



UNIVERSIDADE ESTADUAL PAULISTA

“JÚLIO DE MESQUITA FILHO”

Instituto de Ciência e Tecnologia de Sorocaba

Campus de São João da Boa Vista

NELSON CAMILO DE ALMEIDA

**Rede Híbrida LoRa em Malha – LoRaWAN para
Aplicações de Internet das Coisas**

Sorocaba

2022

NELSON CAMILO DE ALMEIDA

**Rede Híbrida *LoRa* em Malha – *LoRaWAN*
para Aplicações de *Internet* das Coisas**

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica, junto ao Programa de Pós-Graduação em Engenharia Elétrica, Interunidades, entre o Instituto de Ciência e Tecnologia de Sorocaba e o Campus de São João da Boa Vista da Universidade Estadual Paulista "Júlio de Mesquita Filho".

Área de Concentração: Automação

Linha de Pesquisa: Mecatrônica

Orientador: Prof. Dr. Eduardo Paciência Godoy

Sorocaba

2022

A447r Almeida, Nelson Camilo de
Rede Híbrida LoRa em Malha – LoRaWAN para Aplicações de Internet das Coisas / Nelson Camilo de Almeida. -- Sorocaba, 2022
84 p. : il., tabs., fotos, mapas

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Eduardo Paciência Godoy

1. Engenharia elétrica. 2. Automação. 3. Internet das coisas. I.
Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

As redes de comunicação LoRaWAN são uma tecnologia de comunicação sem fio que permite o envio de dados por longas distâncias. Assim, a pesquisa sobre estas redes pode trazer muitos benefícios para a sociedade, impactando na melhora da gestão de recursos como: naturais, resíduos, tráfego, energia, saúde e bem-estar e a eficiência da agricultura.

LoRaWAN communication networks are a wireless communication technology that allows data to be sent over long distances. Thus, research on these networks can bring many benefits to society, improving the management of resources such as: natural resources, waste, traffic, energy, health and well-being and the efficiency of agriculture.



UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Sorocaba


CERTIFICADO DE APROVAÇÃO


TÍTULO DA DISSERTAÇÃO: Rede Híbrida LoRa em Malha – LoRaWAN para Aplicações de Internet das Coisas

AUTOR: NELSON CAMILO DE ALMEIDA

ORIENTADOR: EDUARDO PACIÊNCIA GODOY

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área: Automação pela Comissão Examinadora:


Prof. Dr. EDUARDO PACIÊNCIA GODOY (Participação Presencial)
Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia - UNESP - Câmpus de Sorocaba


Prof. Dr. DHIEGO FERNANDES CARVALHO (Participação Presencial)
Departamento de Eng. Controle e Automação / Universidade Estadual Paulista - UNESP - Campus de Sorocaba - ICTS


Prof. Dr. ROBERTO SILVA NETTO (Participação Presencial)
Departamento de Engenharia Elétrica / Centro Universitário FACENS

Sorocaba, 03 de novembro de 2022

Agradecimentos

Nesta caminhada de mestrado, agradeço primeiramente a Deus por permitir termos força para continuar caminhando! Agradeço a minha esposa Fabiane que com sua perspicácia possibilitou que eu começasse essa caminhada que agora se completa. A meus filhos, Diego e Juliana, que não entendem ainda os motivos de tanto trabalho. Gostaria de enviar um abraço forte ao Prof. Dr. Eduardo Paciência de Godoy por sempre indicar o melhor caminho a seguir, que com sua visão sabe conduzir muito bem os trabalhos e ainda por apresentar um mundo novo de possibilidades que foi para mim. Aos colegas do Laboratório Gasi, que sempre me apoiaram com tantas boas palavras e sempre ajudando a todos para seguir no caminho: Rodrigo Rolle, Maria Fernanda, Murilo Ferreira, Ricardo Pasquati, José Neto, Marcos Fábio e tantos outros em suas maneiras. Um agradecimento especial à Prof. Dra. Juliana Verona, diretora da Fatec de Itu, por abrir espaço para os trabalhos e a todos da Fatec que fixaram seus pensamentos de apoio.

*"A mente que gera o problema
é a mesma mente que cria a solução"
(Autor desconhecido)*

Resumo

A *Internet das Coisas (IoT)* possibilita o sensoriamento e controle remotos, obtenção e análise de dados e deve permitir a conexão de mais dispositivos. Uma infraestrutura de rede de comunicação é necessário para habilitar a troca de informações entre os elementos de uma aplicação de *IoT*. Neste sentido, várias tecnologias de comunicação em rede foram desenvolvidas e têm sido utilizadas para este fim. Entre essas tecnologias, as recentes *LPWANs* que permitem conexões de rede sem fio de longa distância e baixo consumo de energia em dispositivos normalmente conectados a baterias, porém, com baixas taxas de transmissão de dados. As tecnologias *LPWANs* mais comuns são: *SigFox*, *NB-IoT* e *LoRaWAN*, sendo que todas estas enfrentam um problema de aplicação em comum: a presença de obstáculos e localização topográfica dos dispositivos originam áreas de sombra (sem cobertura) dentro da área de cobertura de comunicação. A topologia utilizada nessas redes (estrela-de-estrelas) agrava ainda mais esta situação. Neste sentido, faz-se necessário buscar alternativas para melhorar a cobertura dessas redes e a possibilidade de superar estes obstáculos permitindo assim que áreas, antes de sombra, possam ser sensoreadas permitindo um melhor aproveitamento dos protocolos de longa distância. Este trabalho apresenta uma solução para este problema através do desenvolvimento de uma rede híbrida composta pela integração de uma rede *LoRa* em malha com uma rede *LoRaWAN*. Essa rede híbrida é baseada em um *hardware* de baixo custo, chamado de *Proxy* que coordena uma rede *LoRa* em malha e conecta, de forma transparente, os dispositivos dessa rede à rede *LoRaWAN*. Essa solução habilita a comunicação de dispositivos em áreas de sombra da cobertura *LoRaWAN*, além da possibilidade de estender a área de cobertura *LoRaWAN* através da rede *LoRa* em malha.

Palavras-chave: *LPWAN*, *LoRaWAN*, Cidades Inteligentes, Áreas de sombra, *LoRa*, malha.

Abstract

The Internet of Things (IoT) enables remote sensing and control, data obtaining and analysis, and should allow the connection of more devices. An infrastructure of network communication is necessary to promote the exchange of information between the elements of an IoT application. In this sense, several network communication technologies have been developed and have been used for this purpose. Among these technologies, the most recent LPWANs enable low-power, long-distance wireless network connections in devices normally connected to batteries, however, with low data rates of transmission. The most common LPWANs technologies are SigFox, NB-IoT, and LoRaWAN, all of which face a common application problem: the presence of obstacles and topographic location of devices give rise to shadow areas (no coverage) within the communication coverage area. The topology used in these networks (star-of-stars) makes this situation even worse. In this sense, it is necessary to seek alternatives to improve the coverage of these networks and the possibility of overcoming these obstacles thus allowing areas, before shadow, to be sensed allowing better use of long-distance protocols. This job presents a solution to this problem by developing a Hybrid Network composed of the integration of a LoRa Mesh network with a LoRaWAN network. That hybrid network is based on low-cost hardware, called a Proxy that coordinates a LoRa Mesh Network and transparently connects the devices on that network to the LoRaWAN network. This solution enables device communication in shadow areas of LoRaWAN coverage, in addition to the possibility of extending the LoRaWAN coverage area through the LoRa Mesh network.

Keywords: LPWAN, LoRaWAN, Smart Cities, Shadow areas, LoRa, Mesh.

Lista de ilustrações

Figura 1 – Infraestrutura em malha: (a) infraestrutura/ <i>backbone</i> , (b) roteadores em malha, (c) clientes em malha, (d) <i>WMNs</i> Híbridas.	16
Figura 2 – Pacote <i>LoRa</i>	18
Figura 3 – Distribuição de canais para a América do Norte.	19
Figura 4 – Tecnologia da pilha <i>LoRa WAN</i>	20
Figura 5 – Janelas de recepção <i>LoRa WAN</i>	20
Figura 6 – Janelas de recepção <i>LoRa WAN</i> Classe B.	21
Figura 7 – Janelas de recepção <i>LoRa WAN</i> Classe C.	21
Figura 8 – Topologia da rede <i>LoRa WAN</i>	22
Figura 9 – Segurança da rede <i>LoRa WAN</i>	23
Figura 10 – Ativação da rede <i>LoRa WAN</i> : (a) <i>OTAA</i> e (b) <i>ABP</i>	24
Figura 11 – Diagrama de funcionamento do <i>e-Node</i> quando em ativação <i>ABP</i>	30
Figura 12 – Apresentação da rede <i>LoRa Mesh / LoRAWAN</i>	33
Figura 13 – Módulo <i>Heltec ESP32 Wi-Fi LoRa 32</i>	34
Figura 14 – Módulo <i>NideRF SX1276</i>	35
Figura 15 – Módulo <i>Espressif DevKitC</i>	35
Figura 16 – <i>Gateway LoRaWAN RD43HAT / RD43HATGPS</i>	36
Figura 17 – <i>Gateway LoRaWAN</i> - Unesp Sorocaba: (a) <i>Gateway</i> instalado e (b) antena externa.	37
Figura 18 – <i>ChirpStack interface</i> de controle.	38
Figura 19 – <i>ChirpStack</i> diagrama de blocos.	39
Figura 20 – Exemplo de <i>Gateway LoRaWAN ChirpStack</i>	40
Figura 21 – Estado dos serviços do <i>Gateway LoRaWAN ChirpStack</i> em funcionamento.	41
Figura 22 – Diagrama de blocos dos dispositivos em malha.	42
Figura 23 – Detalhes dos dispositivos <i>LoRa</i> em malha: a) sensor DHT11, b) identificador e c) conexões internas.	43
Figura 24 – Parâmetros dos dispositivos <i>LoRa</i> em Malha.	44
Figura 25 – Algoritmo dos dispositivos da Malha.	45
Figura 26 – Rota do dispositivo 0xB4 para o <i>Proxy</i>	46
Figura 27 – Pacote de dados do dispositivo 0xB4.	46
Figura 28 – Diagrama de blocos do <i>Proxy</i>	47
Figura 29 – (a) <i>Proxy</i> em campo e (b) aspectos construtivos.	49
Figura 30 – Algoritmo de gerenciamento da rede em malha.	52
Figura 31 – Algoritmo de reencaminhamento do <i>Proxy</i>	54
Figura 32 – Virtualização de dispositivos no <i>Proxy</i>	55
Figura 33 – Criando uma Aplicação no <i>ChirpStack</i>	56

