software* de suporte. O protocolo escolhido terá de ser capaz de permitir, que um elevado número de dispositivos sejam interligados, sem que o desempenho da rede não seja afetada e não envolva um gasto anormal de bateria. Outra questão associada às escolhas anteriores, é o modelo escolhido, visto existirem múltiplas opções de como implementar um fluxo de dados, desde os sensores até ao software de suporte de análise de dados. É exigido que não tenha uma complexidade muito elevada, por forma a ser fácil colmatar as falhas que possam ocorrer no mesmo, bem como não faça com que o custo de implementação se torne avultado.

Será necessário, como peça fundamental do projeto, a escolha de quais variáveis de ambiente recolher, visto estarem ao dispor inúmeras, como temperatura, humidade, concentração de gases e outras. Deste modo, a escolha correta das mesmas permite ajudar a detetar com fiabilidade a existência de incêndio. Também é necessário, avaliar o impacto ambiental e económico dos dispositivos, visto terem uma vida útil curta, derivado aos fatores associados ao ambiente em que se vão encontrar.

11.1.2 Variáveis de Dados

Para a correta deteção / monitorização de fogos, é necessário a escolha certa dos elementos a recolher, neste caso segundo o levantamento anteriormente feito de artigos sobre projetos semelhantes, foi analisado que as variáveis que melhor se enquadram na deteção de fogo em sensores com energia limitada são a temperatura, humidade e concentração de gás. A conjunção das três variáveis permite-nos obter uma simulação do que está a ocorrer no ambiente que envolve determinado local, sendo que caso esteja a ser ateado ou deflagre um foco de incêndio, rapidamente seja detetado. Para tal, torna-se necessário que os dados recebidos sejam inicialmente tratados sobre a forma de identificar o local de onde provém, que pode ser obtido através de um identificador do dispositivo. No entanto, para sabermos a localização do mesmo, após o seu *deploy*, mostra-se necessário recorrer ao uso de um módulo *GPS* que nos oferece uma grande precisão na obtenção das coordenadas do mesmo. Deste modo, permite mapear os locais de cada nó / dispositivo. (Hariyawan, Gunawan, & Putra, 2013)

Como segundo ponto essencial na deteção e análise dos dados, mostra-se importante ter um algoritmo que permita através das variáveis recolhidas fazer o tratamento, com a intenção da obtenção de um valor de risco de incêndio, assim como a deteção de fogo. Para tal, com base em indicadores previamente definidos podemos calcular a percentagem através da diferença inversa do valor obtido face ao valor padrão de incêndio, sendo que desta forma quanto mais elevada a percentagem, menor é a diferença entre o valor padrão e o obtido. Através dos valores em formato percentagem, permite ao utilizador final ter uma melhor perceção do risco, permitindo que em determinadas situações como medida preventiva o mesmo diminua o intervalo de recolha de dados dos dispositivos. (Solobera, 2010) (Krull, Tobera, Willms, Essen, & von Wahl, Early forest fire detection and verification using optical smoke, gas and microwave sensors, 2012)

11.1.3 Tipos de dispositivos

Em inúmeros artigos que analisam e apresentam soluções sobre o tema, apresentam dispositivos de pequenas dimensões, compostos por um reduzido grupo de sensores integrados, bateria de dimensão reduzida e uma *board* composta por uma placa de rede integrada para as comunicações.

No caso de um dos projetos retratados anteriormente, *SISVIA*, que se encontra a ser utilizado no terreno, emprega um dispositivo proprietário chamado *Waspmotes*, que é desenvolvido por uma empresa espanhola, chamada *Libelium*, no entanto este dispositivo não é nada menos do que uma espécie de *arduino* com sensores de temperatura, humidade, luz e GPS integrados com a adição de uma bateria e painel recarregável.

Figura 21- Dispositivo Waspmote

(Solobera, 2010)

Outro exemplo de dispositivo proprietário, é o referido no artigo sobre o protocolo *OCARI*, onde é explicado que este protocolo devido ao seu baixo consumo e falta de necessidade de placa de rede complexa comparativamente com o *ZigBee*, permite a criação de dispositivos com dimensões quatro vezes inferior ao que era necessário num baseado em *ZigBee*. O mesmo é composto por um módulo de comunicação de rede, com dois transmissores, módulo de deteção de fogo. (Carlotti, et al., 2019) Visto serem dispositivos proprietários, podemos utilizar outros semelhantes, permitindo ter-se liberdade de escolha quanto ao sistema que corre nos mesmos e possibilitar efetuar toda a configuração necessária de acordo com o que se achar mais correto, resumindo um maior controlo, mas como desvantagem irá trazer um maior esforço no desenvolvimento e configuração de software. No entanto, trás a mais valia de reduzir o custo de aquisição dos dispositivos.

A utilização de dispositivos comuns e que são facilmente adquiridos poderá ser o exemplo do *Rasperry pi* ou *Arduino* que permite juntamente com a acoplação dos sensores necessários (Temperatura, Humidade, Pressão atmosférica e *GPS*) e da fonte de

alimentação (bateria e painel fotovoltaico) criar um dispositivo de monitorização com todos os recursos que se mostrem necessários para o projeto. Este tipo de equipamentos para além do seu baixo custo, tem como grande vantagem encontrarem-se disponíveis em múltiplos locais o que poderá permitir que o custo dos mesmos seja mais baixo conforme o fornecedor do qual se adquira, graças ao fator de competitividade existente. Para além disso, existem múltiplas ofertas de sensores disponíveis no mercado, graças ao marcado e desenvolvimento existentes à volta dos mesmos. (Sharma, Siddiqui, Baig, & Ansari, 2017)

Noutros artigos / projetos existentes sobre o tema, para além dos dispositivos *IoT* já indicados anteriormente, podemos também ver a utilização de dispositivos de auxílio na deteção ou monitorização de incêndios como os *UAV*, torres de vigilância e satélites. Estas formas de vigilância e recolha de dados podem muito bem ser conjugadas à vigilância com os dispositivos *IoT*, aumentando assim a sua eficácia. No caso dos *UAV,* passa pela utilização de *drones* nalguns dos casos e balões de ar quente, funcionando em altitudes relativamente baixas devido às suas particularidades. Os *UAV,* são dispositivos aéreos não tripulados e completamente autónomos se desejável, equipados com camaras de infravermelhos, sensores de gás e partículas ou recolha de imagens de espectro do visível, permitindo assim que em condições noturnas seja possível a sua utilização para atuarem como suporte caso necessário. No caso dos balões de ar quente, são equipados com sensores de gases e fumo, camara de infravermelhos e ultravioletas. (Krull, Tobera, Willms, Essen, & von Wahl, Early forest fire detection and verification using optical smoke, gas and microwave sensors, 2012) No caso das torres de vigilância, são compostas por camaras de infravermelhos, *LIDAR* que permitem um extenso alcance, visto serem pontos de elevada altitude especificamente criados com o intuito de cobrir uma grande área da floresta que a rodeia. No entanto, estes pontos são fixos, o que faz com que não seja coberta a restante área onde o alcance das camaras não capta, para além do custo da construção das mesmas, que foi estimado ser 25000 dólares, devido à sua estrutura e locais remotos. (Eugenio, et al., 2016) (Tsetsos, Sekkas, Tsoublekas, Hadjieythymiades,

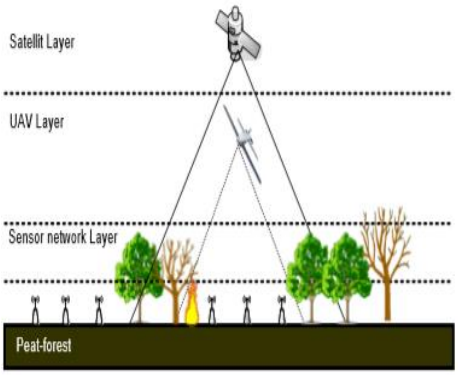

Figura 22 - Modelo de vigilância em camadas (Teguh, Honma, Usop, Shin, & Igarashi, 2012)

& Zervas, 2012) Em relação aos satélites apesar de cobrirem extensas áreas possibilitando a obtenção de imagens no espectro de infravermelho, se equipado com camaras infravermelhos. A sensibilidade dos dados recolhidos que os mesmos permitem, não é suficiente para a deteção de fogos de pequenas dimensões e as condições climatéricas terem influencia na leitura, tornando-se inviáveis para um sistema de deteção em tempo real. (Sudha & Murugan, 2017)

11.1.4 Vantagens e desvantagens das *WSN* em deteção de fogo

Apesar dos dispositivos terem um custo baixo e serem relativamente comuns, tem algumas desvantagens. Contudo, oferece múltiplas vantagens, como ajudar nos problemas de segurança na floresta, permite a recolha de variáveis meteorológicas, permite alta densidade de nós, elevada rapidez e fiabilidade na deteção e prevenção de fogo, fácil controlo da rede remotamente, permitem adicionar dispositivos sempre que se necessitar. No entanto, derivado em grande parte à complexidade e características associadas às mesmas, existe um número de limitações associadas, como os dispositivos serem facilmente desconectados da rede, devido a perturbações no sinal, serem danificados ou por avaria, assim como a limitação no processamento não sendo possível correr sistemas e algoritmos muito complexos, robustez em ambiente especialmente adversos, com grande humidade e elevadas temperaturas que podem levar à sua falha, vida útil dos dispositivos devido ao *deploy* em áreas de acesso restrito podendo não ter fontes de energia. Para além destas limitações, direcionadas ao sentido físico dos dispositivos, existem as referentes à tecnologia, como a largura de banda disponível, vida útil da bateria, capacidade de processamento e memória limitado, rapidez no envio da informação até ao destino em emergências e desastre. (Kaur & Manshahia, 2017)

11.2 Rede

11.2.1 Arquitetura

A arquitetura que compõe os dispositivos de uma rede *WSN* é composta por cinco camadas, que fazem parte do modelo OSI, denominadas de Camada de Aplicação, Camada de Transporte, Camada de Rede, Camada de ligação de dados e Camada Física. Como se trata de redes de sensores, onde a ligação entre dispositivos é feita de forma *ad hoc*, é necessário tipicamente a boa gestão de energia, o correto cálculo de rotas e qualidade de serviço. Para serem alcançadas essas necessidades, é necessário a articulação das camadas do modelo em planos, sendo estas subdivididas, de forma

perpendicular em três ou em cinco planos conforme o que o autor considere, cada um destes, é composto pelas cinco camadas do modelo OSI. As camadas que constituem os planos, são denominadas de plano de gestão de tarefas, plano de gestão de mobilidade, plano de qualidade de serviço, plano de gestão de segurança e plano de gestão de energia. (Ma, 2017) Estas camadas trabalham em conjunto, por forma a interligar os sensores e a melhorar a eficiência da rede, através da gestão da rede. (Alkhatib & Baicher, 2012)

No caso da eficiência relativamente ao consumo de energia a camada *MAC*, tem um papel importante para disponibilizar poupanças de energia e baixa latência, no entanto encontra-se dependente da informação, como taxa de transmissão e condições do canal utilizado, proveniente de outras camadas. A camada de ligação de dados, é necessária para a recuperação de perda de pacotes, enquanto que a *MAC*, pode prever a troca de dados permitindo-lhe a redução de energia. A junção da camada de ligação de dados com a camada de rede, são utilizadas para a descoberta de dispositivos vizinhos e reduzir a quantidade de pacotes de *broadcast*, que por sua vez, traz melhorias energéticas, no consumo.

Relativamente à segurança, a interceção entre as várias camadas do nó e as que constituem os outros nós, permite a deteção e controlo de segurança, no entanto podem ser usadas soluções como o uso de *frameworks*, (Pugliese, Pomante, & Santucci, 2009) que considera camadas fora da *WSN*, em vez de apenas a camada de Transporte do nó para implementar mecanismos de segurança. A utilização de *Middleware*, como exemplo o *AGILLA*, (Pugliese, Pomante, & Santucci, 2009) fora da rede *WSN*, permite que a interação entre camadas seja conseguida, oferecendo flexibilidade, escalabilidade e alcance global.

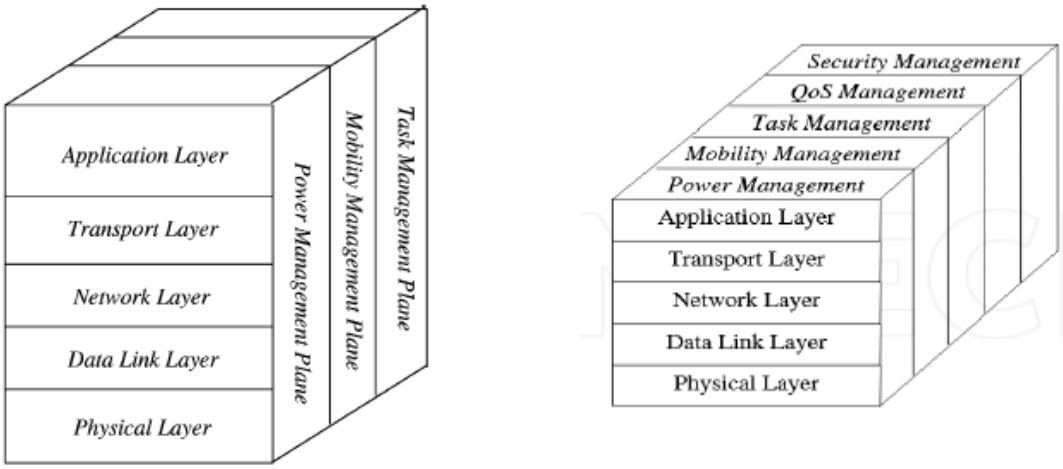

Figura 23 – Os dois modelos de Arquitetura WSN (Alkhatib & Baicher, 2012)

Mais detalhadamente, cada uma das camadas que fazem parte do modelo OSI:

- **Camada Física** – Forma a interface com o meio físico de comunicação, definindo a representação lógica da informação, que é transmitida através um fluxo de bits, com valores 0 e 1. A transmissão desses valores é realizada através da transformação dos mesmos em símbolos físicos, como variação de tensão, ondas eletromagnéticas ou feixes óticos; (Marques da Silva, 2016) É responsável pela seleção da frequência, gerar a frequência de portadora, deteção de sinal, modulação e encriptação de dados. (Abed Alhameed Alkhatib & Singh Baicher, 2012)
- **Camada de Ligação de dados -** Garante a comunicação num dado troço de rede e disponibiliza mecanismos locais de controlo de fluxo de informação e de controlo de erros. Está ainda dividia em duas subcamadas**, MAC –** que efetua o controlo de quando é que cada subestação de rede pode transmitir informação, mitiga o problema de interferência entre canais, normalmente denominado por *co-channel interference*; e **LLC –** que se encarrega de controlo de fluxo, controlo de erros e controlo de sequência. (Marques da Silva, 2016)

  Para a deteção de erros, podem ser utilizados dois métodos, *CheckSums* ou *CRC*, podendo ser feita a deteção através de *PAR*, que é reconhecimento positivo ou *NAK* através de reconhecimento negativo. (Marques da Silva, 2016)

FEC – Diminui o número de retransmissões, acrescentando dados redundantes em cada mensagem, para o que recetor detete e corrija os erros.