Figura 34 – Adicionando um dispositivo da malha no <i>ChirpStack</i>	57
Figura 35 – Cadastro das credenciais LoRaWAN no <i>ChirpStack</i>	58
Figura 36 – Dispositivos cadastrados no <i>ChirpStack</i> para o teste da rede em malha.	58
Figura 37 – Dispositivo <i>Proxy</i> interligando a rede <i>LoRa</i> em malha a um <i>gateway LoRaWAN</i>	59
Figura 38 – Configuração da Integração do <i>LoRaWAN</i> ao Banco de Dados <i>InfluxDB</i>	60
Figura 39 – Configuração da Integração do <i>LoRaWAN</i> ao Banco de Dados <i>InfluxDB</i> e indicação da função <i>Decode</i>	61
Figura 40 – <i>Script</i> em <i>Python</i> que envia os dados ao Banco de Dados <i>InfluxDB</i>	61
Figura 41 – Visualização dos dados apresentado pelo <i>Grafana</i>	62
Figura 42 – Teste de cobertura da rede <i>LoRaWAN</i> e o posicionamento do <i>Proxy</i>	63
Figura 43 – Posicionamento do <i>Proxy</i> para teste da Rede Híbrida.	64
Figura 44 – Áreas de cobertura (a), sombra (b) para a rede <i>LoRaWAN</i> e o posicionamento do <i>Proxy</i>	65
Figura 45 – Posicionamento dos dispositivos para o teste da rede <i>LoRa</i> em malha.	67
Figura 46 – Visualização dos pacotes <i>LoRaWAN</i> dentro da <i>interface</i> do <i>ChirpStack</i> destacando em vermelho o <i>DevAddr</i> recebido.	68
Figura 47 – Demonstração do tempo utilizado pelo <i>Proxy</i> para gerenciar a criação de uma rede com 2 vizinhos. (a) início do processo de criação da rede, (b) descoberta do primeiro vizinho: 0xB4, (c) descoberta do segundo vizinho: 0xDE, (d) final da descoberta de vizinhos e rotas válidas e (e) custo das rotas criadas.	70
Figura 48 – No detalhe pode ser visto os dispositivos vizinhos do dispositivo ndD2.	71
Figura 49 – Aplicação de teste da rede <i>LoRa</i> em Malha apresentada pelo <i>Grafana</i>	72
Figura 50 – Rendimento dos dispositivos da Rede <i>LoRa</i> em Malha.	74

Lista de tabelas

Tabela 1 – Comparação entre <i>LoRa SF</i>	18
Tabela 2 – Taxa de rendimento dos dispositivos da malha.	73
Tabela 3 – Taxa de <i>PDR</i> da rede Híbrida <i>LoRa</i> em Malha / <i>LoRaWAN</i>	74

Lista de abreviaturas e siglas

ABP	<i>Activation By Personalization</i>
ADR	<i>Adaptative Data Rate</i>
AES	<i>Advanced Encryption Standard</i>
AODV	<i>Ad-hoc On-Demand Distance Vector Routing</i>
AppKey	<i>Application Key</i>
AppSKey	<i>Application Session Key</i>
API	<i>Application Programming Intercafe</i>
AS	<i>Application Server</i>
AU915	<i>Australian 915MHz band</i>
AWS	<i>Amazon Web Server</i>
BLE	<i>Bluetooth Low Energy</i>
BW	<i>Band width</i>
CR	<i>Code Rate</i>
CRC	<i>Cyclic Redundancy Check</i>
CSS	<i>Chirp Spread Spectrum</i>
DevAddr	<i>Device Address</i>
DevEUI	<i>Device End-device Unique Identifier</i>
DSDV	<i>Destination Sequenced Distance Vector</i>
DSR	<i>Dynamic Source Routing</i>
EU868	<i>European 868MHz band</i>
EUI	<i>End User Identifier</i>
FIFO	<i>First In First Out</i>
Gasi	Grupo de Automação e Sistemas Integráveis

GPS	<i>Global Position System</i>
gRPC	<i>Google Remote Procedure Call</i>
HWMP	<i>Hybrid Wireless Mesh Protocol</i>
ID	<i>Identifier</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IoT	<i>Internet of Things</i>
ISM	<i>Industrial Scientific Medical</i>
LMIC	<i>LoRaWAN Made In C</i>
LoRa	<i>Long Range</i>
LoRaWAN	<i>Long Range Wide Area Network</i>
LPWAN	<i>Low Power Wide Area Network</i>
MAC	<i>Media Access Control</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
MPR	<i>Multi-path Relays</i>
NB-IoT	<i>Narrow Band for Internet of Things</i>
NS	<i>Network Server</i>
NwkSKey	<i>Network Session Key</i>
OLED	<i>Organic Light Emitter Diode</i>
OTAA	<i>Over-The-Air Activation</i>
PHY	<i>Physical Layer</i>
REST	<i>Representational State Transfer</i>
RF	<i>Radio Frequency</i>
RFID	<i>Radio Frequency Identification</i>
RREP	<i>Route Reply</i>
RREQ	<i>Route Request</i>

RSSI	<i>Received Signal Strength Indicator</i>
SF	<i>Spread Factor</i>
SNR	<i>Signal-to-noise Ratio</i>
TOA	<i>Time on Air</i>
TTN	<i>The Things Network</i>
UDP	<i>User Datagram Protocol</i>
US902	<i>United States 902MHz band</i>
VPS	<i>Virtual Private Server</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
WMN	<i>Wireless Mesh Networks</i>

Sumário

1	INTRODUÇÃO	11
1.1	JUSTIFICATIVA	11
1.2	OBJETIVO	13
1.3	ESTRUTURA E CONTEÚDO	14
2	CONCEITOS RELACIONADOS	15
2.1	Rede em Malha	15
2.1.1	Arquitetura em Malha	15
2.2	<i>LoRa</i>	16
2.3	<i>LoRaWAN</i>	19
2.3.1	Características do protocolo	19
2.3.2	Segurança	23
2.3.3	Métodos de ativação	24
2.3.3.1	Ativação <i>OTAA</i>	25
2.3.3.2	Ativação <i>ABP</i>	25
3	REVISÃO BIBLIOGRÁFICA	26
4	REDE HÍBRIDA <i>LORA</i> EM MALHA / <i>LORAWAN</i>	32
4.1	Proposta do Trabalho	32
4.2	Materiais	33
4.2.1	Dispositivo <i>LoRa</i> em Malha	33
4.2.2	Módulo transceptor <i>LoRa SX1276</i>	34
4.2.3	Módulo <i>Espressif ESP32-DevKitC</i>	34
4.3	Infraestrutura <i>LoRaWAN</i>	35
4.4	Servidor de Rede e de Aplicação <i>ChirpStack - GatewayOS</i>	37
5	DESENVOLVIMENTO	42
5.1	Dispositivos <i>LoRa</i> em malha	42
5.1.1	Algoritmo dos dispositivos <i>LoRa</i> em Malha	44
5.2	Dispositivo <i>Proxy</i>	46
5.3	Integração <i>LoRa</i> em Malha – <i>LoRaWAN</i>	48
5.3.1	Algoritmo gerenciador da rede <i>LoRa</i> em Malha	50
5.3.2	Algoritmo redirecionador: <i>Proxy</i>	51
5.3.3	Virtualização dos dispositivos da malha para a rede <i>LoRaWAN</i>	54
5.4	Infraestrutura <i>LoRaWAN (ChirpStack)</i>	55

5.5	Aplicação para visualização dos dados da Rede em Malha	59
6	RESULTADOS	63
6.1	Verificação da comunicação <i>LoRaWAN</i> utilizando o <i>ChirpStack</i> . .	67
6.2	Testes de Comunicação e Validação da Rede <i>LoRa</i> em Malha . . .	69
6.3	Testes de Comunicação da Rede Híbrida <i>LoRa</i> em Malha / <i>LoRaWAN</i>	72
6.3.1	Taxa de entrega de pacotes - <i>PDR</i> dos dispositivos da malha	73
6.3.2	Taxa de entrega de pacotes - <i>PDR</i> da Rede Híbrida <i>LoRaWAN</i>	74
7	CONCLUSÕES	75
7.1	Proposta de melhorias	77
	REFERÊNCIAS	78

1 INTRODUÇÃO

Em sensoriamento, utilizando redes de sensores sem fio, principalmente utilizando redes *LoRaWAN*, fatores externos podem fazer com que apareçam áreas de sombra para esta rede, mesmo estando próximos a um *gateway* pode não ser possível o envio de dados entre um dispositivo e um *gateway LoRaWAN*. O presente trabalho apresenta uma solução para estas problemática e através deste desenvolvimento outras melhorias são apontadas.

1.1 JUSTIFICATIVA

O termo *IoT* (*Internet of Things*), conhecido no Brasil como *Internet das Coisas*, refere-se à interligação de objetos físicos e virtuais em rede permitindo que ocorra a coleta, compartilhamento e armazenamento de uma enorme quantidade de dados. Tais dados podem ser processados e analisados bem como podem fornecer inúmeras informações que podem prover serviços em uma escala até então nunca imaginada. Esta tecnologia *IoT* propicia que a informação esteja disponível em qualquer lugar, a qualquer momento, em qualquer situação e de forma integrada às atividades do cotidiano (TUDO... , 2015). Desta forma, a tecnologia *IoT* trabalha com sistemas que permitem a conexão de dispositivos com identificação por rádio frequência (*RFID - Radio-frequency Identification*), onde estes sistemas abrangem vários tipos de tecnologias de comunicação que estão presentes nas mais diversas aplicações (LUVISOTTO et al., 2018).

Atualmente com o acentuado crescimento de tecnologias *IoT*, muitas aplicações podem ser encontradas em diversos campos de aplicação, tais como segurança, agricultura, medição inteligente, *smart cities*, *smart homes*, *smart wearables* entre outros e principalmente como pode ser consumida essa grande quantidade de informação (MEKKI et al., 2018). Tal conectividade foi viabilizada pelo crescimento das redes sem fio que torna o acesso e transmissão de dados para a *internet* facilitados.

As aplicações em *IoT* são caracterizadas principalmente como um longo alcance, baixa transferência de dados, baixo consumo de energia e devem ser economicamente viáveis. Alguns sistemas de comunicação sem fio de baixo alcance, tais como *ZigBee*, *Bluetooth* e *Wi-Fi* trabalham em áreas pequenas a centenas de metros, já as tecnologias de comunicação de rede celular 3G e 4G, fornecem um longo alcance, mas exigem um consumo elevado de energia e alto custo (SINHA; WEI; HWANG, 2017).