ARQ – Ao contrário do *FEC*, não é utilizado em redes *WSN* devido ao custo de retransmissão e o *overhead*, não sendo eficiente.

Os problemas de *shadowing* e *multipath fading* existentes nesta e na camada física, são resolvidos com o *FEC* e *ARQ*. (Abed Alhameed Alkhatib & Singh Baicher, 2012)

- **Camada de Rede -** Oferece os meios para que seja possível a ligação entre dois pontos, independentemente da sua localização geográfica e sub-redes que atravessa. Tem como principais funções, permitir o endereçamento e encaminhamento da informação utilizando para tal, protocolos e mecanismos. O protocolo IP, é aquele que é o mais utilizado nesta camada. (Marques da Silva, 2016) (Briscoe, 2000) No entanto, nas redes *WSN* o encaminhamento é uma das peças fundamentais para a preservação da energia, memória limitada e carregamento de dados. O objetivo do protocolo de encaminhamento, é definir um percurso que seja fiável e redundante, com base numa determinada métrica que é diferente para cada protocolo. *Data aggregation* e *data fusion* são usados como forma de disponibilizar cobertura total de uma área, mesmo acontecendo falhas de nós. É necessário a colocação de nós redundantes, para disponibilizar a repetição de dados, apesar da utilização de estilo *multi-hop* onde é enviado de nó em nó até ao destino e *flood*, onde cada sensor encaminha os dados para os seus vizinhos e assim sucessivamente.

  O encaminhamento pode ser feito através de vários tipos, *Flat routing*, *Hierarchy routing* ou dividido em periodicidade, com base em eventos ou intervalos.

  O tipo *Hierarchy routing* é composto por variados protocolos, como *PEAS*, *CLD*, *MET* e *LEACH*, sendo que todos possibilitam resolver os problemas de encaminhamento e de energia através da utilização de *clustering* e métodos de distribuição. O mais utilizado trata-se do *LEACH*, que divide a rede em clusters e que de forma aleatória seleciona um cluster para desempenhar o trabalho de cluster principal de encaminhamento para

agregação de dados e que posteriormente, após desempenhar a função é desativado.

(Abed Alhameed Alkhatib & Singh Baicher, 2012)

- **Camada de Transporte –** Assegura a comunicação entre extremos, garantindo independência quanto ao tipo e qualidade das sub-redes utilizadas, através do uso de mecanismos de deteção e recuperação de erros, controlo de fluxo e controlo de sequência. Contudo, nesta camada é onde o *QoS* é possível de ser disponibilizado. Estes mecanismos descritos anteriormente estão associados ao protocolo *TCP*, que é orientado à ligação, implicando que a mesma seja estabelecida e finalizada entre os pontos, garantindo a fiabilidade da informação que é trocada entre os pontos. Existe ainda outro protocolo nesta camada, que é o *UDP*, onde não implica que seja estabelecida uma ligação para a troca de dados entre pontos, que é importante para alguns serviços onde queremos trocas rápidas de dados e a organização da rede é dinâmica, mas como contra não dá garantias de fiabilidade na troca da informação. (Marques da Silva, 2016)

  Os protocolos existentes nesta camada que tem a capacidade de deteção de perdas, utilizam diferentes mecanismos, que podem ser *ACK*, *NACK* e *sequence number*. Já para recuperação de perdas, podem usar *End to End* ou *Hop by Hop*, no entanto a utilização deste último *Hop by Hop* é energeticamente mais eficiente que o outro, razão pela qual o *TCP* não deve ser usado. Normalmente a ligação proveniente do destino é considerada *downstream* numa transmissão *multicast* e tráfego UDP devido à limitada memória disponível e para evitar *overhead*.

  Os protocolos de transporte podem ser divididos em dois tipos:

  - *Packet driven* – todos os pacotes enviados por um emissor devem chegar ao destino;
  - *Event driven* – o evento deverá ser detetado, mas basta que uma das mensagens seja recebida no destino.

A seguir encontram-se alguns dos protocolos utilizados nesta camada do modelo OSI mais direcionados ao tema das *WSN*:

- STCP – É um protocolo *upstream* que é aplicado na estação base e disponibiliza confiabilidade, função de deteção de congestionamento e técnicas de evita congestionamento. O seu funcionamento consiste em um nó enviar um pacote de iniciação para à estação base que contém informações de taxa de transmissão, confiabilidade e encaminhamento, ficando a aguardar um ACK para iniciar a transmissão de dados.

  A estação base, efetua o cálculo do tempo que os pacotes deverão demorar a percorrer, em que caso exista uma falha na entrega de pacotes, a estação base verifica se a confiabilidade da ligação ainda mantém os requisitos, caso não seja envia um NACK, para que o nó voltar a retransmitir o pacote.
- PORT – É um protocolo *downstream* que assegura que o destino receba informação suficiente de um fenómeno físico. Por forma, a que adapte o encaminhamento dos pacotes para que o destino receba mais informação sobre determinada região:
  a) Método 1 – O custo de um nó é o total de transmissões antes do primeiro pacote chegar ao destino e é usado para definir o custo da comunicação. Cada pacote é enviado com o custo do emissor encapsulado, por forma a que o recetor / estação base ajuste a taxa de envio.
  b) Método 2 – Uso do custo da comunicação ponto a ponto para redução do congestionamento na rede. O destino reduz a taxa de envio de pacotes para as fontes com maior custo e aumenta a taxa de envio de pacotes para as fontes com um custo de comunicação menor.
- PSFQ – É um protocolo *downstream* oferece confiabilidade, escalabilidade e robustez. É composto por três funções:
  a) Pump – Faz uso de 2 cronómetros Tmin e Tmax, onde o nó espera Tmin antes de transmitir, por forma a recuperar os pacotes perdidos e remover redundantes. O nó espera Tmax se existir algum pacote perdido.
  b) Fetch – Solicita aos nós vizinhos a retransmissão de pacotes perdidos.

c) Report – Envia ao utilizador o relatório/informação ao utilizador.

(Abed Alhameed Alkhatib & Singh Baicher, 2012)

- **Camada de Aplicação –** Disponibiliza mecanismos de comunicação de alto nível orientados à disponibilização de serviços provenientes das aplicações. Os mecanismos prestados podem ser comuns a várias aplicações ou específicos a apenas determinada aplicação. A título de exemplo, para mecanismos comuns, pode ser o estabelecimento e término de associações entre entidades de aplicação e para aplicações especificas, mecanismos orientados à transferência de ficheiros ou mecanismos para aplicações de terminal virtual. (Marques da Silva, 2016)

Os planos criados sob as camadas do modelo OSI tem como objetivo, mitigar as falhas / dificuldades que a camada contém, oferecendo soluções. É efetuado sobre a forma de plano, devido a estas falhas/dificuldades serem causadas pelas múltiplas camadas e não apenas uma, sendo que a otimização de uma não iria trazer grandes vantagens, tornando-se ineficiente.

A pilha de planos que forma a gestão de planos, é composto por:

- Plano de gestão de mobilidade, tem como objetivo detetar os movimentos dos sensores, permitindo a um ponto da rede manter e conseguir encontrar outros pontos.
- O plano de gestão de tarefas, permite agendar e balancear tarefas sensíveis a determinada área, verificando quais os nós que se encontram na área ou não.
- O plano de gestão de energia, faz a gestão da atividade do dispositivo, em que o vai ligando e desligando conforme a necessidade de envio ou receção de mensagens.
- Plano de qualidade de serviço, faz a gestão da tolerância a faltas, controlo de erros e otimização da disponibilização do serviço. (Alkhatib & Baicher, 2012) (Harrison, 2016)

11.2.2 Distância entre os dispositivos

Para a realização de comunicações neste tipo de ambientes, mostra-se necessário avaliar e proceder a vários testes quanto à distância que é possível efetuar a transmissão de sinal, estando esta dependente da variação do vento e interferências do que os rodeia, pelo que é necessário através de simuladores específicos verificar qual a distância mínima e estabelecer uma distância máxima no pior caso possível. Desta forma, conseguimos garantir que no mundo real mesmo com alterações não esperadas, derivado de alterações climatéricas, seja possível a comunicação entre múltiplos pontos, mais precisamente os dispositivos que a compõem, garantindo assim que existem pontos de redundância e em momento algum parte da rede fica sem conseguir ter conectividade com a estação base.

Existem várias tecnologias de comunicação entre os dispositivos *IoT*, que permitem ter um maior *throughput* mas contudo, proporcionalmente maior consumo de bateria ou então maior alcance mas com o mesmo ponto negativo. Algumas das tecnologias mais utilizadas nas redes *WSN* são o IEEE802.15.4, IEEE802.15.4h, ZigBee, 6LoWPAN, Wireless HART e IEEE802.15.3.

Com base num artigo realizado por N. Ahmed, H. Rahman e Md.I. Hussain, estudaram o comportamento de dois standard de redes sem fios que podem ser utilizadas por dispositivos *IoT*, garantindo o seu baixo consumo, um é o *IEEE 802.11ah* e outro o *IEEE 802.15.4*, onde no caso do primeiro indicam que permite um alcance de cerca de 1000 metros em laboratório, que em condições ideais no exterior permite até 870 metros e interior até 543 metros mas um consumo de 63 mJ/Pacote num conjunto de 100 dispositivos, enquanto que para o standard *IEEE 802.15.4* tem um alcance bastante mais reduzido, com máximo de 37 metros e um consumo de 17 mJ/Pacote, também num conjunto de 100 dispositivos. (Ahmed, Rahman, & Hussain, 2016)

Com base em tais números, é possível ter uma ideia da densidade de dispositivos que será necessário ter para que uma rede *WSN* trabalhe conforme esperado, por exemplo se utilizarmos o *standard IEEE 802.11ah*, podemos considerar um alcance máximo de 500 metros, de forma a garantir que caso as condições se agravem as comunicações permaneçam, desta forma num hectare que é 10000m2 (10000 / 500 = 20), iremos necessitar de pelo menos 20 dispositivos, o que para dispositivos de baixo custo, não emprega um grande impacto orçamental. Desta forma, consegue-se ter uma densidade mais baixa de dispositivos e uma maior cobertura sem que tenha grandes desvantagens.

Contudo existem protocolos que ajudam a melhorar a performance da tecnologia sem fios, como o caso do protocolo OCARI que oferece meios para baixar o consumo dos dispositivos e disponibilizar redes em malha de grandes proporções.

11.2.3 Tipo de dados

Sendo uma rede baseada na deteção, com base nos dados recolhidos é fulcral a escolha correta dos dados. Os vários tipos de dados que deverão ser recolhidos, tem que obedecer ao uso mínimo de diferentes sensores, de forma a reduzir o custo dos dispositivos, bem como a economizar a bateria dos mesmos. Com base em vários projetos e artigos, verifica-se que quando inclui o uso de dispositivos de *IoT* são normalmente utilizados como parâmetros de recolha a temperatura, humidade, concentração de dióxido de carbono e monóxido de carbono. (Solobera, 2010) (Krull, Tobera, Willms, Essen, & Wahl, Early forest fire detection and verification using optical smoke, gas and microwaves sensores, 2012)

11.3 Segurança

As redes *WSN* ao serem desenhadas para baixo consumo e utilizarem como forma de comunicação meio não guiado, apresentam-se especialmente suscetíveis a ataques, pois os protocolos utilizados neste tipo de redes, não fazem uso de técnicas especiais de segurança utilizadas noutro tipo de dispositivos devido ao esforço requerido para tal.

O processamento e aumento da informação requerida na troca de pacotes, representa um acréscimo no consumo da energia dos mesmos, tanto no tempo de processamento como na distribuição da informação ao longo da rede até ao nó final. Isto é principalmente causado pelas técnicas de encriptação e desencriptação, que envolvem utilização de recursos de processamento, bem como os cabeçalhos associados à utilização e configuração dos mesmos. Outro dos motivos, é estarem a ser transmitidas comunicações que podem ser facilmente intercetadas por outros dispositivos que se encontram no raio de funcionamento. Tal é possível porque, os dispositivos recebem os pacotes dos demais que se encontrem próximos, descartando-os caso não sejam direcionados para si, sendo derivado das características da tecnologia utilizada.

11.3.1 Objetivos da segurança em redes *WSN*

Para a segurança nestes tipos de redes existem múltiplos objetivos, sendo os principais a confidencialidade, integridade, autenticação e disponibilidade dos dados,

contudo existem mais, a atualização dos dados, auto-organização, sincronização e localização segura. A seguir, encontra-se descrito cada um dos objetivos mencionados anteriormente.

- A **confidencialidade** é representada pela troca de informação sem que terceiros a consigam interpretar, tornando-se um dos pontos mais importantes na segurança da rede.
- A **integridade** é fundamental para se assegurar que a informação trocada é legitima e que a mesma não foi adulterada por terceiros. Apesar de ser aplicada confidencialidade na comunicação é possível que a integridade dos dados possa ser violada, pela alteração dos dados. Isto pode ser conseguido através da inserção de dados falsos na rede por um dispositivo malicioso ou interferências na frequência utilizada o que pode originar perda de dados e danificação dos dados através de bits corrompidos.
- A **autenticação** é outro fator importante, porque assegura que os dados recebidos pertencem à origem que diz ser, contudo a sua aplicação em redes de sensores é extremamente difícil. A autenticação é assegurada através da identificação dos emissores e dos recetores, conseguida com mecanismos assíncronos ou síncronos, onde existe a partilha de chaves secretas. Desta forma, impede que sejam injetados pacotes falsos na rede, que são compostos por informação adulterada com intenções maliciosas.
- A **disponibilidade** determina a habilidade de um determinado nó da rede *WSN*, conseguir utilizar recursos de rede para o envio de dados. A falha de uma estação de base pode levar a que uma rede de sensores seja inativada, que é extremamente importante que esteja ativa para desempenhar o seu papel.
- A **atualização dos dados** é importante pois apesar de garantida a confidencialidade e integridade dos dados é necessário assegurar que os dados que estamos a receber, são os mais atuais e não se tratam de pacotes que circulavam na rede. Para mitigar tal problema, basta criar um contador, que é incluído em cada um dos pacotes trocados, assegurando assim, que não se recebe dados repetidos ou mais antigos dos que já foram recebidos.
- A **auto-organização** é um ponto que é requerido visto numa *WSN* a comunicação normalmente é feita de forma *ad-hoc*, o que obriga a que

cada dispositivo seja independente e flexível para que se organize e configure na rede, de acordo com diferentes situações que possam ocorrer. Por ser assim, faz com que tal seja um desafio à segurança de uma *WSN*, pois se um dos dispositivos da rede não o consegue fazer pode resultar num ataque ou ter consequências devastadoras na rede.

- A **sincronização** é importante porque na maioria das aplicações baseadas nestes tipos de redes, funcionam com base em espaço temporal. Torna que seja importante existir sincronização entre os vários pares da rede, para que o grupo se encontre sincronizado entre si e o tempo de envio de pacotes seja sempre o mesmo ao longo emissor-destino.
- Em inúmeros casos, é necessário que seja calculada a localização de determinado dispositivo da rede, mostrando-se assim importante, o conceito de **localização segura**. A habilidade de determinar automaticamente a localização de um dispositivo permite que caso ocorra falhas seja detetada a origem. No entanto, um atacante poderá facilmente manipular os sinais de terminado dispositivo com recurso a replicação de sinais e aumento da força do sinal.

(Shelby & Bormann, 2009) (Padmavathi & Shanmugapriya, 2009)

11.3.2 Ataques a *WSN*

Normalmente, estes tipos de dispositivos encontram-se especialmente vulneráveis, devido à natureza de comunicação utilizada e ao ambiente que os rodeia ser hostil ou perigoso e, por não existir proteção física aos mesmos. Os ataques levados contra os dispositivos podem ter classificação do tipo ativo ou passivo.

11.3.2.1 Ataques ativos

Os ataques deste tipo são não autorizados e podem visualizar, modificar ou monitorizar os dados presentes nas comunicações entre dispositivos da rede.

11.3.2.1.1 Ataques de encaminhamento

São ataques que ocorrem ao nível da camada de rede e que são efetuados enquanto está a ser feito o encaminhamento dos pacotes na rede. Estes tipos de ataques podem ser subdivididos em inúmeras variantes, três exemplos deste tipo são: ***Spoofed, altered and replayed routing information*** - consiste na criação de novas rotas, criação de mensagens

de erro falsas e aumento de latência; ***Selective forwarding*** – Um dos nós de rede malicioso, apenas remove determinados pacotes e poderá combinar com outro tipo de ataque para encaminhar determinado tráfico através do nó, no entanto o nó poderá bloquear o envio dos pacotes e começar a utilizar uma nova rota de encaminhamento; ***Sinkhole attack*** – Tem como objetivo fazer com que o tráfego envolvente seja direcionado para o nó comprometido, através da alteração de informações de custo, passando a ser considerado como melhor nó para encaminhar.

11.3.2.1.2 Ataques de negação de serviço

Este tipo de ataque é produzido pela falha não intencional dos nós ou ações maliciosas, que tem como objetivo fazer com que uma determinada rede fique inoperacional, com interrupções momentâneas ou fora dos parâmetros normais, que tem como finalidade prejudicar a disponibilização de determinado serviço. Tratando-se de *WSN's* é possível desempenhar tais ataques a diferentes camadas, sendo estas a camada física onde pode ser efetuado através de interferência e adulteração; camada de ligação de dados através da colisão e exaustão da rede; camada de rede através de *misdirection*, *black holes* e *homing*; camada de transporte que pode ser através de *flooding* e dessincronização.

11.3.2.1.3 Node subversion

É a captura de um dos nós da rede, que permite através do mesmo a obtenção dos dados criptográficos utilizados pela rede, o que compromete toda essa rede.

11.3.2.1.4 Node Malfunction

O mau funcionamento de um dos nós da rede *WSN*, pode provocar com que sejam gerados dados errados, que assim sendo prejudica a integridade da rede.

11.3.2.1.5 Note Outage

Representa a situação de quando um nó que é líder de um cluster interrompe o seu funcionamento. Nesta situação, é necessário que protocolo utilizado seja robusto ao ponto de ser capaz de mitigar os efeitos da interrupção, através do cálculo de uma nova rota alternativa.

11.3.2.1.6 Physical attacks

Estando os nós sujeitos a ambientes hostis devido à sua forma de *deploy*, permite que sejam coletados por terceiros, o que pode acabar na sua destruição, que por usa vez

interrompe o funcionamento do mesmo na rede. Porém, pode ser utilizado para outros fins como obter as chaves criptográficas caso se esteja a utilizar, ou mesmo alterar o comportamento dos sensores.

11.3.2.1.7 Message corruption

Representa a modificação do conteúdo de uma mensagem, o que por sua vez compromete a integridade da informação.

11.3.2.1.8 False node

Consiste na utilização de um nó falso, para injetar informação maliciosa, que pode passar por colocar informação incorreta ou inibir que informação correta seja encaminhada. É um dos ataques mais temidos, pois pode injetar código malicioso na rede da *WSN*, que por sua vez pode ser transferido para todos os nós e assim destruir toda a rede ou utilizá-la para seu proveito.

11.3.2.1.9 Node replication attacks

É a utilização não autorizada da identidade de um dispositivo da rede, passando por copiar o seu ID e replicar os pacotes recebidos pelo mesmo, fazendo com que o desempenho da rede se degrade devido à corrupção e encaminhamento incorreto dos pacotes.

11.3.2.1.10 Passive Information gathering

É conseguido através da recolha de informação de uma *WSN*, quando esta não se encontra encriptada. Ao recolher a informação permite ao mesmo que decifre os locais físicos de cada um dos dispositivos da rede e posteriormente os destrua ou observar os dados que são trocados na rede.

11.3.2.2 Ataques passivos

Este tipo de ataque ao contrário dos ativos, passa apenas pela monitorização e leitura não autorizada dos dados que são comunicados na rede. Pode-se dizer que este tipo de ataque é direcionado contra a privacidade / confidencialidade. Um dos maiores problemas não é a recolha da informação transmitida, mas sim a possível disponibilização da mesma externamente.

11.3.2.2.1 Monitor and Eavesdropping

É o ataque passivo mais comum, que consiste na receção de informação não autorizada, possibilitando ao atacante apanhar informação relacionada com o controlo da

rede, que pode permitir outro tipo de acesso, mais propriamente ao servidor, podendo posteriormente ser aplicado o *eavesdropping*.

11.3.2.2.2 Traffic Analysis

Consiste na análise dos dados recolhidos, o que pode levar à descoberta de padrões na comunicação. Estes padrões encontrados, podem indicar possíveis informações úteis aos atacantes, apesar de os dados se encontrarem encriptados.

11.3.2.2.3 Camouflage adversaries

É a colocação de um nó na rede ou comprometer nós da rede para os omitir. Posteriormente, podem ser copiados e servirem para receber a totalidade dos pacotes para análise dos mesmos.

(Padmavathi & Shanmugapriya, 2009)

11.4 Middleware

O *middleware* fornece serviços para aplicativos de computação de alto nível que se baseiam em deteção e usam uma rede de sensores sem fios com um sistema operativo incorporado. Assim, preenche a falha que existe entre os requerimentos (fiabilidade, flexibilidade e reutilização) que tem uma rede baseada em deteção e as operações complexas que ocorrem na rede *WSN* (topologia de rede dinâmica, sistemas operativos de baixo nível e recursos restritos).

A abstração do hardware, possibilita a re-utilização de código relativo a serviços, como de recolha de dados e filtragem de dados, sem que o programador tenha que se preocupar com o seu *deploy* e execução e, a adaptação e gestão da infraestrutura da rede, através dos serviços que são disponibilizados pelo mesmo.

O modelo de *middleware* é composto por quatro componentes, abstração de programação, serviços de sistema, suporte a *runtime* e mecanismos de *QoS*. (Wang, Cao, Li, & K. Dasi, 2008)

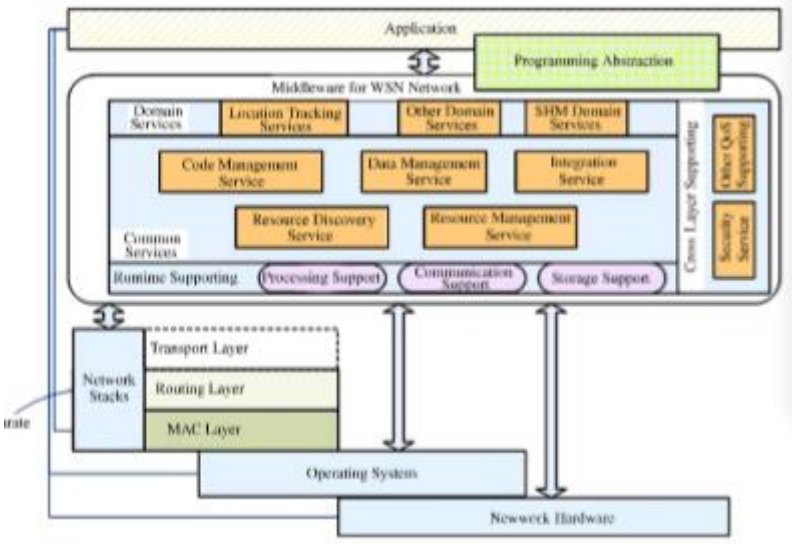

Figura 24- Modelo de Middlware (Wang, Cao, Li, & K. Dasi, 2008)

A Abstração de Programação oferece interfaces de programação de alto nível, separando o desenvolvimento de aplicações baseadas em redes *WSN* das operações que ocorrem na infraestrutura da *WSN*, desta forma facilita o desenvolvimento tornando-o mais flexível e menos dispendioso de recursos, pois permite gerar automaticamente a componente relacionada com o comportamento dos nós, colaborando entre si para desempenhar as tarefas exigidas. O suporte a *runtime*, oferece um ambiente onde as

aplicações podem ser executadas, podendo ser visto como uma extensão do sistema operativo, o que permite o agendamento da execução de tarefas, comunicação entre processos, controlo de memória e controlo de energia. Os mecanismos de QoS, apesar de serem um quebra-cabeças nas redes *WSN*, permite que as comunicações decorram sem problemas de maior, com todos os problemas que envolve redes deste tipo, como largura de banda reduzida.

A gestão dos dados nas *WSN* oferece serviços às aplicações para a aquisição de dados, processamento de dados e armazenamento de dados. Dando assim suporte a eventos de registo de dados, alteração e deteção de eventos relacionados com os dados, assim como facilita a manipulação dos dados, através do uso *querys* semelhantes às realizadas em base de dados.

Sendo este projeto baseado numa rede *WSN*, o *middleware* desempenha um papel importantíssimo de modo a assegurar a gestão e suporte da intercomunicação entre os vários dispositivos presentes na rede, visto as ligações e sua topologia poderem variar rapidamente. No entanto, envolve que se torne mais complexo do ponto de vista de implementação onde o tipo de dispositivo escolhido terá de ser compatível.

11.5 Sistema Operativo

Os sistemas operativos direcionados para *WSN's* são tipicamente mais simples do que os típicos para computadores e outros dispositivos. Pois, cada um dos dispositivos que os utilizam apenas precisam de umas centenas de linhas de código, visto estes sistemas operativos, ao contrário dos outros, não necessitam de ter suporte para interface de utilizador. Os constrangimentos de memória e mapeamento de memória, faz com que mecanismos de memória virtual se tornem dispensáveis e implementáveis. Ao contrário do que se possa pensar, os componentes presentes neste tipo de dispositivos não são completamente diferentes dos dispositivos embebidos mais comuns, o que possibilita a utilização desses sistemas operativos.

Normalmente, este tipo de sistemas operativos, são desenhados para recolha de dados em tempo real. Sistemas como, *TinyOS* ou *Contiki* são alguns exemplos dos especialmente desenhados para redes de sensores, sendo baseados em modelos de programação *event-driven*, em vez de *multithreading*, o que na prática significa que os métodos que constituem o sistema operativo são direcionados para gerir eventos e tarefas com conclusão semântica. A linguagem utilizada para estes dispositivos é *nesC* caso

utilizem o *TinyOS*, que é uma extensão da linguagem de programação C, no entanto por exemplo o *Contiki* utiliza linguagem C. Bem como este permite a abstração de programação semelhante a *threads*, com um *overhead* reduzido da memória, denominadas de *threads proto*. Este tipo de redes normalmente tem uma composição heterogénea, o que no caso de ambos sistemas operativos trazem o protocolo *NETCONF built in*, que é usado para gerir e resolver eficientemente problemas. A inclusão deste protocolo nos sistemas operativos traz como vantagem uma maior robustez e segurança. (Kaur & Manshahia, 2017) (Silva, Rodrigues, Al-Muhtadi, Rabêlo, & Furtado, 2019)

Existem alguns requerimentos que se deve ter em conta para a escolha de sistemas operativos, como já indicado anteriormente, estes tipos de dispositivos são bastantes limitados em termos de recursos. Alguns dos requerimentos a ter em consideração são:

- **Eficiência energética** – A grande maioria dos dispositivos de *IoT* funcionam com base em baterias ou outras fontes de energia limitada, o que se torna como um requerimento obrigatório à boa eficiência energética dos dispositivos. Os dispositivos *IoT* incluem componentes como microcontroladores, transmissores de rádio e sensores, que tem como característica a boa eficiência, o que obriga a que estes, consigam entrar em modos de baixo consumo e assim colocar o dispositivo em modo adormecido o máximo de tempo possível.
- **Baixa pegada na memória** – Comparativamente a outros dispositivos, os dispositivos *IoT* são bastantes limitados em termos de memória. Mostra-se necessário que o sistema criado esteja preparado para caber nesse espaço limitado disponível. Normalmente, a memória disponível são apenas alguns kilobytes, o que obriga aos criadores de conteúdo para os mesmos a disponibilizar bibliotecas muito bem otimizadas, que podem servir múltiplas camadas com funcionalidades comuns e estruturas de dados eficientes.
- **Suporte a diferentes componentes** – Devido à diversidade de componentes e protocolos que podem ser usados pelos dispositivos *IoT*, leva à necessidade de ser feito desenvolvimento de uma grande variedade de componentes e tecnologias de comunicação para diferentes arquiteturas e famílias de microcontroladores.