As redes citadas e o avanço de aplicações de *IoT* tem-se impulsionado após o ano de 2013 (SINHA; WEI; HWANG, 2017), o desenvolvimento de novas formas de comunicação sem fio, como as *LPWANs* (*Low Power Area Networks*) (MEKKI et al., 2018)

que surgiram como boas alternativas frente às aplicações de *IoT* (USMAN; KULKARNI; SOORIYABANDARA, 2017). Essas aplicações *IoT* têm sido utilizadas de forma mais acentuada tanto no setor industrial como na comunidade acadêmica (MEKKI et al., 2018).

Entre as principais *LPWANs* podem-se destacar *LoRaWAN* (*Long Range Wide Area Network*), *NB-IoT* (*Narrow Band for Internet of Things*), *SigFox* entre outras. Estas redes se caracterizam por possuírem um equilíbrio satisfatório entre distância e alcance de comunicação podendo chegar a quilômetros com um baixo consumo de energia, permitindo uma vida útil extremamente longa de baterias que pode chegar a anos de duração. Tais características estão sendo recentemente empregadas em aplicações de *IoT* (USMAN; KULKARNI; SOORIYABANDARA, 2017), (SANCHEZ-IBORRA; CANO, 2017). Embora várias tecnologias baseadas em *LPWAN* estejam sendo desenvolvidas, no Brasil observa-se um avanço e disponibilidade de uso da rede *LoRaWAN* mas, as redes *SigFox* e *NB-IoT* utilizam, no Brasil, rede baseadas nas operadoras de celular, o que introduz custos altos. Já a rede *LoRaWAN* utiliza comunicação em bandas sub-GHz não licenciadas e de uso livre, oferecendo um custo mais baixo.

LoRaWAN utiliza o protocolo *LoRa*, acrônimo de *Long Range* (Longo Alcance) e *NB-IoT* utiliza um protocolo de banda estreita (SINHA; WEI; HWANG, 2017). O autor apresenta um roteiro do funcionamento da rede em malha desenvolvida e as características do *gateway LoRaWAN* utilizado *et. al.*, algumas destas características podem ser usadas para escolher qual tecnologia *LPWAN* utilizar e indica que o *LoRa* pode ser utilizado junto a aplicativos de baixo custo enquanto o *NB-IoT* é direcionado para aplicações que exigem alta qualidade de serviço e baixa latência.

O protocolo *LoRa* possibilita um longo alcance de transmissão de dados, da ordem de alguns quilômetros (km) em ambientes urbanos e 15km em ambiente rural (ABRARDO et al., 2019) e para sua utilização se faz necessário a implantação de uma certa infraestrutura como os dispositivos (*end points*) onde são conectados sensores ou atuadores, os *gateways* que fazem a interconexão de rádio entre os dispositivos até o gerenciador de rede permitindo acesso até a aplicação do usuário utilizando servidores dedicados, como por exemplo a rede *TTN* (*The Things Network*) que é uma rede *IoT* aberta e global mantida pelos seus usuários, a *American Tower ATC* que é uma rede particular *LoRaWAN* e até infraestruturas locais como é o caso do Servidor de Rede e Aplicação *ChirpStack*.

Nos *gateways* são instaladas antenas que proporcionam uma boa área de cobertura para o *link* de dados *LoRa* proveniente dos dispositivos. Estes *gateways* devem ainda ter conexão com a *internet*, seja ela *Wi-Fi*, *ethernet* ou celular 3G/4G. Porém, algumas situações podem dificultar o alcance dos *gateways LoRaWAN*, tais como: a presença de obstáculos como paredes ou grandes construções e ainda obstáculos naturais como montanhas, florestas ou colinas. Estes obstáculos diminuem muito a distância de comunicação entre um dispositivo e o *gateway* próximo e fazem com que apareçam áreas de sobra onde

os dispositivos *LoRaWAN* não consigam transmitir os dados a nenhum *gateway*. Uma solução possível seria reposicionar o *gateway* para minimizar essas áreas de sombra, porém, o *gateway* necessita de conexão com a *internet* para retransmitir os dados o que restringe a possibilidade de reposicionamento para áreas onde o *gateway* esteja dentro da cobertura de uma rede *Wi-Fi*, por exemplo.

Diante desta situação, várias possibilidades de melhoria do alcance da rede *LoRa* e *LoRaWAN* vem sendo estudadas, como por exemplo:

- a criação de uma rede *point-to-point* (ponto a ponto) de sensores que estende o alcance da rede *LoRa*, mas não faz conexão entre as redes *LoRa* e *LoRaWAN*, apresenta ainda alguns inconvenientes em que se um dispositivo intermediário falhar ou desligar, não existe outro caminho para os dados continuarem trafegando do último dispositivo até a aplicação tornando esta solução pouco eficaz (ABRARDO et al., 2019).
- outra possibilidade é um extensor *multi-hop* (multi-saltos) para a rede *LoRa* convencional que permitiria a retransmissão dos dados provenientes de sensores através de dispositivos retransmissores nas proximidades, denominados *Relay Nodes*, organizados de forma a estender a cobertura da rede *LoRaWAN* (ASLAM et al., 2020).
- outro extensor para a rede *LoRaWAN* foi discutido em (SISINNI et al., 2019) onde os autores desenvolvem um dispositivo especial que permite o reenvio dos dados para um dispositivo que esteja mais distante, aumentando assim o alcance da rede *LoRaWAN*. Este extensor apresenta total compatibilidade com o protocolo *LoRaWAN* garantindo assim a comunicação de dados com um dispositivo mais distante através do *e-Node* (*extender node* - nó extensor) permitindo assim que os dados sejam transmitidos e recebidos respeitando o protocolo *LoRaWAN*.

1.2 OBJETIVO

O presente trabalho apresenta uma possível solução: a Rede Híbrida *LoRa* em Malha / *LoRaWAN* através do desenvolvimento de um dispositivo especial denominado *Proxy* que permite a comunicação com uma infraestrutura *LoRaWAN* já existente e fará a ponte entre as redes *LoRa* em Malha e *LoRaWAN*, permitindo que em regiões de sombra para a rede *LoRaWAN*, possam ser sensoreadas e assim, permitir que se possa aumentar a área de cobertura para um ou outros *gateways* próximos sendo que, em caso de falha de um ou outro dispositivo a reorganização da rede em malha possa encontrar outra rota para enviar os dados, possibilitando um maior alcance na transmissão *LoRaWAN* e maior confiabilidade mesmo que um dispositivo venha a falhar.

O objetivo deste trabalho é então, desenvolver um dispositivo concentrador, denominado *Proxy*, totalmente compatível com a rede *LoRaWAN*, que fará a interconexão entre a Rede Híbrida *LoRa* em Malha e *LoRaWAN*, de tal forma que amplie o alcance da rede *LoRaWAN* em áreas de sombra, utilizando para isto uma topologia de malha. O *Proxy* é um dispositivo compatível com a rede *LoRaWAN* e tem duas funções: gerenciar a rede *LoRa* em Malha, permitindo a comunicação dos dados recebidos dos dispositivos *LoRa* em Malha e reencaminhando os dados para um *gateway LoRaWAN* próximo.

1.3 ESTRUTURA E CONTEÚDO

A estrutura deste trabalho está dividida em 7 capítulos, considerando esta introdução e desconsiderando as referências bibliográficas.

O capítulo 2 trata dos conceitos relacionados aos principais tópicos acerca do tema proposto com relação à rede *LoRaWAN*, bem como do protocolo *LoRa* e os protocolos de roteamento para a rede em malha.

O capítulo 3 apresenta uma revisão bibliográfica dos principais trabalhos relacionados ao tema proposto e apresenta alguns estudos que demonstram a importância do presente trabalho.

O capítulo 4 traz a proposta de trabalho que é o desenvolvimento de uma Rede Híbrida *LoRa* em Malha / *LoRaWAN* e apresenta o *hardware* utilizado em toda infraestrutura para o funcionamento dos dispositivos *LoRaWAN*, o *Proxy* e *LoRa* em malha.

O capítulo 5 mostra o Servidor de Rede e Aplicação utilizado bem como seus principais detalhes e o desenvolvimento dos elementos do trabalho, tais como: a infraestrutura do *gateway LoRaWAN*, o *Proxy*, os dispositivos *LoRa* em malha e versa sobre a proposta da integração da rede *LoRaWAN* e *LoRa* em malha. Também apresenta um roteiro do funcionamento da rede em malha desenvolvida e as características do *gateway LoRaWAN* utilizado.

O capítulo 6, por sua vez, apresenta os resultados obtidos com o desenvolvimento do *Proxy*, da rede *LoRa* em malha e como são feitas as leituras dos dados na *interface* do *ChirpStack* e *Grafana*. Apresenta também, os resultados dos testes de campo realizados em área de sombra para a cobertura do *gateway LoRaWAN* instalado no Campus da Unesp e algumas considerações sobre os testes.

Por fim, o capítulo 7 apresenta as considerações finais e as conclusões obtidas com a elaboração do trabalho, bem como, aponta possibilidades de melhorias sobre os resultados obtidos.

2 Conceitos relacionados

A seguir são apresentados os conceitos mais importantes relacionados com o presente trabalho, pois apresenta os conhecimentos necessários para o entendimento do que é proposto.

2.1 Rede em Malha

As redes em malha *Wireless Mesh Networks* (*WMNs*) são redes dinamicamente auto-organizadas e auto-configuradas onde os nós estáticos estabelecem automaticamente uma conexão *ad-hoc* (do Latin "para este"), mantendo a conectividade em malha (*mesh*). A rede *mesh* possui dois tipos de nós: roteadores *mesh* e clientes *mesh*. Os roteadores *mesh* além de funcionar como *gateway/bridge* (redirecionador/ponte) controlam a rede *mesh* através de funções de roteamento gerenciando a rede *mesh* e apresentam pouca mobilidade em relação aos clientes. Os clientes *mesh* podem atuar como roteadores, mas não como *gateways* ou *bridges* pois, apresentam um *hardware* mais simples. Através da rede *mesh* com múltiplos nós, é possível cobrir uma grande área consumindo menos energia (AKYILDIZ; WANG, 2005).