- **Conectividade** – A necessidade de comunicação existente para interagir localmente ou externamente, é uma funcionalidade chave da tecnologia IoT. Estes dispositivos normalmente são munidos de pelo menos uma interface de rede e utilizam uma grande variedade de tecnologias de rádio de baixo consumo, bem como outras caso necessário. A necessidade de suportar múltiplas tecnologias de camadas de ligação de dados e comunicar com outros dispositivos leva à utilização do modelo do protocolo IP.
- **Interoperabilidade** – A existência de uma vasta divergência de requerimentos torna-se impossível de incluir num único projeto, o que obriga a sejam criados múltiplos projetos que dão resposta a diferentes grupos de requisitos. Por sua vez, faz com que componentes do sistema sejam reutilizados o máximo possível, obrigando a que sejam bem definidos e estruturados, para que seja possível a sua fácil implementação por terceiros.
- **Capacidades em tempo real** – A precisão necessária para a execução de tarefas, como de leitura de dados em áreas florestais, obriga a que o sistema operativo cumpra este tipo de requisitos, que são denominados de *Real-Time Operating System.* É desenvolvido para garantir o pior cenário de tempo disponível para execução e de latência. Implica que funções *kernel* tenham de ser executadas com um tempo de execução determinístico, isto é, que cumpram os intervalos de tempo determinados.
- **Segurança** – Como grande parte dos dispositivos encontram-se acessíveis tanto fisicamente como através da rede à internet, é esperada a necessidade de meios de segurança e padrões de privacidade. No entanto, para tal é necessário que sejam utilizados mecanismos de segurança como criptografia e protocolos de segurança, conservando a usabilidade e flexibilidade dos dispositivos.
  (Afzal, Yu, Rehmani, & Zikria, 2018)

11.5.1 Event Handlers

No *TinyOS* cada componente efetua a gestão de determinados eventos, quando determinado evento ocorre, como por exemplo a receção de pacotes ou leitura de dados, o sistema operativo seleciona o componente correto para a situação. O evento vai fazer

com que seja invocado o contexto necessário, assim que concluída a execução do mesmo, é retornado de volta para o sistema. (Kaur & Manshahia, 2017)

11.5.2 Tasks

Quando o tratamento de um evento leva demasiado tempo no processamento dos cálculos envolvidos, pode levar a que outras tarefas sejam atrasadas ou mesmo inibidas de serem executadas, para tal o *TinyOS* disponibiliza as *Tasks* como um mecanismo de execução. Desta forma permite que determinada tarefa seja executada em segundo plano sem que interfira com os eventos importantes do sistema. (Kaur & Manshahia, 2017)

11.5.3 Contiki OS

É um sistema operativo *open-source*, desenhado para arquiteturas de microcontroladores como *AVR, 805 1* e *MSP430*. Inclui uma pequena implementação do protocolo IP chamado *uIP* e *IPv6* com *6LoWPAN* chamado *uIPv6*. A arquitetura é desenhada para suportar redes IP através de transmissão de rádio de baixo consumo e outras interfaces. A linguagem de programação utilizada é C e utiliza ficheiros *make* para criação do ambiente de compilação. Disponibiliza inúmeros exemplos e aplicações pré-criadas, bem como possibilita a simulação de inúmeras plataformas e microprocessadores, devido à virtualização dos mesmos no simulador chamado *Cooja*.

O baixo nível de abstração é dividido em plataforma e *CPU* para permitir portabilidade, que inclui *drivers* para os componentes. Contudo o *Contiki* apenas disponibiliza suporte básico para *thread* e temporizador. O sistema *RIME* é uma camada flexível de acesso ao meio e biblioteca de protocolo de rede que inclui paradigmas de comunicação de baixo nível. A pilha *uIPv6* faz uso do *RIME* e disponibiliza uma *API* chamada *protothreads* do estilo *socket* para ser utilizada pelas aplicações, que podem ser do utilizador, assim como as incorporadas no sistema operativo *Contiki.* (Shelby & Bormann, 2009)

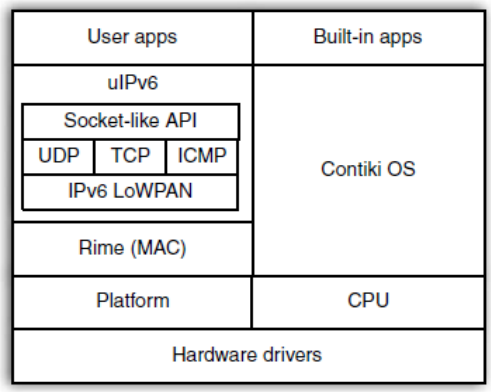

Figura 25 - Arquitectura Contiki (Shelby & Bormann, 2009)

11.6 Configuração

A periodicidade na recolha de dados por parte dos dispositivos, é extremamente importante não só na vertente do consumo da energia, bem como na veracidade/fiabilidade dos dados para a deteção atempada de incêndio. Tais variáveis obrigam que sejam feitas cedências tanto ao nível do consumo como fiabilidade, para que se chegue a um equilíbrio que traga mais vantagens do que desvantagens. É necessário adotar na configuração dos dispositivos uma abordagem de *duty cycle* que já é utilizada em grande parte dos dispositivos de *IoT* ou com fonte de energia limitada. Com a utilização deste método de funcionamento dos nós que compõem a rede *WSN*, permite que os recursos de energia sejam inferiores, não comprometendo a fiabilidade dos dados, visto os períodos de funcionamento permitirem que a informação estritamente necessária seja enviada. No entanto, é criado um atraso no levantamento dos dados, visto existir um período temporal onde não existe leitura dos dados.

A análise destes dados recolhidos por um software de suporte à deteção, permite que seja detetado um padrão onde indicie a possibilidade ou existência de fogo em determinado local. Este padrão é obtido através, do cálculo da média das várias variáveis recolhidas, compostas pela temperatura, humidade e concentração de gases. Nestes casos, sendo despoletado o alerta da existência de fogo, um *drone* ou *UAV* equivalente é ativado e enviado às coordenadas do ponto quente calculado pelo sistema, permitindo que a recolha de dados provenientes desse dispositivo, ajude a equipa de monitorização a constatar a veracidade e caso se justifique, a encaminhar os meios necessários para o local.

11.7 Composição da Arquitetura proposta

A arquitetura proposta tem como topologia escolhida, em malha, visto tratar-se de uma rede composta por centenas de dispositivos de baixo consumo. Neste tipo de topologias, os mesmos encontram-se ligados a mais do que um vizinho devido à particularidade deste tipo de rede, no entanto não existe relação hierárquica entre os mesmos. Embora, estes não tenham uma relação hierárquica, existem múltiplos caminhos possíveis até chegar à estação de perímetro, o que faz com que cada dispositivo atue como uma espécie de *router* / *host*, mais precisamente, torna-se um simples *host* quando desempenha o papel de coletar dados através dos seus sensores e os envia para o(s) dispositivo(s) vizinho(s) mais próximos, passando a desempenhar papel de *router* quando passa por receber pacotes de outros dispositivos vizinhos. Neste último papel, desempenha a função de encaminhamento, ou seja, encaminha os pacotes recebidos de dispositivos vizinhos para outro(s) dispositivo(s) vizinho(s) seguinte(s). A rota com que os mesmos fazem o encaminhamento, foi previamente definida e calculada pelo protocolo utilizado, que através de um algoritmo cria o caminho de menor custo. Desta forma, é criada uma rota teórica entre os vários nós que constituem a rede, por forma a garantir que os dados cheguem à estação de perímetro. Caso a rota se encontre indisponível, devido a um dispositivo intermédio já não se encontrar ativo ou na zona de cobertura, é calculada nova rota. (Gomes, 2010)

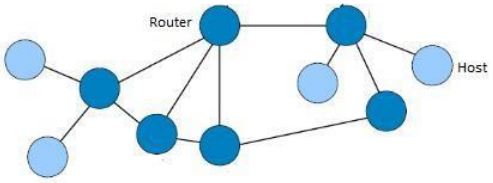

Figura 26- Rede em malha (Gomes, 2010)

Como meio de transmissão será utilizado o padrão *IEEE 802.15.4*, que tem um *MTU* pequeno de 127 *bytes* e taxa de transmissão de *250 kbps*, operando no intervalo de frequências da banda dos *2.4 Ghz*, o que é abaixo relativamente a outros padrões, no entanto oferece encriptação *AES-128* e método de autenticação. Suporta endereços curtos de *16-bit* para os dispositivos, para além dos endereços longos de *64-bit*, o que oferece uma redução no *overhead* das comunicações.

Aliando o *IPv6*, permite ter um intervalo de endereçamento disponível bastante elevado face ao *IPv4*, providenciando uma capacidade superior de endereços disponíveis, visto que o *IPv4* tem disponível *32 bits* versus os *128 bits* da versão IPv6 para criação de endereçamento. Tratando-se do *IPv6* o *MTU* tem como mínimo os *1280 bytes*, que é bastante superior ao tamanho disponível do padrão escolhido, tornando intensivo o uso

de recursos para a transmissão sobre *LoWPAN*. Para tal o protocolo *6LoWPAN*, implementa uma camada de adaptação entre a rede e ligação de dados para suportar a transmissão dos pacotes de *IPv6*, possibilitando as normais decisões de encaminhamento e redireccionamento, ao que se chama de *mesh-under*, por ser tomada pela camada de adaptação ao invés da camada de rede. Isto implica que exista diferença na processo de fragmentar os pacotes. Ao ser utilizado este modo, um pacote é enviado ao nó vizinho para que este envie a outro próximo até ao destino, a camada Multiple link, permite que seja feito um "salto IP" até ao destino. Mais detalhamente, no pacote inicial é colocado o IP de origem e o de destino, permitindo que os pacotes que são fragmentados sigam diferentes caminhos até ao destino, sendo posteriormente concatenados no destino caso todos cheguem com sucesso. Caso não cheguem todos ao destino, são novamente transmitidos todos os fragmentos pertencentes ao pacote IP. Este último ponto pode ser mitigado empregando o uso de *Selective Retransmission*, que é um mecanismo baseado em *negative acknowledgement* (NACK), entre fonte e destino. Ao usar este método permite que os fragmentos recebidos com sucesso pelo destino, sejam guardados num buffer, sendo apenas retransmitidos os fragmentos perdidos. Posteriormente, a camada de adaptação do nó de destino efetua a concatenação dos fragmentos anteriormente recebidos e os novos para formar o pacote IP. (Chowdhury, et al., 2009) (Gomes, 2010)

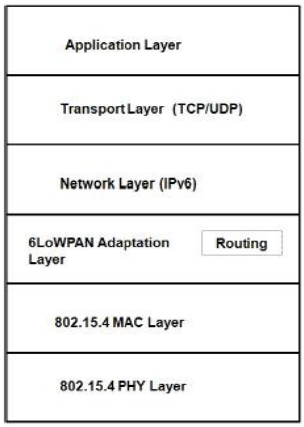

Figura 27 – Encaminhamento 6LoWPAN mesh-under (Chowdhury, et al., 2009)

Quanto à estrutura que compõem o projeto, passa pela existência de três pontos fundamentais para este projeto, sendo eles a rede *WSN*, estação de perímetro e o sistema de suporte à deteção. A rede *WSN* como já indicado anteriormente é composta por dispositivos *IoT* e *UAV* ligados por uma rede *IEEE 802.15.4* em que a qual serve de meio de transmissão dos dados pertencentes aos vários nós, até à estação de perímetro, que pode ser chamada de *gateway router*. Esta, posteriormente por *WebService* através de uma rede dedicada ou pública, consome o serviço disponibilizado por o servidor pertencente ao software de suporte à decisão, que através do mesmo recebe os dados pertencentes a cada um dos dispositivos e efetua o seu registo numa base de dados. Através dos dados existentes na base de dados, uma tarefa encontra-se continuamente a obter dados relativos às variáveis recolhidas e a calcular se determinado valor é

ultrapassado, para despoletar o processo de análise. Permitirá com base nesses dados mostrar os dados relativos a cada nó durante os vários momentos do dia, que com recurso a *Business Intelligence* permitirá compreender padrões, bem como questões climatéricas.

11.8 Pacote de dados

Os dispositivos utilizados, sendo estes munidos de interface de rede *IPV6*, permite que se utilize o protocolo *6LoWPAN* para as comunicações entre os mesmos na rede *WSN*. Ao se

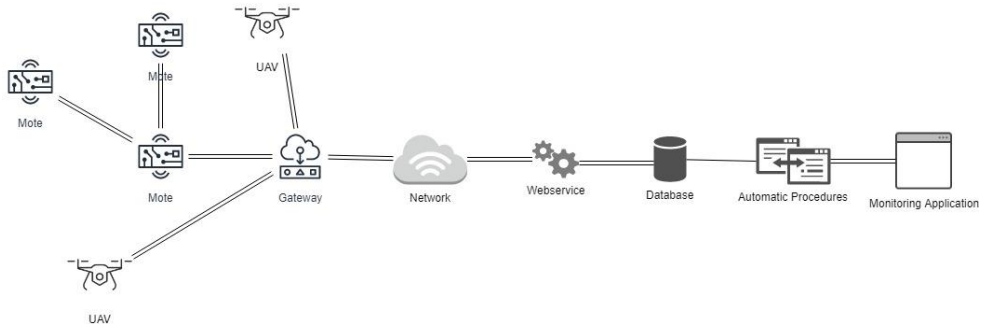

Figura 28 - Modelo da Rede

implementar a utilização deste protocolo é introduzida a camada de adaptação que tem como funções principais comprimir o cabeçalho *IPv6* e fragmentar / desfragmentar o pacote *IPv6*. O pacote *6LoWPAN* é composto pelos campos *802.15.4*, *Mesh Addressing Header*, *Fragment Header*, *IPv6 Header Compression e IPv6 Payload*. Durante a fragmentação e ao utilizar encaminhamento *mesh* é possível serem adicionados os dois cabeçalhos, sendo que no primeiro caso é correspondente ao *Fragment Header* e no segundo ao *Mesh Addressing Header.*

Existe a possibilidade de efetuar a compressão do cabeçalho *IPv6*, o que permite eliminar parcialmente ou totalmente os campos que o compõem. No caso de se tratar de ligações locais podemos omitir os 64 *bits* usados para 1 *bit*, reduzir o campo *Next Header* para apenas 2 *bits* por utilizar-se *TCP*, *UDP* ou *ICMPv6*, eliminar o campo *Payload lenght* visto podermos utilizar o cabeçalho do padrão. (Gomes, 2010)

Utilizando a estrutura do pacote *6LoWPAN*, podemos utilizar o campo de *payload* para colocar a informação que se considerada útil enviar até à estação de perímetro, que neste caso será a relativa aos valores dos sensores e identificação de cada dispositivo. Conforme explicado anteriormente, recolhemos inúmeras variáveis provenientes dos sensores e temos liberdade para colocar esses mesmos valores da forma que melhor se adequa ao projeto. A proposta é passar por enviar os dados em conjuntos separados pelo

carater “;”, o que torna uma forma mais fácil de dividir os dados recolhidos, sendo que cada conjunto será constituído da seguinte maneira:

id;temperatura;humidade;concentração de gases;latitude;longitude;bateria;potência

12 Implementação

12.1 Simulador

Para a realização de um protótipo / simulação da arquitetura anteriormente apresentada, foi necessário a utilização de um simulador que permitisse simular uma rede *WSN*. De entre os vários simuladores existentes no mercado, verificou-se que o *Cooja* pertencente aos desenvolvedores do *Contiki OS*, era aquele que melhor se encaixava para a realização do proposto. Visto ser um simulador capacitado de vários tipos de dispositivos, para utilização nas simulações e ainda por oferecer vários exemplos, estando disponíveis o código relativo à leitura de temperatura ou leitura dos valores de bateria, por exemplo. Para além disso, os dispositivos encontram-se “escritos” na linguagem de programação C, sendo uma das linguagens mais utilizadas, permite que se encontre uma grande quantidade de informação relativa à mesma.

Visto existirem vários dispositivos disponíveis, optámos por utilizar o modelo *Z1* da empresa *Zolertia*. Verificou-se que o mesmo permitia a utilização de sensores de temperatura, humidade, *GPS* e concentração de *CO2* que era os requisitos necessários para o projeto. Tal envolveu que fosse necessário desenvolvimento do código relativo à integração dos mesmos. Para além de tal foi necessário alterar o pacote de dados que enviava, passando a constar os dados recolhidos e o formato de informação projetado anteriormente indicado.

Para a simulação dos vários eventos, foi necessário a adaptação do código por forma a simular as mudanças de valores, que serão mais à frente explicados no capítulo da simulação. Para as simulações necessárias, foi necessário criar redes para cada um dos casos com uma composição de cinco dispositivos, sendo quatro equipados com sensores, que vão fazer a recolha dos dados e encaminhá-los, e uma estação de perímetro / *gateway* que agirá como um dispositivo que vai receber os pacotes enviados pelos vários dispositivos da rede e encaminhá-los para fora da

Figura 29 - Rede utilizada para as simulações

mesma, que pode ser via rede partilhada ou rede dedicada até ao sistema de suporte à deteção. A escolha de apenas quatro dispositivos de recolha de dados e uma estação de perímetro tem como intuito simplificar a recolha de dados e analisar o desempenho da rede mais facilmente.

12.2 Aplicação de Suporte à deteção de fogo

12.2.1 Conceito do protótipo

Para a criação do sistema de suporte foi necessário o desenvolvimento integral do código do mesmo para ir ao encontro do que era pretendido. Como se pretende que a aplicação se encontre disponível em qualquer tipo de dispositivo e permita que seja consultado em múltiplos locais, foi desenhado para ser disponibilizado como uma aplicação Web. Deste modo, permite que um telemóvel, tablet ou computador tenham acesso à aplicação, trazendo como grande vantagem poder ser acedido em qualquer local pelos meios no terreno ou num posto remoto onde exista acesso à internet.

Para o desenvolvimento do projeto, foi tido em atenção a utilização de aplicações e ferramentas que sejam *open-source*, para que não sejam aplicados custos no desenvolvimento do mesmo. No desenvolvimento deste projeto a linguagem escolhida foi, o *Java EE 8*, por ser uma das linguagens mais utilizadas na área do desenvolvimento de soluções profissionais, ser a linguagem que utilizo profissionalmente e ter sido aquela que foi mais abordada durante a Licenciatura e Mestrado frequentado, o que permite um desenvolvimento mais rápido e sólido.

Para a criação do projeto, foi utilizada a tecnologia *Maven* que oferece inúmeras vantagens na importação de bibliotecas e *framework's* necessárias ao projeto, sendo a sua principal, possibilitar a sua transferência a partir de um repositório. Como *backend*, é utilizada a *framework Spring* para possibilitar a integração das várias componentes do modelo de estrutura do projeto, que através da utilização de anotações permite poupar código duplicado e ajudar a cumprir o modelo *Model-View-Controller*, que divide a estrutura do projeto em três partes, controlador, modelo de dados e apresentação ao utilizador. Em conjunção, pela sua grande compatibilidade é utilizada a *framework Hibernate*, que explicando de forma simplificada, permite a configuração e desenvolvimento das operações de acesso, alteração, eliminação e modelo de dados à

base de dados. Como meio para armazenar dados provenientes tanto da aplicação, como dos dispositivos *IoT*, foi escolhida uma base de dados, que oferece um desempenho consistente para a quantidade de dados, solicitações e ser compatível com a *framework* utilizada. O motor de base dados utilizado, é o *MySQL*, por fazer uso da linguagem *SQL*, ao qual recorre uma grande variedade de aplicações, cumprindo os requisitos de atomicidade e desempenho requeridos. Porém, também oferece a possibilidade de utilização de uma ferramenta de gestão chamada *MySQL Workbench*, que permite o controlo absoluto da base de dados através de interface gráfica, ao contrário de outras propostas disponíveis no mercado.

Relativamente à camada de apresentação, o projeto utiliza a *framework Primefaces* visto ter integração direta com a linguagem java, permitindo uma maior flexibilidade na passagem de dados e configuração em conjunto com as outras tecnologias utilizadas neste projeto. Como auxiliar da *framework*, também é utilizada a tecnologia *bootstrap* por forma a embelezar, estruturar as páginas da aplicação *WEB* e oferecer compatibilidade com as diferentes resoluções dos equipamentos utilizados.

Para a realização da comunicação da estação de perímetro da rede *WSN* com a base de dados, foi criada uma segunda aplicação, que utiliza a tecnologia *REST*. A aplicação protótipo criada no âmbito do trabalho atuará como recetor dos dados provenientes da rede *WSN*, assim como *endpoint* da aplicação *WSN*. Permitirá que os dados recebidos através serviço sejam consumidos e registados na base de dados. Para tal, foi criado um serviço que está constantemente à escuta de datagramas através de uma porta selecionada, sendo que cada vez que recebe despoletará uma chamada ao *endpoint* da *WSN*. O *endpoint*, por sua vez será um método disponibilizado num determinado endereço, que quando invocado, permite que sejam enviados os dados encaminhados por cada um dos dispositivos pertencentes à rede *WSN*. A estação de perímetro utiliza o serviço disponibilizado pela aplicação *WEB* para envio dos dados recolhidos, sendo que posteriormente os dados são mapeados em objeto e guardados na base de dados. Inclusive, foi criado um serviço que recebe dados provenientes do IPMA, visto a mesma disponibilizar uma API de consulta de dados meteorológicos. Permite que os dados que não são possíveis de recolher pelos dispositivos, pela falta de sensores ou impossibilidade, sejam assim conseguidos. Por forma a simplificar o projeto que tem os serviços, foram incluídos os múltiplos serviços utilizados no mesmo projeto.

12.2.2 Detalhe da interface do sistema implementado

Este sistema, ao nível de interface de utilizador disponibilizará ao utilizador várias páginas onde poderá realizar múltiplas operações de consulta. Estas são o acesso aos últimos dados individuais recebidos dos dispositivos existentes, sendo estes a temperatura, humidade, concentração de CO2, bateria, potência do sinal de comunicação e data da última atualização recebida, permitindo monitorizar em absoluto cada dispositivo. Outra página disponível referente a cada dispositivo, é a página de inativos onde é apresentado numa tabela, quais os dispositivos que se encontram sem comunicação à mais de uma hora. Permite desta forma, que ao selecionar um dispositivo, se consulte a página especifica do mesmo, onde se encontram disponíveis informações relativas às coordenadas que tinha antes de ter ficado inativo por falha do mesmo, condições adversas ou outra razão desconhecida.

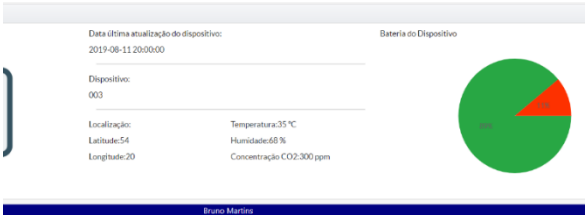

Figura 30 - Painel de dispositivo

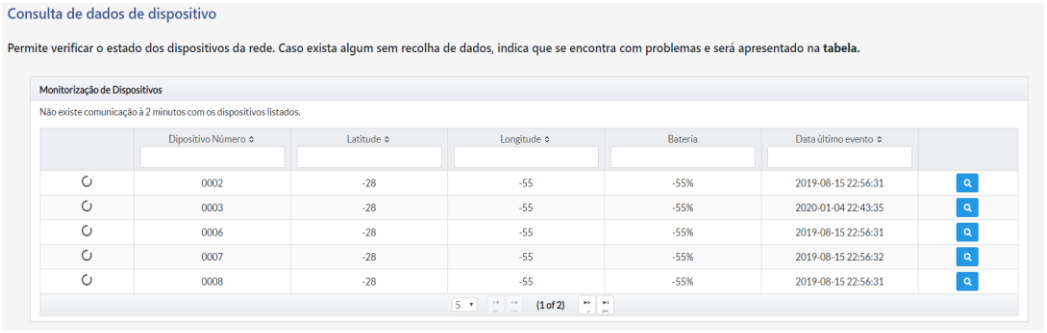

Figura 31 – Funcionalidade de consulta de dispositivos inativos.

Ainda dentro do mesmo menu de consulta, existe a funcionalidade Geolocalização, que permite ao utilizador visualizar o local que os dispositivos ocupam. Tal é possível, graças à utilização de fotos de satélite do terreno, disponibilizadas pelo *Google Maps* e a utilização das coordenadas enviadas por cada um dos dispositivos. Desta forma, permite que seja visualizada a sua triangulação no terreno e perceber qual a posição que ocupam na comunicação da rede *WSN*. Ao selecionar cada um destes pontos, permite que o utilizador fique a saber a identificação do mesmo e os seus dados da última atualização.

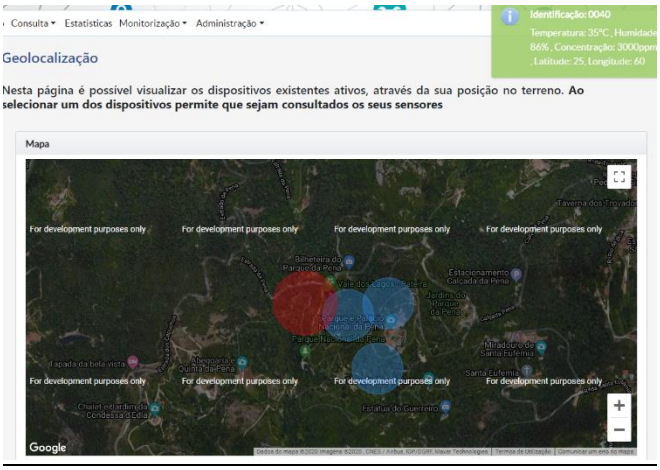

Figura 32 – Página que permite visualizar os dispositivos no terreno.

Outra opção disponibilizada, é a consulta de estatísticas, que permite ao utilizador observar graficamente a média dos dados recolhidos referentes a cada período e variáveis recolhidas. A página encontra-se divida em quatro partes, em que cada, tem uma descrição da janela visualizada e um gráfico referente aos últimos trinta dias, sete dias, últimas vinte e quatro horas e incidente, permitindo que através dos mesmos se observe a variação que existe ao longo dos dias ou horas, conforme o caso retratado. As estatísticas permitem aos operadores do sistema, fazerem previsões quanto às épocas de risco de incêndio ou simplesmente serem usados para fins estatísticos.

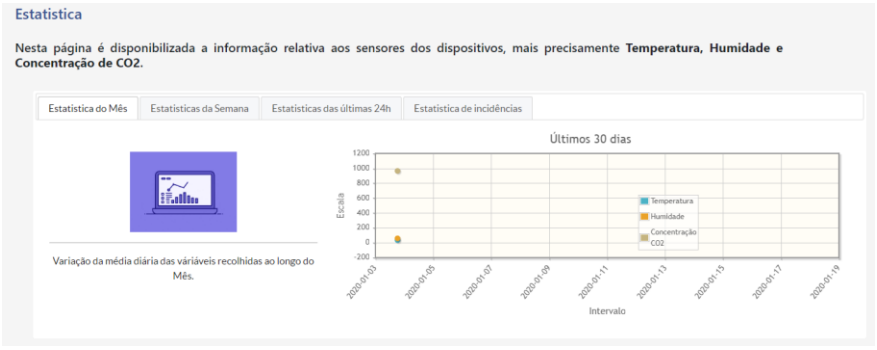

Figura 33 – Página dedicada a estatísticas.

Outro menu existente na aplicação é o da monitorização que se encontra dividido em duas páginas, sendo uma referente aos índices e outra relativa à monitorização da existência do incêndio. A página referente aos índices permite a consulta do *FWI* - Índice de Perigo de Incêndio e *FFMC* - Índice de Humidade dos Combustíveis Finos referente às últimas duas horas, em que o primeiro índice representa o nível de perigo de existência de incêndios e o segundo é referente ao potencial de deflagração de incêndios. No primeiro caso, o mesmo é calculado através de seis outros índices, sendo o segundo

calculado diretamente dos dados referentes à temperatura, humidade, vento, precipitação e mês, encontrando-se disponível no *website* do *IPMA* a explicação de cada componente do índice.

Para o cálculo dos índices, foi necessário a utilização de um algoritmo que envolve inúmeros cálculos, necessitando que fosse adaptado para funcionar em conjunto com o código do projeto, que por outras palavras tornasse possível integrar o formato de leitura dos dados realizado a partir da base de dados. Este algoritmo encontra-se disponibilizado num documento dos serviços florestais do Canadá (Wang, Anderson, & Suddaby, 2015). Como os dispositivos da rede, não tem presente sensores de medição de velocidade do vento e precipitação tornou-se necessário recorrer-se a recursos externos que nos indiquem tais valores, que é o caso dos *web services* disponibilizados pelo *IPMA* (IPMA, 2017). Quando estes índices ultrapassam os valores no caso do *FWI* índice 10 e *FFMC* índice 89, é gerada uma notificação na página com o nível de alerta para que o mesmo seja alertado dessa alteração e tome as precauções necessárias. Os valores utilizados para gerar a notificação baseiam-se em artigos existentes, onde explicam os vários valores de gravidade. (Bouabdellah, Noureddine, & Larbi, 2013)

Figura 34 - Monitorização de índices

No mesmo menu de Monitorização existe uma outra página disponível que apresenta os locais de dispositivos onde o risco de incêndio é extremamente elevado ou já se encontra a deflagrar. Para tal utiliza uma escala de gravidade entre um e três, sendo três o mais elevado, sendo que sempre que é detetado risco ou fogo ativo, será notificado do mesmo. Esta página tal como a dos índices, atualizam de forma automática os dados dentro de um intervalo definido, para que o utilizador não necessite de recarregar a página. Esta escala, foi baseada mais uma vez em artigos que abordavam o assunto. (Toledo-Castro, Santos-González, Hernández-Goya, & Caballero-Gil, 2017) Ainda nesta página, o utilizador tem a possibilidade de fazer o reconhecimento do incêndio enviando um *drone* para o local selecionado, encaminhar para as autoridades competentes de combate as coordenadas e

por último registar num histórico de ocorrências o evento selecionado, para que mais tarde possa ser analisado através de um gráfico de ocorrências. A comunicação com o *drone* e as entidades poderá ser feita por *web service* caso desejem, não tendo sido implementada

Figura 35 - Página onde permite o envio de meios, dispositivos de reconhecimento e guardar registo do incêndio

no trabalho por este se tratar de um protótipo, sendo apenas notificado o utilizador de que efetuou a operação.