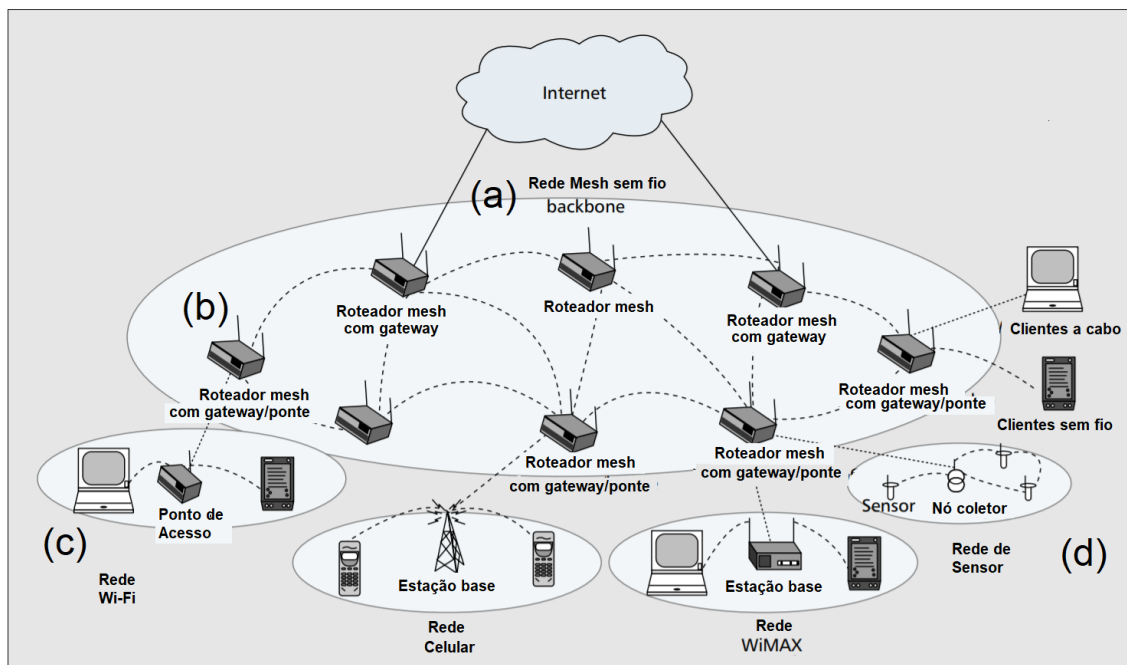
Conseqüentemente, sendo um outro tipo de rede *ad-hoc*, as *WMNs* trazem várias vantagens como baixo custo, fácil manutenção, robustez, cobertura confiável, entre outras e vem sendo amplamente utilizada em vários cenários de aplicações como redes domésticas, redes comunitárias, automação predial, redes metropolitanas de alta velocidade, entre outras. Embora as aplicações da rede *mesh* estejam crescendo alguns problemas existem, por exemplo, endereços *MAC* (*Media Access Control*) e os protocolos de roteamento não são escaláveis e a taxa de transferência diminui conforme o aumento do número de nós aumenta, portanto, os protocolos de roteamento existentes constantemente precisam de melhorias para as *WMNs* que apresentam adaptações do protocolo *Institute of Electrical and Electronic Engineers* - IEEE 802.11-(2004) e novas atualizações feitas compõem a especificação IEEE 802.11s-(2012) que possibilita a comunicação entre diversos dispositivos *mesh*.

2.1.1 Arquitetura em Malha

Para compor uma rede em malha, podemos nomear alguns componentes como pode ser visto na Figura 1, onde temos: (a) a conexão **infraestrutura/backbone** com a *internet* que forma a infraestrutura para conectar os clientes dos roteadores até a *internet*, esta infraestrutura pode utilizar várias tecnologias de rádio comunicação, não somente as

citadas em *IEEE* 802.11, utilizando as características de *gateway/bridge* dos roteadores; (b) **roteadores em malha** que podem apresentar diversas funcionalidades: *gateway*, *bridge* ou *gateway/bridge*, estes gerenciam as rotas entre os diversos clientes até a conexão com o *backbone* de *internet*; (c) os **clientes em malha** que fazem as requisições de envio e recebimento de dados para a *internet*; e (d) um exemplo de **WMNs Híbridas** que combinam a infraestrutura dos clientes e roteadores em malha, que se conectam com a *internet* e entre outros clientes da própria malha em mais sub-redes, como por exemplo *Wi-Fi*, *WiMAX* (*Worldwide Interoperability for Microwave Access*), celular e redes de sensores.

Figura 1 – Infraestrutura em malha: (a) infraestrutura/*backbone*, (b) roteadores em malha, (c) clientes em malha, (d) *WMNs* Híbridas.



Fonte: Akyildiz e Wang (2005) adaptado.

Esta arquitetura híbrida apresenta algumas características como: suporta a rede *ad-hoc* e permite auto-formação e auto-organização, rede em malha multi-saltos, pouca mobilidade para os roteadores, mobilidade dos clientes *wireless*, possibilidade de rede heterogênea, consumo de energia diferentes entre roteadores e clientes e ainda, as redes em malha não funcionam sozinhas e precisam dos *gateways* para operar com as outras redes.

2.2 LoRa

A tecnologia *LoRa* é um protocolo proprietário desenvolvido pela empresa *Semtech* que utiliza uma técnica de espalhamento de sinal, que se assemelha a um "assobio", chamada de *Chirp Spread Spectrum (CSS)*, que foi desenvolvida na década de 1940 para uso em

radar militar, proporciona envio de sinais a longas distâncias e alta imunidade a ruídos (ANGRISANI et al., 2017), e está alocada em bandas de transmissão de rádio abaixo de 1GHz.

Utilizando uma técnica alternativa de espalhamento de sinal, se comparada às técnicas mais comuns, envia o sinal como se assobiasse (*chirp*) repetidamente entre um máximo e um mínimo de forma a espalhar o sinal onde o intervalo de frequência deste envio equivale ao espectro da onda de frequência deste sinal (ANGRISANI et al., 2017) apud (SEMTECH, 2015).

Dentre as principais características do *LoRa* pode-se citar:

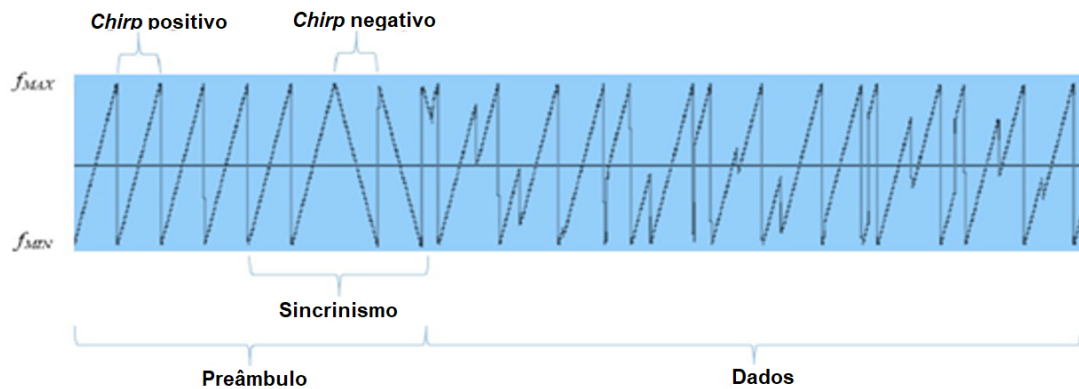
- Largura de banda (*Band Width - BW*) entre 125-250 KHz;
- Fator de Espalhamento (*Spread Factor - SF*) com número de *bits* decodificados por caractere de 7 a 12;
- Taxa de codificação (*Code Rate - CR*) que é medido com o valor de correção dos erros.

A estrutura dos dados de um pacote *LoRa* pode ser visto na Figura 2, estes pacotes podem apresentar os seguintes parâmetros:

- Preâmbulo (*Preamble*) referente ao *LoRa*;
- Mensagem de sincronismo (*Sync*) que inicia a rede *LoRa*;
- Um cabeçalho opcional (*Optional header*) de informações de comunicação;
- A informação (*Data Payload*);
- *Cyclic Redundancy Check - CRC* para checar erros.

O preâmbulo no início do pacote apresenta uma sequência de sinais (*chirp*) que variam sua frequência de alta para baixa em intervalos determinados, após temos um *byte* de valor longo (*word*) usado para identificar diferentes redes *LoRa*, este *byte* é utilizado como identificador em que se não for conhecido o nó rejeita a conexão e ainda pode ser utilizado para sincronismo que é formado pelo segundo ou o quarto sinal (*chirp*) de baixa frequência com duração de 2,25 símbolos. Após este preâmbulo, que é opcional, são adicionados os dados (*payload*) que apresenta um limite de 255 *bytes* (ANGRISANI et al., 2017).

Em termos de *LoRa*, o código de espalhamento aplicado ao dado original é chamado de Fator de Espalhamento ou *Spreading Factor (SF)*, este fator tem um total de seis

Figura 2 – Pacote *LoRa*.

Fonte: Angrisani et al. (2017).

parâmetros ($SF7$ a $SF12$), quanto maior o SF utilizado mais longe o dado poderá ser enviado sem ser recebido com erro por um receptor (SEMTECH CORPORATION, 2019).

Na Tabela 1, é mostrado a relação entre quatro diferentes SF ($SF7...SF10$) usados em envio (*uplink*) de dados em um canal de 125KHz. Esta tabela mostra a taxa de transferência de *bits* (*bit rate*) em comparação com a distância estimada de envio, mas, para este alcance ainda é necessário levar em consideração alguns fatores, tais como, o terreno, se está em área rural ou urbana. Também podemos ver o tempo de permanência no ar *Time On Air* (*TOA*), estes valores são para um *payload* de 11 *bytes* para cada SF (SEMTECH CORPORATION, 2019).

Tabela 1 – Comparação entre *LoRa SF*

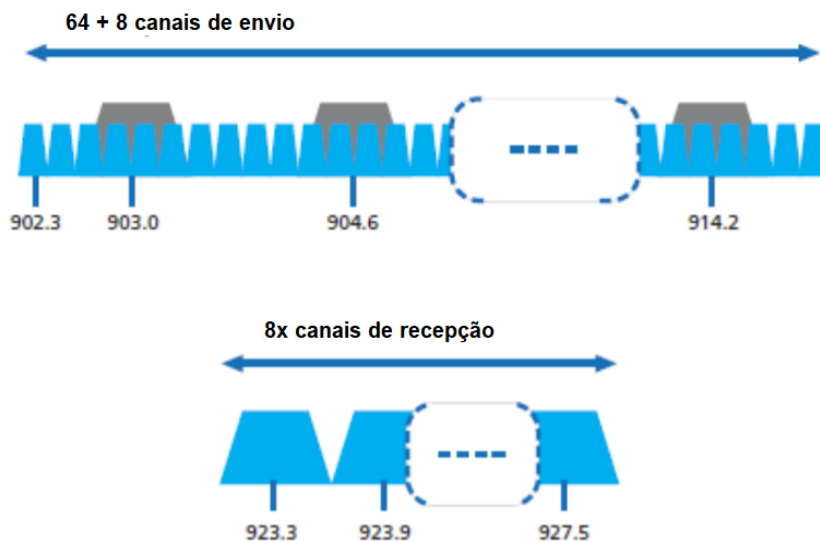
SF (para <i>uplink</i> a 125KHz)	Taxa de <i>bits</i>	Distância (depende dos obstáculos)	<i>TOA</i> para um <i>payload</i> de 11 <i>bytes</i>
$SF10$	980 <i>bps</i>	8 km	371 ms
$SF9$	1760 <i>bps</i>	6 km	185 ms
$SF8$	3125 <i>bps</i>	4 km	103 ms
$SF7$	5470 <i>bps</i>	2 km	61 ms

Fonte: SEMTECH.