No menu de Administração, existe duas páginas disponíveis, uma referente à parametrização de variáveis do projeto e outra referente aos dados existentes na base de dados. Na página de "Dados gerais", é possível consultar os dados que se encontram na base de dados. Apesar de os mesmos virem listados em bruto, a tabela permite a filtragem e ordenação por coluna, para por exemplo verificar-se os dias em que determinada temperatura esteve ou verificar se o dispositivo registou dados em determinada data. Na página de parametrização, permite que o utilizador configure parâmetros referentes a aplicação e seus serviços, por exemplo o local dos dados provenientes do IPMA, valor da chuva, entre outros necessários ao funcionamento da aplicação.

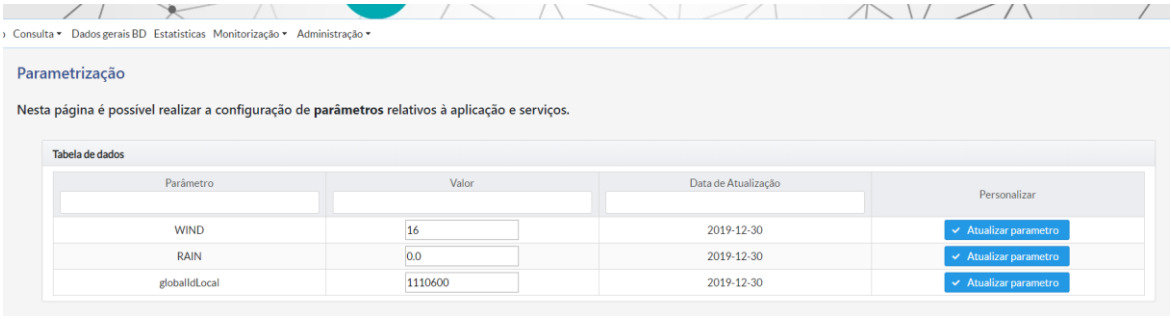

Figura 36 – Página dos parâmetros.

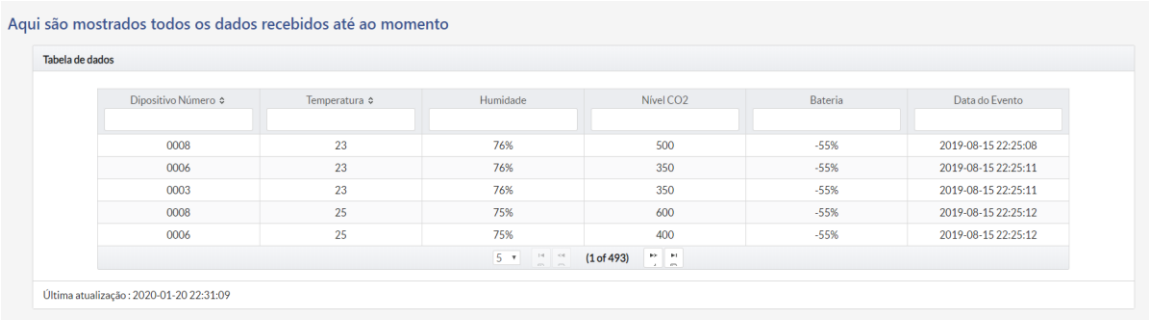

Figura 37 – Página de dados gerais.

12.2.3 Serviços disponibilizados

Relativamente ao projeto dedicado aos serviços, este é composto por quatro recursos REST. O primeiro, tem como função disponibilizar um *endpoint*, onde possibilite o envio de dados por uma estação de perímetro para a aplicação. Este recurso encontra-se disponível através do endereço **/projetoWSNServicesREST/recebeDadosMote* recorrendo a um POST com um objeto do tipo Mote. Como se mostrava necessário receber dados meteorológicos atualizados, foi implementado um serviço, que utiliza a API disponibilizada pelo IPMA. Foi necessário criar um recurso que permita à aplicação ir atualizando os dados relativos ao vento e ocorrência de precipitação, para tal o recurso, recebe um objeto JSON que é convertido em objeto, do qual são guardadas as variáveis. O recurso encontra-se disponível para invocação através do endereço **/projetoWSNServicesREST/IPMAService*, que por sua vez faz uma chamada ao endereço da API do IPMA, *http://api.ipma.pt/open-data/forecast/meteorology/cities/daily/{globalIdLocal}* através de um *GET*.

Por forma a simular uma estação de perímetro / *gateway*, foi criado um recurso que permite invocar o recurso *recebeDadosMote* de modo a encaminhar os dados para a aplicação. Como a rede WSN está a ser simulada numa máquina virtual, a comunicação com o hospedeiro foi conseguida através do envio de datagramas para um porto especificado. O recurso criado quando invocado, fica indefinidamente à escuta dos mesmos na rede. Sempre que o recurso receba um datagrama, encaminha os dados recebidos para o recurso de destino, *recebeDadosMote*, através de um pedido GET, simulando como se processaria no terreno com uma estação de perímetro. O recurso encontra-se disponível através do endereço **/projetoWSNServicesREST/enviaDados*.

Por último foi criado um recurso simples que permite invocar para saber se o servidor aplicacional se encontra disponível. O recurso pode ser invocado através do endereço **/projetoWSNServicesREST/status*, que quando invocado, retorna a data atual.

13 Avaliação

13.1 Simulação da *WSN*

Neste subcapítulo, serão simuladas diferentes condições de uma rede *WSN*, tendo por base a observação do comportamento dos dispositivos que a compõem submetidos a eventos de inexistência de fogo, deteção de fogo ou início de queimada. Com base nesta recolha de dados mais à frente será efetuada a sua análise em conjunto com o sistema. Devido à restrição que o simulador utilizado impõe nas leituras realizadas a partir dos sensores incorporados no dispositivo, foi necessário de forma sintética simular o comportamento das variáveis conforme a simulação pretendida. Isto, obrigou a que no código dos dispositivos fossem incorporadas variáveis que vão fazer o papel de dados recolhidos, que assim permite a manipulação dos valores das mesmas em função do tempo. Desta forma podemos por exemplo aumentar e diminuir a temperatura com base no número de iterações / comunicações do dispositivo. Esta variação foi definida com base nos valores de deteção aplicados no sistema de suporte à deteção de incêndio. (Bouabdellah, Noureddine, & Larbi, 2013) (Wang, Anderson, & Suddaby, 2015)

O projeto de simulação foi configurado para um alcance de sinal máximo de cem metros, com uma taxa de sucesso de transmissão de setenta e cinco por cento, por forma a tentar simular um ambiente real. Como temos disponível um alcance de cento e cinquenta metros os mesmos foram separados uns dos outros perto do alcance máximo à exceção daqueles que pretendemos ter redundância. Todos os motes foram programados para comunicar a cada dez segundos, por forma a aumentar a eficiência energética.

13.1.1 Simulação sem existência de fogo

Nesta simulação efetuamos a simulação de uma *WSN* onde as condições ambientais não apresentam a existência de fogo. Para tal, utilizaram-se nós onde apenas efetuam a obtenção de valores compreendidos entre zero e vinte e nove graus de temperatura, humidade entre sessenta e oitenta por cento e concentração de *CO2* até quinhentos partes por milhão. Pretende-se averiguar se o sistema de deteção tem falso positivo com as condições apresentadas.

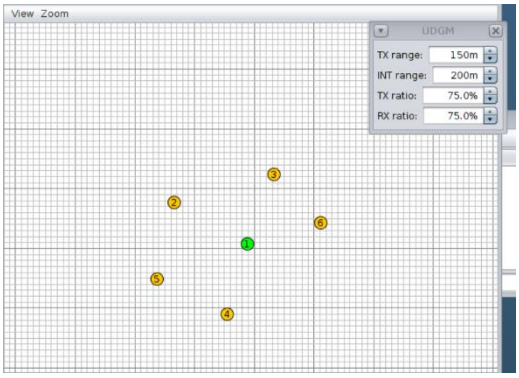

Figura 38 – Constituição da rede com parâmetros normais

13.1.2 Simulação com existência de fogo

Nesta simulação de uma *WSN*, inicialmente começamos com a medição de valores normais, onde foi acrescentado um contador a dois dos quatro dispositivos que compõem o modelo, que através de um contador presente nesses dois, permitirá que quando se alcance 40 interações de medição, os valores correspondentes aos sensores de temperatura, humidade e concentração de *CO2* iniciais passaram a aumentar gradualmente, simulando-se assim o despoletar do incêndio e posterior existência de chamas. Posteriormente vamos avaliar o tempo desde que as leituras foram obtidas até ao alerta no sistema de suporte e verificar o intervalo de tempo de resposta.

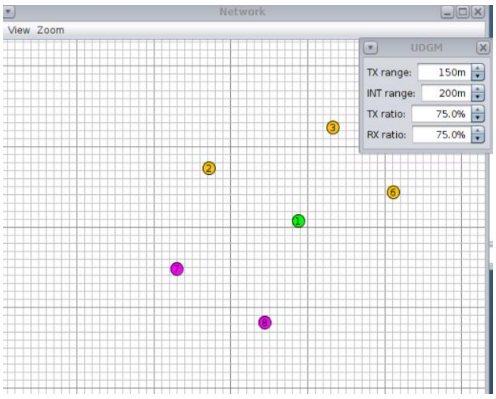

Figura 39 - Constituição da rede com dispositivos que vão detetar incêndio

13.1.3 Simulação onde um ou mais dispositivos ocorre falha

Nesta simulação de uma *WSN*, inicialmente começamos com todos os nós ativos e a efetuar a medição dos valores do ambiente no qual se encontram, sendo estes a temperatura, humidade, concentração *CO2*. Posteriormente, alguns destes dispositivos devido a um contador colocado passaram por ser inativados com o intuito de simular a falha dos mesmos devido a diversos fatores, como roubo, danificação ou simplesmente falha dos mesmos. Esta simulação permite observar o recalculo dos encaminhamentos inicialmente definidos e verificar-se o tempo que os mesmos demoram a recalcular e verificar se no sistema o utilizador é alertado para esse fato.

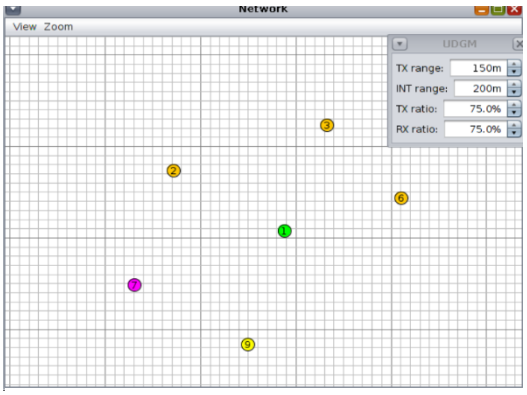

Figura 40 - Constituição da rede com dispositivo que ocorre falha.

13.1.4 Simulação de deteção de risco de incêndio

Nesta simulação da *WSN*, pretende-se que através das sucessivas leituras dos dados recolhidos pelos dispositivos, o sistema de suporte alerte o utilizador para o fato de existir um perigo de incêndio elevado. Para tal, vamos colocar dois dispositivos que compõem a rede, a apresentar temperaturas elevadas e percentagem de humidade baixa. Condições que são ideais para a deflagração de incêndios, sendo que com tais condições é pressuposto que o utilizador seja informado que existe risco. (Bouabdellah, Noureddine, & Larbi, 2013) (Wang, Anderson, & Suddaby, 2015)

14 Análise

14.1 Análise do desempenho

Neste capítulo será analisado o desempenho da rede e sistema de suporte à deteção de fogo florestal referente às simulações efetuadas no capítulo anterior.

A recolha de informação terá duas componentes que se podem analisar, uma é a comunicação entre o dispositivo que pretende enviar os dados e a estação de perímetro que vai depender de quantos saltos serão necessários em que aos quais, será necessário

adicionar-se o tempo de ativo dos vários nós que terão de encaminhar. Enquanto que a outra componente é a resposta do sistema aos eventos que ocorrem.

Como a recolha de informação depende de uma ligação entre a estação de perímetro e o sistema de suporte à deteção, sendo esta via cabo terá sempre um tempo bastante reduzido na transferência dos dados entre os pontos, não fazendo sentido quantificar-se o mesmo. Fazendo sim sentido entre o tempo em que a informação é transmitida e posteriormente o sistema informa o utilizador. Para as simulações onde foi necessário verificar-se o tempo de resposta, foi utilizado um cronómetro para verificar o tempo de reação do sistema.

14.1.1 Análise da simulação sem existência de fogo

Relativamente à primeira simulação, que se baseia na leitura de valores habituais, o sistema não apresentou falso-positivos de deteção de incêndios, bem como não apresentou alertas relativamente ao índice de incêndio, visto tratar-se de valores que estão para o sistema dentro dos parâmetros considerados normais.

Figura 41 - Valores medidos em condições normais

14.1.2 Análise da simulação com existência de fogo

Para o teste da deteção de incêndio foi inicializado um cronómetro ao momento que o dispositivo começou a recolher os dados relativos a um incêndio, que através do contador programado foi após o seu quadragésimo ciclo, onde a temperatura e concentração de CO2 foram gradualmente subindo e o valor de humidade baixou de forma proporcional. Os valores medidos de máximo e mínimo para a temperatura, humidade e concentração de CO2, pode ser visualizada a seguir.

Variável	**Mínimo**	**Máximo**
Temperatura	23 ºC	51 ºC
Humidade	8 %	76 %
Concentração CO2	350 ppm	4100 ppm

Tabela 3 - Valores medidos na deteção de incêndio

Verificou-se que com base nos valores anteriormente mencionados, o sistema levou trinta e dois segundos, até apresentar ao utilizador a informação que estaria a começar ou a deflagrar um incêndio, avisando-o da gravidade do mesmo. A partir desse momento o utilizador tinha três opções que era enviar meio de reconhecimento caso se tratasse de gravidade nível um, ou enviar logo meios de combate ao incendio caso tratasse de nível dois ou três.

Pode-se concluir que o tempo de resposta do sistema é bastante satisfatório, visto que entre o tempo de início do incêndio e o tempo de alerta do sistema, decorreu ~ 30 segundos, permitindo que seja feito o reconhecimento atempado e combate ao mesmo quando ainda mantém proporções pequenas.

14.1.3 Análise da simulação onde um ou mais dispositivos ocorre falha

Nesta simulação pretende-se verificar qual era o tempo que o sistema levava até que detetasse que determinado dispositivo pertencente à rede WSN, já não se encontrava mais a comunicar com o mesmo derivado por motivos desconhecidos. Foi necessário ter em conta que a funcionalidade considera um dispositivo como inativo após 2 minutos, desde a sua última interação, mais concretamente envio de dados recolhidos pelo mesmo. Através da mesma forma utilizada para a simulação anterior, com recurso a um cronómetro foi medido um intervalo de dois minutos e meio entre o dispositivo ter sido desligado e o alerta na página do sistema ao utilizador.