Há especificações regionais sobre a utilização de cada canal de comunicação referente a cada país ou região, estas definições podem ser vistas em (LORA-ALLIANCE, 2020b), por exemplo, na América do Norte existem 64 canais distantes 200KHz entre eles e com um tamanho de 125KHz cada, ainda temos mais 8 canais para *uplink* e 8 para *downlink*, ambos com 500KHz cada, a Figura 3, mostra esta distribuição de canais.

No Brasil a banda de frequência utilizada pela rede *LoRa* (*AU915*) é a faixa de frequências não licenciadas *ISM* (*Industrial Scientific Medical*) na faixa de 902MHz a 928MHz (ANATEL, 2014) que compreende uma das faixas de comunicação do *LoRa*. Estas faixas *ISM* são limitadas pela potência máxima de 4W e 36dBm e são bandas internacionalmente reservadas para o desenvolvimento industrial, científico e médico.

Figura 3 – Distribuição de canais para a América do Norte.



Fonte: Semtech Corporation (2019).

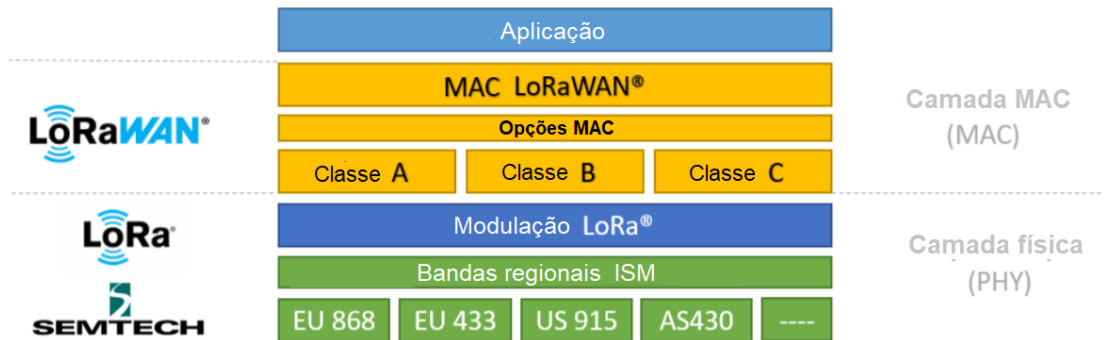
Não existe até o momento uma especificação regional *LoRaWAN* para o Brasil na especificação *LoRa-Alliance*, por tanto, a faixa *LoRa* utilizada é a da Austrália (*AU915*).

2.3 *LoRaWAN*

Para o protocolo *LoRaWAN*, na sequência, observe quais são os principais pontos verificados sobre esta rede e faz se um aprofundamento maior nestes tópicos.

2.3.1 Características do protocolo

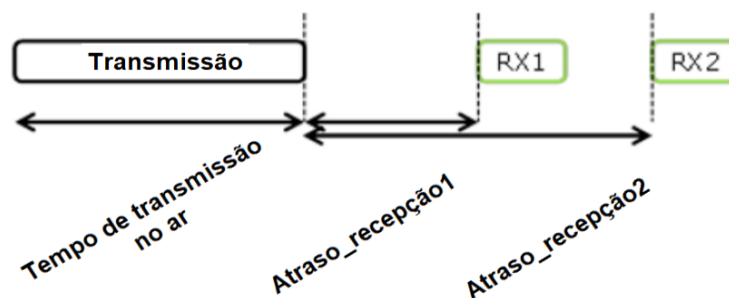
Para um amplo entendimento sobre *LoRaWAN*, vamos entender as diferenças entre *LoRa* e *LoRaWAN*, como podemos ver na Figura 4, o *LoRa* implementa a camada física que é responsável por criar o *link* de longa distância, já o *LoRaWAN* é um protocolo de rede aberto que permite uma comunicação bidirecional, mobilidade e padrões de serviços mantidos pela *LoRa Alliance* (LORA-ALLIANCE, 2020a).

Figura 4 – Tecnologia da pilha *LoRaWAN*

Fonte: Semtech Corporation (2019).

A camada física (*Physical Layer - PHY*) é responsável por gerenciar o *link* e seus parâmetros, tais como: canais, frequência regional e bandas, entre outros. Já na camada *LoRaWAN* (*Media Access Control - MAC Layer*) temos toda a pilha do protocolo *LoRaWAN* e ainda a definição de três classes de dispositivos:

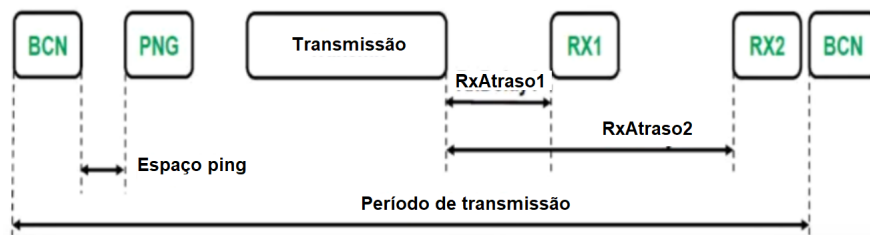
- Classe A (*All*) que permite transmissões sem a verificação de um *gateway* próximo e representa dispositivos com um controle eficiente de consumo de energia. Nos dispositivos Classe A são configuradas janelas curtas de recepção *downlink RX1* e *RX2* que tem um tempo variável a depender do preâmbulo da mensagem. A recepção de uma mensagem *downlink* pode ocorrer em qualquer janela mas não em ambas e, a janela *RX1* inicia em $1s \pm 20\mu s$ e se a mensagem não for destinada ao dispositivo receptor após $2s \pm 20\mu s$ a janela *RX2* é aberta. Ambas as janelas diferem com relação à frequência e o parâmetro *data rate*, dependendo da configuração regional, e ficam disponíveis somente após um envio de dados *uplink*, não sendo possível uma comunicação *downlink* sem que se tenha uma mensagem recebida. Na Figura 5, pode ser visto a distribuição destas janelas de recepção;

Figura 5 – Janelas de recepção *LoRaWAN*.

Fonte: LoRa-Alliance® (2020a).

- Classe B (*Beacon*), que são dispositivos geralmente alimentados por bateria, permitem mais eventos de *downlink* sincronizados em tempos fixos com um *gateway* através de sinalizadores: *beacons*, esta configuração é mais utilizada para atuadores e tem a característica de ter várias janelas de recepção em relação aos dispositivos de Classe A. Os *beacons*, então, são sinais periódicos utilizados para manter o dispositivo sincronizado com o *gateway* o que permite que o dispositivo receba um comando em um tempo conhecido. Na Figura 6 pode ser visto a temporização do evento *beacon* de um dispositivo Classe B;

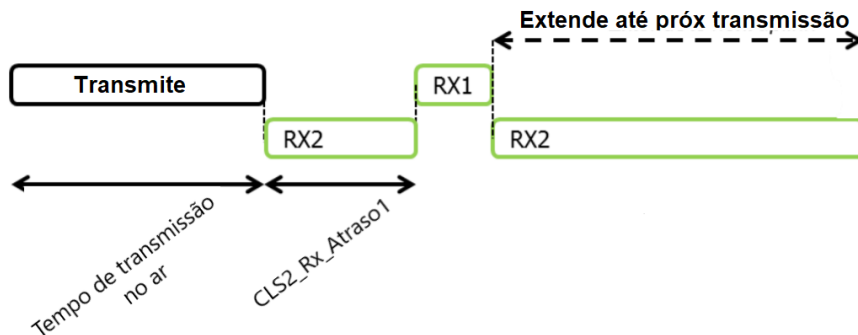
Figura 6 – Janelas de recepção *LoRaWAN* Classe B.



Fonte: LoRa-Alliance® (2020a).

- Classe C (*Continuous*), em que a alimentação deve ser externa e representa a configuração de maior consumo energético, pois o dispositivo deve ficar continuamente em modo de escuta (recepção), implementa as mesmas configurações de janelas de recepção dos dispositivos Classe A, porém, o tempo da janela *RX2* se estende até próxima transmissão, como pode ser visto na Figura 7, o que permite o *downlink* a qualquer instante, mas, esta opção deve ser configurada e não é padrão.

Figura 7 – Janelas de recepção *LoRaWAN* Classe C.



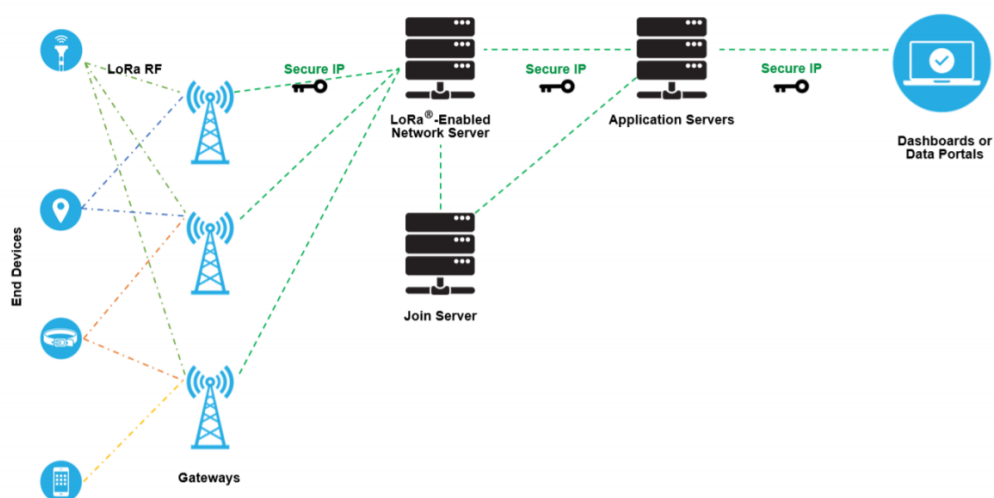
Fonte: LoRa-Alliance® (2020a).