14.1.4 Análise da simulação da deteção de risco de incêndio

Nesta simulação, conforme anteriormente indicado, foram colocados dois dispositivos com código alterado para que simulassem condições ideais para a ignição de fogo. Com base na simulação realizada verificou-se que a página de índices onde é apresentado o índice *FFMC,* apresentou com base nas leituras o alerta ao utilizador de que existia um risco elevado. Sabendo-se que o alerta é dado sempre que o índice se encontra superior a oitenta e nove. Na simulação, os dispositivos obtiveram as seguintes leituras:

Variável	**Mínimo**	**Máximo**
Temperatura	23 ºC	39 ºC
Humidade	55 %	80 %
Concentração CO2	350 ppm	400 ppm

Vento	10 km/h	10 km/h
FFMC	83	89

Tabela 4 - Valores medidos na simulação

Com base na simulação podemos concluir que o sistema teve um desempenho satisfatório, apresentando ao utilizador o alerta do risco que as condições metrológicas apresentavam nas últimas duas horas de dados.

14.2 Estudo da Viabilidade

14.2.1 Técnica

O tema da longevidade é um pouco abstrato visto tratar-se dependente de múltiplas variáveis e abordar múltiplos temas. A longevidade pode ser abordada ao nível da rede, *hardware* e próprio sistema.

Quanto à rede, existe sempre um risco de a mesma ser atacada através da injeção de dados maliciosos ou efetuar um ataque com o intuito de criar o colapso da rede. Apesar desse risco, é necessário também considerar o volume de informação que vai percorrer a mesma, caso existam demasiados dispositivos na mesma rede, pode criar um aumento na latência na transmissão dos dados recolhidos, bem como a um consumo de energia dos dispositivos maior.

Outro tema, é o hardware, estando exposto aos elementos meteorológicos e dependente das condições de luminosidade e bateria, é um ponto com risco elevado de ocorrência de falha. Como forma de mitigar as condições meteorológicas é possível a instalação dos dispositivos em caixas especiais para o efeito, protegendo da entrada de água da chuva o que já elimina um risco. Relativamente às condições de energia reserva-se como forma de mitigação a utilização de uma bateria de elevada capacidade, para que permita ao dispositivo o seu normal funcionamento mesmo quando inibido de luz solar durante largas horas. Para solucionar tal problema, o protótipo conta com uma bateria com capacidade de 6600 *mAh*, que através de um simulador de consumo, considerando-se que o dispositivo e os módulos dos sensores consomem em ativo 120 *mA*, consumo em *stand-by* de 20 *mA*, um ciclo de funcionamento de 30 em 30 segundos permite que funcione através da bateria cerca de 3 dias. (Battery life calculator sleep mode, s.d.) (Youssef_Ismail, 2014)

Relativamente ao sistema de suporte à deteção, não exige grandes necessidades, apenas uma base de dados de alta disponibilidade e um servidor aplicacional *Tomcat*. Quanto à base de dados visto, serem recolhidos inúmeros dados em intervalos de tempo curto, é exigido que a mesma tenha disponível pelo menos 8 *Gigabytes* de memória *RAM*, por forma a ter margem disponível para a escalabilidade da rede *WSN*. Com base num simulador disponível, para o cálculo da quantidade de *RAM*, tendo um cenário de uma rede de 1000 dispositivos e 100 utilizadores todos ligados em simultâneo, vai obrigar a ter disponível pelo menos 3,6 *Gigabytes*. (MySQL Memory Calculator, s.d.) Já para o servidor aplicacional *Tomcat* não necessita de grandes recursos, visto grande parte da aplicação efetuar os cálculos diretamente na base de dados.

14.2.2 Económica

Para alcançar tal objetivo, é proposto a apresentação de um sistema de suporte, composto por uma rede de dispositivos *IoT* e de *Drones*, visto uma rede composta por tais dispositivos permite que seja dinâmica e com um custo de manutenção baixo comparado com outros métodos de deteção existentes no mercado, pois implicam um elevado valor inicial e de manutenção.

Como forma de baixar o custo unitário de cada dispositivo, passamos pela solução de escolher dispositivos que existem à venda globalmente ao público em geral, não tendo assim de se ficar fidelizado a uma empresa e distribuidor, que podem praticar valores como bem entenderem ou efetuar contratos de compra. Desta forma, para além da existência de mais documentação sobre os mesmos relativamente a integração de protocolos e outras soluções, como sensores genéricos de outras empresas, permite mitigar problemas como a empresa abandonar a área de negócio ou deixar de dar suporte ao modelo existente. Um dos exemplos é a empresa *Libelium* que é especializada em soluções *IoT*, em que o seu kit tem custos elevados comparativamente com o exemplo a seguir, bem como os equipamentos disponibilizados serem proprietários.

No caso de se optar por dispositivos baseados na plataforma *Arduino*, apesar de ser pertencente a uma empresa proprietária, é vastamente utilizado para desenvolvimento de múltiplas soluções e protótipos, havendo sempre uma larga comunidade por detrás, o que ajudará a expandir o seu tempo de vida no mercado. Considerando-se o *Arduino*, como dispositivo base eleito e capacitando-o de sensores CO2, sensor de humidade, sensor *GPS* e sensor de temperatura, bateria e painel fotovoltaico, podemos na tabela a

seguir verificar o preço unitário de cada dispositivo. Os preços apresentados são os praticados na sua loja e uma loja especializada à presente data, disponíveis através do endereço https://store.arduino.cc e https://www.cooking-hacks.com/ respetivamente.

Descrição	Preço
Arduino Uno Rev3	19,95€
Communication Shield (XBee, Bluetooth, RFID) - XBee Shield	18,00€
XBee Pro 802.15.4 SMA Module	29,00€
Grove-Temperature & Humidity sensor (SHT31)	13,90€
Internal GPS Antenna	10,00€
GPS Module for Arduino, Raspberry Pi and Intel Galileo	51,00€
CO2 Gas Sensor	45,00€
6600mA/h Rechargeable Battery	36,00€
Solar Charger Shield V2.2	16,20€
1w Solar Panel 80X100	4,20€
Total	243,25 €

Tabela 5 – Custo dos componentes de cada dispositivo

Com base nos valores apresentados, verifica-se que o valor de cada dispositivo será à volta de 243,25, o que numa WSN composta por 100 dispositivos tem um custo de 24325,00€, que caso proporcionem a deteção inicial de incêndios facilmente poderá cobrir os custos de um incêndio que ganhe grandes proporções, obtendo-se assim o retorno do valor inicial investido.

14.2.3 Ambiental

O tema da pegada ambiental é bastante importante atualmente, visto ser prioritária a preservação da natureza existente. Os dispositivos utilizados, visto serem munidos de fontes de energia renovável, devido a serem compostos por um painel fotovoltaico e bateria, ajudam a reduzir o risco de o equipamento ser inativado por falta de energia como sucede com algumas soluções onde não tem posterior possibilidade de recarregar a bateria. Para além disso, devido à existência de um módulo GPS, permitirá que fique registado o local onde o mesmo se encontra, sendo que caso o mesmo, derivado a uma anomalia num dos componentes que compõem o mesmo, provoque a inativação do mesmo, permitirá que posteriormente uma equipa, se desloque ao terreno e verifique se é

viável a reparação do mesmo, permitindo pelo menos a reutilização e reciclagem dos componentes eletrónicos.

Contudo, caso os equipamentos durante o *deploy* por alguma razão sejam danificados, derivado da forma de colocação dos mesmos, tendo estas proporções pequenas não apresentaram um grande impacto na área florestal devido ao tipo de componentes que os compõem, sendo apenas a bateria um dos componentes que criará preocupações. Visto que a mesma poderá libertar químicos tóxicos devido à sua degradação ou por último, em casos de condições extremamente desfavoráveis provocar explosão da mesma, sendo a probabilidade de isto acontecer muito baixa devido às técnicas de construção das mesmas, que impede que tal suceda.

Em jeito de conclusão os benefícios que uma solução como esta proporcionam para prevenir incêndios e detetar atempadamente são superiores ao impacto negativo da degradação dos materiais que compõem o dispositivo, não tendo uma pegada ambiental que se prenuncie relativamente a outros agentes que existam como por exemplo depósito de resíduos químicos e não degradáveis.

15 Conclusão

15.1 Considerações

Para o desenvolvimento do projeto envolveu inicialmente uma ampla investigação sobre o tema das redes *WSN*, tendo incidido numa primeira fase a investigação de projetos semelhantes onde eram aplicados sensores em áreas florestais para monitorização e deteção de incêndio. Foi possível concluir que, existem múltiplas formas de aplicação de monitorização de áreas florestais, sendo que na sua maioria aborda o tema da utilização de sensores *IoT* e meios aéreos não tripulados, que foram os elementos escolhidos para este trabalho, mostrando que o tema é vastamente abordado e contudo com conceções diferentes, no entanto raramente é aplicado no terreno devido ao investimento inicial face aos ganhos observados no curto espaço de tempo. Este ponto é um dos grandes entraves na sua maioria, pois poderá nunca ter retorno da sua aplicação.

Ao desenvolver o projeto, foi necessário analisar os vários problemas existentes que traz a aplicação do mesmo, encontrando-se explicado no capítulo da caracterização, alguns deles foram as distâncias e áreas onde os dispositivos seriam colocados, derivado dos terrenos serem irregulares e de difícil acesso na grande parte dos casos, trazendo como segundo problema o impacto que trazem nas comunicações. Posteriormente ainda no mesmo capítulo foram abordadas quais as variáveis de dados deveriam ser recolhidas, pois existem múltiplas e apenas uma pequena parte podia ser aplicada através de sensores no dispositivo, sem que encarecesse o seu custo. Outro entrave, foi a escolha dos dispositivos que melhor se adaptavam ao projeto em particular, pois existem mais uma vez, múltiplas possibilidades de escolha, desde dos dispositivos base até aos sensores colocados nos mesmos, sendo uns apresentados como mais eficientes com as suas desvantagens e outros com preço mais reduzido, mas igualmente com desvantagens. A escolha da arquitetura a aplicar no projeto foi um dos grandes problemas encontrados no projeto, tendo sido alvo de um custo de tempo assinalável na pesquisa e análise, derivado das particularidades da rede, bem como das distâncias envolvidas e as restrições dos dispositivos em termos de recursos. A segurança foi um dos temas abordados, pois como bem sabemos estamos numa era em que cada vez mais os sistemas são alvos de ataques maliciosos, tornando-se um ponto importante a ter em conta, no entanto neste tipo de redes implica um esforço acrescido aos sensores, para que utilizem métodos de mitigação, tendo um grande impacto na sua eficiência energética, de encaminhamento e de complexidade. Abordou-se também o tema da utilização de soluções *middleware* para

retirar uma parte da complexidade da aplicação de um sistema operativo nos dispositivos de modo a que fosse centralizada a configuração e manutenção dos mesmos. Ligado ao mesmo tema foram abordados múltiplos sistemas operativos existentes no mercado e explicada a forma de arquitetura do escolhido para ser retratado.

Posteriormente, foi explicada a arquitetura aplicada neste sistema, tendo sido escolhida a topologia em malha, com a utilização de uma ou mais estações de perímetro para encaminhamento dos dados, sendo que cada dispositivo presente na rede iria desempenhar dois papeis, o encaminhamento de pacotes de dispositivos vizinhos e a recolha de dados. Para além disso, passou pela utilização do protocolo *6LoWPAN*, para simplificar e ultrapassar as restrições existentes na comunicação dos dados entre dispositivos, derivado do tamanho reduzido dos pacotes.

No capítulo da implementação, foram explicadas as escolhas realizadas em termos da aplicação da componente prática, neste caso o simulador e o sistema de suporte. Relativamente ao simulador exigiu que fosse feita investigação das soluções existentes no mercado e que possibilitassem o seu uso sem restrições de compra de licenças. Para além do problema apresentado, também era necessário cumprir que incluía as tecnologias aqui faladas anteriormente, assim como permitir o desenvolvimento de sistemas personalizados para a execução das simulações. Foram ainda abordadas as tecnologias utilizadas na aplicação do sistema de suporte, que se mostrou necessitar de recorrer a múltiplas *framework* existentes para que proporcionasse ao utilizador um aspeto agradável e com um desempenho constante na iteração com o mesmo, assim como permitisse um desenvolvimento mais rápido e seguindo as boas práticas de programação como código bem organizado.

O capítulo das simulações obrigou, a que fosse ponderado quais as condições que o projeto deveria ser sujeito a avaliação para mostrar o seu desempenho na execução das tarefas para as quais foi desenvolvido. Foi decidida a sua avaliação em quatro situações, uma em que apenas é feita a monitorização com valores normais sem que exista sinais de ocorrência de incêndio para se verificar se o mesmo indicava situações de incendio sem que não existissem. Outra onde ocorre o aparecimento de incêndio, pretendendo-se mostrar a resposta que o sistema tem nestas situações, tanto a nível de tempo como se notificaria o utilizador de tal evento, dando soluções para o seu combate. Outra das simulações pretendia-se mostrar, o que aconteceria caso um dos dispositivos deixasse de comunicar, tornando-se vital numa rede de tais dimensões notificar o utilizador que

existiria problemas com os sensores, de modo a tomarem-se as ações necessárias à sua resolução. Para além de tal objetivo, tinha também verificar-se a reação da rede ao fato de um dos dispositivos que executava o encaminhamento de outros vizinhos, obrigando neste caso, a que os seus vizinhos procurassem outras rotas disponíveis para o envio dos dados até á estação de perímetro. Outra das simulações, era mostrar que quando existem condições de perigo de ignição de incêndio o utilizador do sistema é alertado do mesmo para que sejam tomadas precauções externas, como meios de emergência prontos a intervir caso suceda o cenário de incêndio.

Por último, foi feito o estudo técnico da aplicação do projeto, económico e ambiental, que relativamente à componente técnica são mais uma vez abordadas, as dificuldades existentes na aplicação e soluções para a sua mitigação. A nível económico é apresentado o valor necessário para a implementação de uma rede com as características indicadas no trabalho, sendo apresentado um exemplo de um dispositivo e seus componentes. Este dispositivo foi o escolhido no meio das várias ofertas, derivado da liberdade que o mesmo proporciona a nível de escolha de sistema operativo e componentes a utilizar no mesmo. A nível ambiental, foi abordado o tema do que aconteceria com os dispositivos caso os mesmos se tornassem inutilizáveis derivado de diversos fatores e qual o seu impacto no ambiente que os rodeia.