A Rede *LoRaWAN* implementa uma topologia de estrela-de-estrelas onde temos uma conexão em estrela para cada *gateway* conectado à rede, um exemplo geral da

arquitetura de uma rede *LoRaWAN* (Semtech Corporation (2019)) pode ser vista na Figura 8, onde observa-se que a rede apresenta diversos elementos como se segue:

- *End device*: são elementos onde temos sensores ou atuadores que se conectam à rede por meio dos *gateways* através da modulação *RF LoRa*;
- *Gateway*: estes recebem os dados modulados no *link LoRa* e enviam estas mensagens para um Servidor de rede *LoRaWAN* através de um *link* seguro com a *internet*, esta conexão com a *internet* pode ser feita de diversas formas, como *Wi-Fi*, via cabo *ethernet* ou conexão celular;
- *LoRaWAN Network Server (NS)*: é responsável por gerenciar toda a rede e controlar os parâmetros desta estabelecendo conexões seguras que irão transportar os dados entre o *End Device* até a aplicação final;
- *Join Server*: a função deste servidor é controlar as chaves de acesso para cada *End Device*, mais detalhes veremos mais a frente;
- *Application Server (AS)*: este servidor fica responsável por gerenciar os dados provenientes sensores dos *End Devices* e controlar as informações de retorno aos atuadores conectados a estes, tais dados são direcionados às aplicações dos usuários;
- *Dashboards* ou *Data Portals*: representa a aplicação final do usuário onde temos a apresentação dos dados captados pelos sensores e pode enviar comandos de controle para os atuadores conectados à rede *LoRaWAN*.

Figura 8 – Topologia da rede *LoRaWAN*.

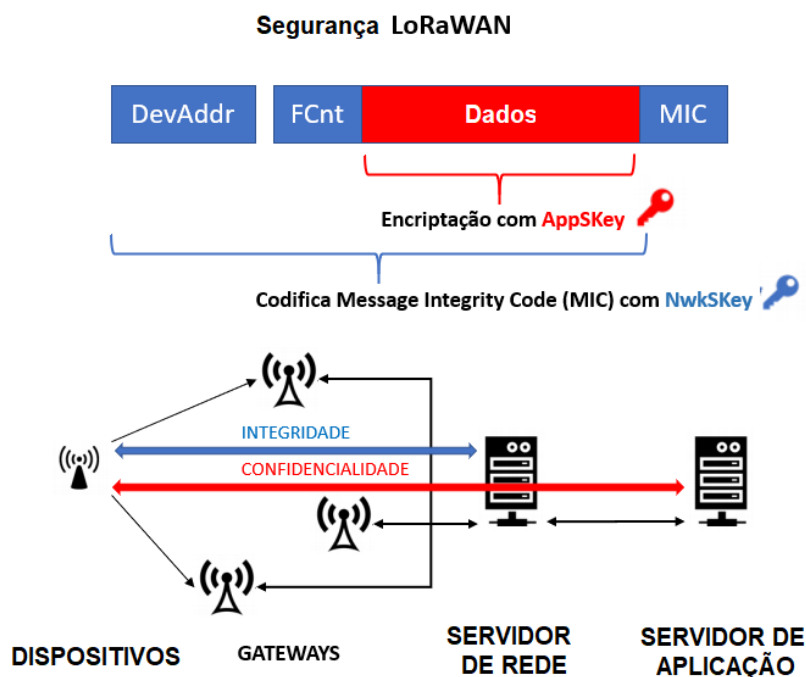


Fonte: Semtech Corporation (2019).

2.3.2 Segurança

Como o protocolo *LoRaWAN* é otimizado para baixo consumo de energia e baixo custo, a segurança de seus *links* de dados também é otimizada para baixo consumo de energia, baixa complexidade, baixo custo e alta escalabilidade (GEMALTO; ACTILITY; SEMTECH, 2017). Na Figura 9, pode se ver como as chaves de encriptação encaminham os dados através de mútua autenticação, proteção de integridade e confidencialidade dos dados na rede *LoRaWAN*.

Figura 9 – Segurança da rede *LoRaWAN*.



Fonte: Autor.

Cada dispositivo deve receber um conjunto de chaves que o tornam único, estas chaves são globalmente registradas pelo *IEEE Registration Authority* (Autoridade de Registro *IEEE*) que é um órgão administrativo responsável pelo gerenciamento destas chaves que são:

- *DevAddr*: *Device Address* é o endereço do dispositivo na rede (não único);
- *DevEUI*: *Device End-device Unique Identifier* que é o identificador global único do dispositivo;
- *AppKey*: *Application Key* que identifica a aplicação e é utilizada para gerar a chave de sessão da rede;
- *NwkSKey*: *Network Session Key* que é a chave de sessão da rede;

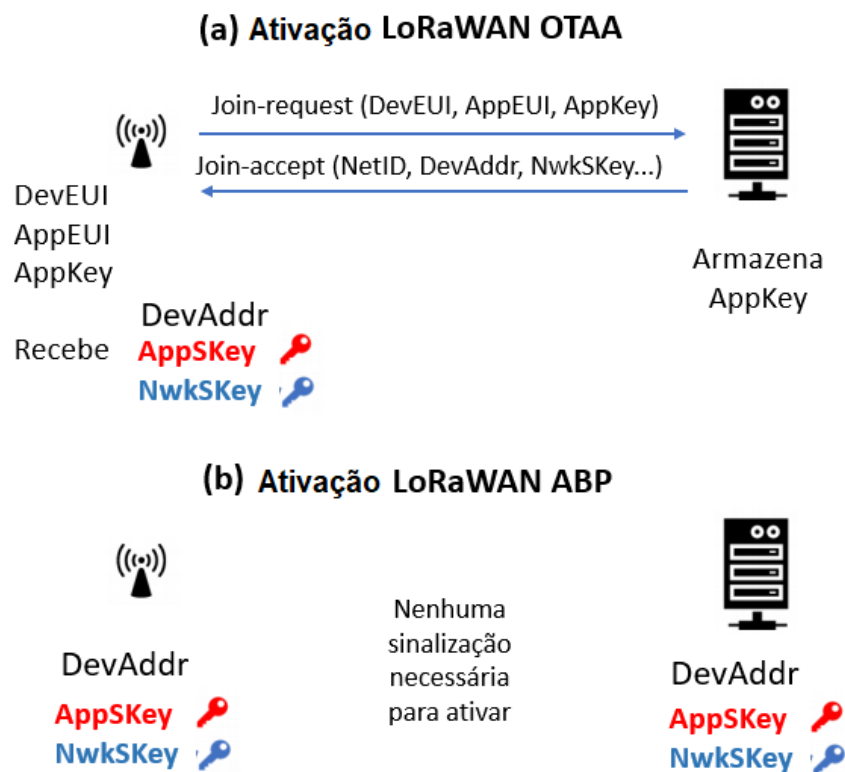
- *AppSKey*: *Application Session Key* que identifica a sessão da aplicação na rede.

Estes mecanismos de autenticação e encriptação mencionados utilizam a criptografia *AES* (*Advanced Encryption Standard*), que é um algoritmo público baseado em chaves simétricas secretas, combinando o *AES* ainda com outros modos de operação. Cada *end device* é personalizado com uma única chave *AES* de 128 bits (*AES-128*), chamada *AppKey* e, um identificador global único *DevEUI*, ambos utilizados no processo de autenticação (GEMALTO; ACTILITY; SEMTECH, 2017).

2.3.3 Métodos de ativação

Para cada dispositivo pode se conectar à rede é preciso se ativar na rede, neste processo de ativação são trocados entre o servidor de rede e o dispositivo as chaves de sessão de rede, para isto podem ser utilizados um dos métodos de ativação: **OTAA** (*Over-The-Air Activation*) ou **ABP** (*Activation By Personalization*) (LORA-ALLIANCE, 2020a). Cada um destes métodos possui uma característica bem definida. Na Figura 10, podemos ver de forma geral estes dois métodos.

Figura 10 – Ativação da rede *LoRaWAN*: (a) *OTAA* e (b) *ABP*.



Fonte: Autor.

2.3.3.1 Ativação OTAA

No modo de ativação *OTAA* há uma troca de mensagens "*Join Request*" - requisição de entrada na rede e "*Join Accept*" - pedido aceito, entre o dispositivo e o *Network Server - NS* para que ocorra o processo de personalização que é a identificação do dispositivo na rede. Neste processo o dispositivo envia um "*Join Request*" ao *Network Server* com as chaves *DevEUI*, *AppEUI* e *AppKey*, que armazena a chave *AppKey* e gera as chaves *AppSKey* e *NwkSKey* além do *NetID* que é o identificador de rede. O *Network Server* retorna o sinal '*Join Accept*' com os dados recebidos e *AppSKey*, *NwkSKey*, *NetID* (identificador de rede), *RxDelay* (indica a temporização da janela de recebimento) e *CFList* (uma lista opcional de canais de frequência).

Após este processo, o dispositivo já é reconhecido na rede e pode enviar os dados para sua aplicação final, sendo que o *Network Server* possui todas as chaves para decodificar os dados e encaminhar para a aplicação configurada.

2.3.3.2 Ativação ABP

No modo de ativação *ABP* ou Ativação por Personalização em cada dispositivo são programadas as chaves *DevEUI*, *DevAddr*, *AppKey*, *NwkSKey* e *AppSKey* que devem ser idênticas às chaves programadas no *Network Server*, estas chaves contém todos os dados para que o dispositivo seja ativado e personalizado na rede sendo um método de ativação pouco seguro, visto que, os dados são enviados pelo dispositivo até a aplicação com apenas um nível de encriptação.

3 Revisão bibliográfica

No recente campo de *IoT* as *LPWANs* vem ganhando espaço pois, permitem várias vantagens, tais como, longas distâncias de comunicação e uma boa relação de consumo de energia por tempo de uso quando alimentados por bateria quando utilizados no sensoriamento e controle remotos nas mais diversas aplicações. Porém, mesmo possibilitando a comunicação em longas distâncias, algumas características do protocolo *LoRa* faz com que apareçam algumas desvantagens, são elas: topologia estrela que permite comunicação com apenas um salto, perdas de sinal que ocorrem devido às construções ou a topografia do local onde se instalam os dispositivos e baixa taxa de transmissão da rede. Estes pontos devem ser observados quando da instalação de um *gateway LoRaWAN* pois, além destes fatores abordados este deve ser instalado em um local onde deve existir uma conexão com a *internet*. Estes fatores somados minimizam o alcance da rede *LoRa/LoRaWAN* ou fazem com que apareçam áreas de sombra para os dispositivos da rede. Neste capítulo apresentam-se algumas soluções no sentido de minimizar estes problemas onde vários autores demonstram suas soluções no sentido de mitigar as áreas de sombra e estender o alcance e a escalabilidade da rede *LoRaWAN*.

Lundell et al. (2018) apresenta uma combinação do *Hybrid Wireless Mesh Protocol (HWMP)* e o protocolo *Ad-hoc On-Demand Distance Vector (AODV)* a fim de criar uma rede leve para o *LoRa*. Após analisar vários protocolos proativos e reativos de roteamento o autor aponta esta combinação para criar um *nano-gateway* (que não implementa o protocolo *LoRaWAN*) e mecanismos de roteamento transparentes para os dispositivos e o Servidor de Rede e aponta que a solução trabalha nos tempos de atraso das transmissões. Esta solução procura desenvolver uma rede *LoRa* em malha para melhorar a viabilidade e escalabilidade da rede *LoRa/LoRaWAN*, porém, aponta que seu protocolo desenvolvido deve ser implementado no *gateway*, modificando o sistema dos *gateways* já existentes. O autor comenta ainda que seu protocolo está vulnerável a ataques demonstrando que necessita de testes neste sentido e aponta necessidade de futuros trabalhos com relação a rotas falsas (LUNDELL et al., 2018).

Azhari et al. (2019) aponta como uma possível solução para estender o alcance da rede *LoRa* um protocolo multi-saltos entre dispositivos *LoRa* em uma *Wireless Sensor Network (WSN)* e comenta que as redes multi-saltos podem ser utilizadas entre os dispositivos *LoRa* ou entre os *gateways*. Alerta que nos dispositivos *LoRa* esta função aumentaria o consumo de energia. Sua principal ideia é implementar funções de roteamento nos *gateways* permitindo que estes se comuniquem e demonstra o funcionamento em testes de laboratório seu algoritmo de transmissão de dados com multi-saltos de forma efetiva entre os dispositivos *LoRa*. Seus resultados mostram um método muito dependente da

infraestrutura proposta, aplicável às *WSNs* mas não aplicável para automação pois seu método não possui mecanismos que certifiquem o recebimento das mensagens transmitidas (AZHARI; FIRMANSYAH; BEJO, 2019).

Dias e Grilo (2020) apresentam uma proposta de um protocolo de roteamento multi-saltos para estender o alcance das redes *LoRaWAN*, compatível com o protocolo de rede *LoRaWAN* mas apenas para a transmissão de dados (*uplink*) e mostra o resultado de alguns testes com um salto, dois, três e quatro saltos onde demonstra que sua proposta funciona corretamente dentro das especificações para um salto, já com mais saltos a taxa de perda de pacotes aumenta. Os autores propõem uma topologia linear, utilizando o protocolo *DSDV* (*Destination Sequenced Distance Vector*) simplificado e adaptado para a proposta, onde implementa um dispositivo chamado *Relay Node* (*RN*) que retransmite os dados para um dispositivo próximo em direção ao *gateway* e um dispositivo *Leaf Node* (*LN*) o qual está no final da rede. Através dos pacotes de controle do protocolo *LoRaWAN*, os autores implementam os algoritmos de controle para gerenciar os pacotes entre pacotes de controle de rotas e o de dados, sendo que um dispositivo *RN* ao receber um pacote ele pode determinar se é um pacote de rota, criando as rotas ou um pacote de dados que deve ser retransmitido, um mecanismo de fila, chamado de "mail", é utilizado para gerenciar as janelas de transmissão. No artigo, os autores não informam quantos dispositivos podem ser conectados à rede, ficando este estudo para futuros trabalhos, mas com quatro saltos já observam uma taxa de 90% de pacotes recebidos pelo Servidor de Rede e indicam que a proposta se torna válida em cenários, onde esta taxa de entrega de pacotes possa ser aceita (DIAS; GRILO, 2020).

Duong e Kim (2018) apresentam uma solução para estender a cobertura da rede *LoRaWAN* em uma proposta linear e multi-saltos baseado na sincronização de tempos entre os dispositivos de sua rede onde cada dispositivo faz a transmissão em uma janela de tempo. Desta forma, para o sensoriamento de uma linha de transmissão de energia, por exemplo, vários dispositivos podem ser ligados e enviam os dados sincronizando-se os tempos. O protocolo proposto trabalha em dois períodos o período de inicialização e o período de transmissão. No período de inicialização, o primeiro dispositivo inicia o processo enviando sinais de controle para o próximo até o último dispositivo da fila e este retorna a resposta, informando aos anteriores as janelas de transmissão. Ao final do processo de inicialização da rede todos os dispositivos mantêm configurado sua janela de transmissão e a rede pode começar a transmitir os dados. No período de transmissão, os dispositivos podem enviar os dados ao próximo e as transmissões acontecem em cadeia na sua janela de tempo, sendo que o dado de um determinado dispositivo pode ser transmitido e ao receber um dado de um dispositivo anterior este dado deve ser retransmitido até o primeiro da fila. Os resultados apresentados demonstram que o sistema operou corretamente, porém, se o número de dispositivos aumenta o tempo que um dispositivo leva para entregar seus dados também aumenta diminuindo a eficiência da rede (DUONG; KIM, 2018).

Abrardo et al. (2019) apresenta uma solução usando a tecnologia *LoRa* para fazer o monitoramento de fluxo de pó preto em dutos de gás implementando um sistema de multi-saltos aproveitando as características do *LoRa* de poder se comunicar a longas distâncias, pois explica que em sistemas de dutos de poucos metros, como em refinarias, por exemplo, outros tipos de redes podem ser empregadas como: *Wi-Fi*, *Bluetooth* ou *ZigBee*, porém, quando os dutos de gás somam muitos quilômetros de distância o monitoramento é um desafio grande pois os dutos podem passar por áreas onde nem a rede celular pode estar disponível e ainda, em lugares muito distantes até a alimentação elétrica não é possível podendo somente um equipamento de monitoramento funcionar por baterias o que o *LoRa* se traduz em solução, pois permite uma longa vida de baterias e a comunicação a longas distâncias (ABRARDO et al., 2019).

Para esta rede, os autores propõem uma rede Linear *LoRa* onde os dispositivos *LoRa* só podem se comunicar com dois outros, um anterior e um posterior e implementa os controles para este fim com dois tipos de dispositivos *LoRa*: um dispositivo *Gateway*, que está no início da rede e está ligado ao sistema de controle e, um dispositivo Sensor.

O dispositivo Sensor é capaz de fazer medições, receber e transmitir dados sendo que o dispositivo que receber dados de outro nó deve retransmitir este dado para frente até chegar ao dispositivo *Gateway*. Para esta rede Multi-saltos foi proposto um protocolo clássico de sincronização por *MAC* e o tempo foi dividido em períodos *SYNC*, *DATA* e *SLEEP*, onde os dispositivos informam em qual momento vão fazer uma transmissão e depois voltam a dormir (*SLEEP*) e ainda controles que evitam a colisão de transmissões.

Esta rede multi-saltos permite o monitoramento de grandes distâncias em dutos de gás, formando uma rede linear, aproveitando as características do protocolo *LoRa* onde não se tem a cobertura de outras redes de dados, com baixo consumo de energia e ainda mostra que as redes entre dispositivos *LoRa* são uma excelente opção para locais onde não existe a cobertura de outras redes e se necessita de uma comunicação de dados eficiente. Os testes realizados foram em laboratório e aponta que testes em campo serão realizados em trabalhos futuros para verificar possíveis melhorias no protocolo proposto.

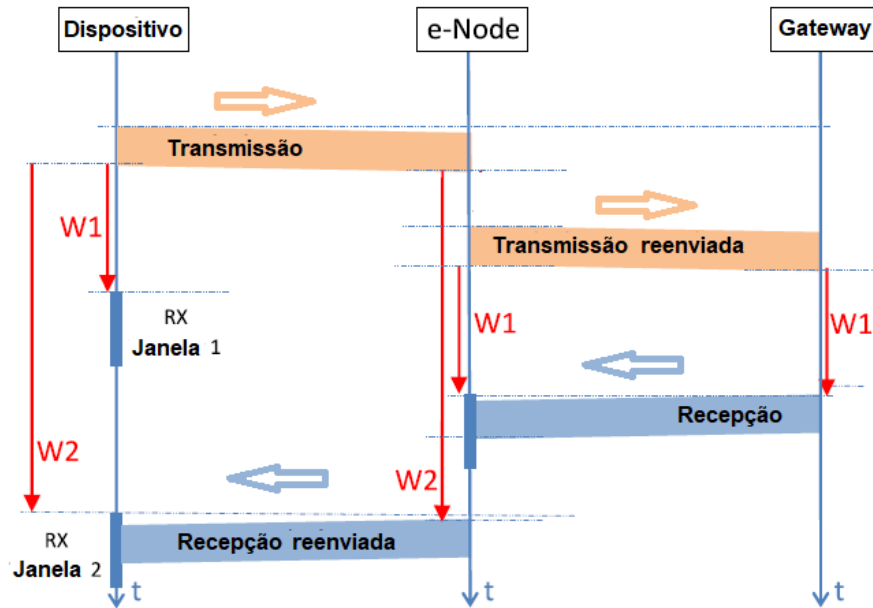
Sisini et al. (2020), comenta que, as redes *LoRaWAN* apresentam uma efetiva solução para o cenário de uso real industrial quando se utiliza transmissões de poucos dados por nós, entretanto, as diversas aplicações necessitam de longas áreas de cobertura tanto em áreas abertas ou fechadas com a presença de obstáculos fixos ou móveis. Ainda comenta que para resolver essas situações utilizando a rede *LoRaWAN* uma solução seria aumentar o número de *gateways* resultando em uma infraestrutura mais complexa (ligações cabeadas de rede entre os *gateways*); outra solução seria aumentar a sensibilidade de recepção dos *gateways*, mas isso penaliza as taxas de transmissão. Essas soluções podem não ser efetivas ou não se sustentam no ambiente industrial necessitando assim de alternativas preservando a segurança da rede *LoRaWAN* como, por exemplo, duas soluções comerciais existentes, o

”ISN-extender” e ”LoRa RF Repeater”. Estas soluções, funcionam como repetidores de sinal que, implementam uma solução proprietária sobre o rádio *LoRa* e o autor conclui que estas soluções ampliam a complexidade dos dispositivos (SISINNI et al., 2019).

Neste sentido, *Sisini et al.* (2020), apresenta um dispositivo transparente denominado *e-Node (Extender Node)* que pode operar nas mais altas taxas de transferência da rede *LoRaWAN* preservando as temporizações do protocolo. Utilizando os mesmos métodos presentes no pacote de dados *LoRaWAN*, os dados são reencaminhados utilizando senhas de administrador que permitem que os dados sejam recuperados e reencaminhados e, ainda facilitam os processos de ativação, seja por *ABP*, menos seguro ou por *OTAA*, em que o dispositivo *e-Node* gerencia as trocas de chaves com o Servidor de Rede *LoRaWAN*, onde o pacote é codificado com a chave *AppSKey*, esta chave é utilizada pelo *e-Node* quando necessário em caso de pacotes diferentes como os ”*join-accept*”, por exemplo, e em casos de mensagens enviadas para o dispositivo (*downlink*).