15.2 Aplicação do projeto

Na aplicação deste projeto, foram sentidas inúmeras dificuldades, apesar dos conhecimentos anteriormente adquiridos, pois foi necessário a escolha de um simulador que preenchesse os requisitos e que oferecesse documentação, para que possibilitasse entender a forma de desenvolver o código dos dispositivos e adaptar às simulações pretendidas. Para além disso, grande parte dos exemplos e documentação existente mostrou-se incompleta ou obsoleta. Grande percentagem do tempo gasto no desenvolvimento da parte prática, foi relativo a tentar encontrar soluções para encontrar um dispositivo que fosse capaz de simular as várias condições e permitir modificar o seu código. Após tal processo, foi necessário encontrar uma forma de se conseguir transmitir os dados provenientes dos sensores e da estação de perímetro do ambiente virtual para o ambiente Windows que serve de anfitrião para a base de dados e o sistema de suporte.

Numa segunda fase, foi necessário encontrar *framework* que permitissem o desenvolvimento do sistema conforme idealizado e com os requisitos necessários, o que

se tornou uma tarefa morosa, visto mostrar-se necessário aprender a utilizar as mesmas, sendo um processo de tentativa erro até entender o seu formato de implementação e moldar aos objetivos do projeto, pois apesar de existirem múltiplos exemplos de implementação nem sempre era os mais corretos ou apresentavam o final pretendido.

Numa terceira fase, a implementação dos índices e cálculo da existência de incêndio foram temas que obrigaram a que fosse realizada investigação quanto a algoritmos de como efetuar o cálculo dos vários índices, existentes na caracterização do risco.

Numa quarta fase foi necessário, mais uma vez, encontrar uma solução para a criação, configuração e implementação do projeto, desta vez dedicado a *web services*. Era necessário que o mesmo fizesse uso da tecnologia *REST*, por ter sido a tecnologia de comunicação escolhida para a realização do trabalho pelas razões já anteriormente indicadas. Para dificultar, era necessário que o mesmo simulasse a estação de perímetro a comunicar com o centro de monitorização, passando por conjugar a ideologia de serviço e implementar a forma de comunicação entre máquina virtual e anfitrião, através de datagramas. Em simultâneo, era exigido que o projeto disponibilizasse um recurso que permitisse receber os dados provenientes da rede *WSN* e de entidades externas, mais propriamente do *IPMA*.

15.3 Evolução futura

Apesar de todo o trabalho desenvolvido até aqui na investigação sobre o tema, execução de um modelo de simulação, abordagem dos protocolos, arquitetura, *middleware e* sistema operativo, não foi possível abordar a arquitetura, protocolos e aplicação dos *drones* neste sistema de suporte, visto ser um trabalho igualmente extenso e bastante exigente no que toca ao tempo requerido de análise. Portanto, esta componente do projeto, será um dos pontos a acrescentar num trabalho futuro. Outra melhoria a ser considerada numa possível evolução do trabalho realizado até aqui, é a aplicação de protocolos mais recentes, assim como novos padrões de comunicação que, entretanto, apareçam disponíveis no mercado, sendo que será sempre necessário avaliar a viabilidade da inclusão dos mesmos. Quanto à viabilidade da adaptação do projeto aos mesmos, terá como peso o consumo de bateria, alcance e compatibilidade com os protocolos utilizados, que são extremamente importantes no tema tratado.

Para além do já indicado anteriormente, torna-se interessante a utilização de um dispositivo físico num próximo desenvolvimento. Este dispositivo selecionado deverá ser semelhante ao apresentado no projeto tanto no seu hardware, como arquitetura utilizada e configuração para que se possa efetuar simulações reais do seu alcance e consumo de energia, o que proporciona uma análise mais minuciosa do desempenho em condições reais com a existência de incêndio de pequenas proporções.

16 Referências

(s.d.). Obtido de Zolertia: https://zolertia.io/

Gomes, J. A. (Novembro de 2010). *Criação de uma rede mesh wireless*, pp. 4-7.

Shelby, Z., & Bormann, C. (2009). *6LoWPAN: The Wireless Embedded Internet.* John Wiley & Sons Ltd.

Abed Alhameed Alkhatib, A., & Singh Baicher, G. (2012). International Conference on Computer Networks and Communication Systems. *Wireless Sensor Network Architecture*.

Afzal, M., Yu, H., Rehmani, M., & Zikria, Y. (Agosto de 2018). *Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques*, pp. 699 - 707.

Ahmed, N., Rahman, H., & Hussain, M. (6 de Agosto de 2016). *A comparison of 802.11ah and 802.15.4 for IoT*.

Alkhatib, A., & Baicher, G. (2012). 2012 International Conference on Computer Networks and Communication Systems (CNCS 2012). *Wireless Sensor Network Architecture*.

Battery life calculator sleep mode. (s.d.). Obtido de Geekstips: https://www.geekstips.com/battery-life-calculator-sleep-mode/

Bhosle, A., & M. Gavhane, L. (2016). Forest Disaster management with Wireless Sensor Network. *International Conference on Electrical, Electronics, and Optimization Techniques*, p. 3. Obtido de Forest Disaster management with Wireless Sensor Network.

Bouabdellah, K., Noureddine, H., & Larbi, S. (2013). The 3rd International Conference on Sustainable Energy Information Technology. *Using Wireless Sensor Networks for Reliable Forest Fires*, pp. 794 – 801.

Briscoe, N. (Julho de 2000). Understanding The OSI 7-Layer Model. *PC Network Advisor*, pp. 13 - 14.

Buratti, C., Conti, A., Dardari, D., & Verdone, R. (31 de Agosto de 2009). *An Overview on Wireless Sensor Networks Technology and Evolution*.

Caldeira, A. (Janeiro de 2017). Aula 4: Plataformas Cliente/Servidor.

Carlotti, T., Silvani, X., Innocenti, E., Morandini, F., Bulté, N., & Dang, T. (4 de Janeiro de 2019). An OCARI-Based Wireless Sensor Network for Heat.

Chowdhury, A. H., Ikram, M., Cha, H.-S., Redwan, H., Shams, S., Kim, K.-H., & Yoo, S.-W. (Janeiro de 2009). *Route-over vs Mesh-under Routing in 6LoWPAN*, pp. 1-5.

Diferença entre Unicast, multicast e broadcast. (16 de Dezembro de 2011). Obtido de No mundo das redes: http://nomundodasredes.blogspot.com/2011/12/diferenca-entre-unicast-multicast-e.html

Ee, G., Ng, C., Noordin, N., & Ali, B. (2010). *A review of 6LoWPAN Routing Protocols*, pp. 71-81.

Eugenio, F., Santos, A., Fiedler, N., Ribeiro, G., Silva, A., Domingues, G., . . . Martins, L. (22 de Abril de 2016). *GIS applied to location of fires detection towers in domain area of tropical forest*, pp. 542-549.

Figueroa, P., Pérez, J., & Amezcua, I. (16 de Maio de 2017). *Performance evaluation of lightweight and secure protocol for wireless sensor networks: A protocol to enableWeb services in IPv6 over low-power wireless personal area networks*.

Fok, C.-L., Roman, G.-C., & Lu, C. (01 de 1 de 2006). *Agilla: A Mobile Agent Middleware for Sensor Networks*.

Framework. (26 de Maio de 2019). Obtido de Wikipédia: https://pt.wikipedia.org/wiki/Framework

Fresnel Zones – What are they and why are they so important? (25 de Julho de 2019). Obtido de Digitalair: https://www.digitalairwireless.com/articles/blog/fresnel-zones-what-are-they-and-why-are-they-so-important

Glissa, G., & Meddeb, A. (2 de Fevereiro de 2018). *6LowPSec: An end-to-end security protocol for 6LoWPAN*, pp. 100-112.

Hariyawan, M. Y., Gunawan, A., & Putra, E. H. (2013). *Wireless Sensor Network for Forest Fire Detection.* TELKOMNIKA.

Hariyawan, M., Gunawan, A., & Putra, E. (Setembro de 2013). TELKOMNIKA. *Wireless Sensor Network for Forest Fire Detection*, pp. 563-574.

Harrison, C. (2016). *Introduction to Wireless Sensor Networks*. Obtido em 29 de 01 de 2019, de https://slideplayer.com/slide/7258153/

Hibernate. (31 de Maio de 2019). Obtido de Wikipédia: https://pt.wikipedia.org/wiki/Hibernate

Introduction. (s.d.). Obtido de Apache Maven Project: https://maven.apache.org/what-is-maven.html

IPMA. (2017). *IPMA API*. Obtido de Interface de Programação de Aplicações do IPMA: https://api.ipma.pt/

Kaur, G., & Manshahia, M. S. (2017). *Wireless Sensor Networks for Fire Detection and Control*. IJFRCSCE.

Krull, W., Tobera, R., Willms, I., Essen, H., & von Wahl, N. (2012). Early forest fire detection and verification using optical smoke, gas and microwave sensors. pp. 584-594.

Krull, W., Tobera, R., Willms, I., Essen, H., & Wahl, N. (2012). *Early forest fire detection and verification using optical smoke, gas and microwaves sensores*.

Lignan, A. (23 de Junho de 2015). *Zolertia z1 motes*. Obtido de Github: https://github.com/contiki-os/contiki/wiki/Zolertia-z1-motes

Linnartz, J.-P. (1995). *Shadowing*. Obtido de Wireless Communication: http://www.wirelesscommunication.nl/reference/chaptr03/shadow/shadow.htm

Ma, Z. (Dezembro de 2017). Cross-Layer Design in Sensor Networks: Issues and Possible Solutions.

Marques da Silva, M. (2016). Cable and Wireless Networks: Theory & Practice. CRC Press.

Martínez, L. (26 de Julho de 2017). *IoT for detecting wildfires*. Obtido de https://www.amplia-iiot.com/iot-detecting-wildfires/

Multithreading. (s.d.). Obtido de Techopedia: https://www.techopedia.com/definition/24297/multithreading-computer-architecture

Multithreading (computer architecture). (s.d.). Obtido de Wikipedia: https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)

MySQL. (14 de Junho de 2019). Obtido de Wikipédia: https://pt.wikipedia.org/wiki/MySQL

MySQL Memory Calculator. (s.d.). Obtido de MySQL Calculator: https://www.mysqlcalculator.com/

O que é Internet das Coisas? (23 de Junho de 2019). Obtido de Hewlett Packard: https://www.hpe.com/br/pt/what-is/internet-of-things.html

O que são redes Ad Hoc e quais as suas vantagens? (25 de Julho de 2019). Obtido de FaqInformatica: https://faqinformatica.com/o-que-e-uma-rede-ad-hoc-quais-vantagens/

Olsson, J. (2014). *6LoWPAN desmystified*.

Padmavathi, G., & Shanmugapriya, D. (2009). A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks. *(IJCSIS) International Journal of Computer Science and Information Security*.

Photodiode Working Principle, Characteristics and Applications. (s.d.). Obtido de Elprocus: https://www.elprocus.com/photodiode-working-principle-applications/

Primefaces. (15 de Junho de 2019). Obtido de Wikipédia: https://en.wikipedia.org/wiki/PrimeFaces

Pugliese, M., Pomante, L., & Santucci, F. (Outubro de 2009). Agent-based scalable design of a cross-layer security framework for Wireless Sensor Networks Monitoring Applications.

Rodriguez, T. (9 de Setembro de 2018). *Understanding Event-Driven Architectures (EDA): the paradigm of the future*. Obtido de Medium: https://medium.com/drill/understanding-event-driven-architectures-eda-the-paradigm-of-the-future-7ae632f056bb

Rouse, M. (Fevereiro de 2019). *SOAP (Simple Object Access Protocol)*. Obtido de SearchMicroservices: https://searchmicroservices.techtarget.com/definition/SOAP-Simple-Object-Access-Protocol

Sensores de deslocamento (MEC113). (s.d.). Obtido de Instituto Newton C. Braga : http://www.newtoncbraga.com.br/index.php/robotica/5383-mec113

Sharma, A., Siddiqui, F., Baig, M., & Ansari, F. (Maio de 2017). *IOT ENABLED FOREST FIRE DETECTION AND ONLINE MONITORING SYSTEM*, pp. 50-54.

Silva, J., Rodrigues, J., Al-Muhtadi, J., Rabêlo, R., & Furtado, V. (7 de Fevereiro de 2019). *Management Platforms and Protocols for Internet of Things: A Survey*.

Solobera, J. (4 de Setembro de 2010). *Detecting Forest Fires using Wireless Sensor Networks*. Obtido de http://www.libelium.com/wireless_sensor_networks_to_detec_forest_fires/

Sudha, P., & Murugan, A. (2017). *Detection of Forest Fire using Dezert-Smarandache*.

Teguh, R., Honma, T., Usop, A., Shin, H., & Igarashi, H. (Julho de 2012). Detection and Verification of Potential Peat Fire Using Wireless Sensor Network and UAV.

The Contiki Operating System. (s.d.). Obtido de Github: https://github.com/contiki-os/contiki

Toledo-Castro, J., Santos-González, I., Hernández-Goya, C., & Caballero-Gil, P. (2017). UBICOMM 2017 : The Eleventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. *Management of Forest Fires Using IoT Devices*, pp. 121-126.

Tranter, W., Taylor, D., Ziemer, R., Maxemchuk, N., & Mark, J. (2007). *The Best of the Best: Fifty Years of Communications and Networking Research.* John Wiley & Sons, Inc.

Tsetsos, V., Sekkas, O., Tsoublekas, G., Hadjieythymiades, S., & Zervas, E. (2012). *A Forest Fire Detection System: The Meleager Approach*, pp. 1-7.

Wang, M.-M., Cao, J.-N., Li, J., & K. Dasi, S. (Maio de 2008). JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY. *Middleware for Wireless Sensor Networks: A Survey*, pp. 305 - 326.

Wang, Y., Anderson, K., & Suddaby, R. (2015). CANADIAN FOREST SERVICE. *Updated source code for calculating fire danger indices in the Canadian Forest Fire Weather Index System*, pp. 17 - 20.

What is Java technology and why do I need it? (s.d.). Obtido de Java: https://www.java.com/en/download/faq/whatis_java.xml

What is REST? (s.d.). Obtido de Codecademy: https://www.codecademy.com/articles/what-is-rest

What is the Fresnel Zone? (25 de Julho de 2019). Obtido de everything RF: https://www.everythingrf.com/community/what-is-the-fresnel-zone

Xu, Y.-H., Sun, Q.-Y., & Xiao, Y.-T. (19 de Outubro de 2018). An Environmentally Aware Scheme of Wireless Sensor Networks for Forest Fire Monitoring and Detection. *Future Internet*.

Youssef_Ismail. (31 de Agosto de 2014). *Arduino Uno Power Consumption*. Obtido de Arduino: https://forum.arduino.cc/index.php?topic=264083.0