Como o *e-Node* deve funcionar de forma transparente na rede *LoRaWAN*, aparecem algumas limitações, pois o *e-Node* deve ser alimentado por fonte externa e não por baterias, o tempo das transmissões em relação ao tempo entre as transmissões (*duty cycle*) obedece diferentes regulamentações em determinadas regiões, mas as transmissões (*uplink*) são mais flexíveis se comparadas ao rigor da temporização das mensagens recebidas (*downlinks*). Na Figura 11 pode-se ver um diagrama de funcionamento do proposto *e-Node*, configurado para ativação *ABP* e, vê-se ainda que este funciona somente para um salto (*hop*) o que permite a extensão da área de cobertura da rede *LoRaWAN* mas, permite apenas um dispositivo apenas a ser conectado. Nesta figura os parâmetros são: ”W1” tempo até próximo evento, ”W2” tempo até o próximo evento, ”RX Janela 1” janela de tempo de recepção 1, ”RX Janela 2” janela de tempo de recepção 2, ”Dispositivo” é o dispositivo *LoRaWAN* que vai transmitir para o *gateway*, ”*e-Node*” é o *Node Extender* e ”*Gateway*” é um *gateway LoRaWAN* operando nas proximidades.

O *hardware* utilizado nos testes do *e-Node* é baseado no rádio *LoRa SX1301-SX1257* da *Semtech*, que são rádios normalmente utilizados em *gateways LoRaWAN* pois, permitem recepção simultânea nos vários *SFs (Spread Factor - fator de espalhamento)* e em diferentes canais, estes módulos foram montados em *SBCs (Single Board Computers) Raspberry pi*, com sistema operacional *Linux*. As temporizações foram captadas, utilizando *SBC IoT2040*, da *Siemens*, com sistema operacional *Linux* e rodando um serviço *Node-RED*, baseado em eventos, para registrar as temporizações dos eventos de transmissão e recepção.

Figura 11 – Diagrama de funcionamento do *e-Node* quando em ativação *ABP*.

Fonte: Autor.

Os testes de cobertura realizados em ambiente industrial na Universidade de Brescia utilizando um pacote de dados de 90 bytes em diferentes *SFs* (*SF7*, *SF8* e *SF9*) e diferentes distâncias, demonstrando que o *e-Node* possibilitou um ganho na transmissão destes sinais, especialmente com as transmissões em *SF7* que tem uma taxa de transmissão mais rápida. Os resultados experimentais, após duas semanas com vários parâmetros demonstram que o *e-Node* funcionou corretamente, inserindo poucos atrasos na temporização das transmissões, mas se mostrou eficaz para a solução proposta. O dispositivo *e-Node* apresentou um bom resultado como regenerador de sinais de transmissão para dispositivos muito distantes. Os autores comentam sobre os custos do aumento do número de gateways ainda é muito alto, mas faz uma ressalva sobre os custos de *hardware* para implementar o *e-Node* que necessita de *SBCs* muito rápidos, pois nos testes com as transmissões em *SF9* excedem o tempo permitido nas especificações *LoRaWAN* e encerra informando que necessita de novas investigações para implementar novas funcionalidades ao *e-Node* sendo este o primeiro *hardware* a estender a transmissão *LoRaWAN*.

Com base nestes estudos apresenta-se uma Rede Híbrida *LoRa* em malha / *LoRaWAN* controlada por um dispositivo *Proxy*, compatível com a rede *LoRaWAN*, que apresenta uma topologia em malha, não linear, permitindo assim que a rede *LoRaWAN* possa mitigar os pontos de sombra em sua cobertura, estender o alcance da rede e desta forma agregar uma nova solução à questão sobre a qual a rede *LoRaWAN* só conecte os dispositivos em um único salto. Sendo o dispositivo *Proxy* totalmente compatível com a

rede *LoRaWAN* não se torna necessária nenhuma alteração nas redes já existentes e a rede em malha permite que mesmo que um dispositivo venha a falhar, através da reconfiguração das rotas, os dispositivos podem encontrar novas rotas e continuar operantes, na próxima seção será apresentado em detalhes esta proposta.

4 Rede Híbrida *LoRa* em malha / *LoRaWAN*

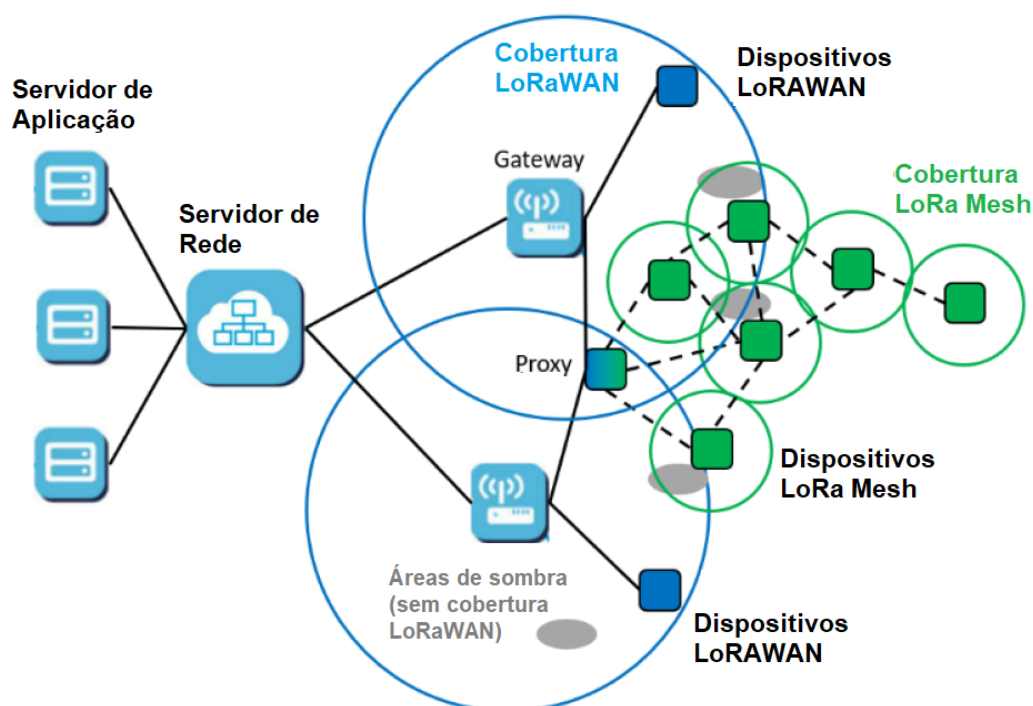
Neste capítulo são aprestadas as propostas para o presente desenvolvimento e os materiais utilizados, bem como, os requisitos de *software* utilizados no *gateway* e a solução em Servidor de Rede e de Aplicação.

4.1 Proposta do Trabalho

A proposta do presente trabalho é desenvolver uma Rede Híbrida com um elemento principal denominado *Proxy*, compatível com o protocolo *LoRaWAN*, onde este permite fazer a ponte entre uma rede *LoRa* em Malha e uma rede *LoRaWAN* já existente, desta forma, os dispositivos interconectados em malha possam se comunicar com a rede *LoRaWAN*, mesmo em uma região de sombra para a rede *LoRaWAN* pode ser possível o sensoriamento e ainda permitir a extensão da rede *LoRaWAN* a regiões onde os dispositivos não se encontram em uma área de cobertura, permitindo aumentar a área de cobertura da rede *LoRaWAN*.

Na Figura 12, pode-se observar que embora vários *gateways* possam cobrir grandes áreas, estas podem apresentar sombras de cobertura *LoRaWAN*. Estas sombras podem formar-se devido a questões como relevo ou grandes obstáculos que impossibilitam a comunicação entre os dispositivos e os *gateways* próximos, sendo recomendado que não se tenha obstáculos entre as antenas dos dispositivos e *gateways*.

O dispositivo *Proxy* tem duas funções: encaminhar os pacotes para a rede *LoRaWAN* e gerenciar a rede em malha. Neste dispositivo são utilizadas duas tabelas a de dispositivos próximos e a de rotas em que são armazenadas as rotas para todos os dispositivos em malha. Para o gerenciamento de rotas foram implementadas algumas características e propriedades do protocolo de rotas *AODV* (*Ad-hoc On-demand Distance Vector*), este protocolo de roteamento foi escolhido pois é o mais popular entre os diversos protocolos para redes em malha, por apresentar melhores resultados no gerenciamento de redes em malha e menor uso do microcontrolador, sendo mais leve em questão de processamento e, ainda outras características, tais como: um protocolo reativo em que as rotas são ajustadas sob demanda e somente as rotas ativas são mantidas, reduzindo a sobrecarga de roteamento, porém acrescenta uma lentidão inicial por conta da criação das rotas (ZHANG; LUO; HU, 2006).

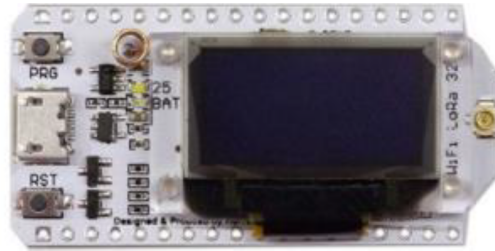
Figura 12 – Apresentação da rede *LoRa Mesh / LoRAWAN*.

Fonte: Autor.

4.2 Materiais

4.2.1 Dispositivo *LoRa* em Malha

Para o desenvolvimento dos dispositivos da malha foi utilizado a placa de baixo custo *Heltec ESP32 Wi-Fi LoRa 32 V2*, esta placa possui as seguintes características: um microcontrolador *SoC Tensilica LX6 dual core* de 32bits que opera a 240MHz, apresenta 520KB de memória *RAM*, 8MB de memória para programa e conectividade *Wi-Fi* e *BLE* (*Bluetooth Low Energy - Bluetooth* de baixa energia), apresenta ainda um *display OLED* (*Organic Light Emitter Diode*) de 0,96" para visualização de mensagens e traz um módulo rádio *SX1276* da empresa *Semtech* que é o transceptor *LoRa* com conector para antena externa. Neste modelo o rádio *SX1276* permite o ajuste para as bandas *LoRa EU868* e *US902*, fazendo com que se possa operar tanto na faixa europeia *EU868* quanto na faixa americana *US902*. Na faixa americana *US902*, que abrange as frequências de 902MHz até 928MHz, temos a faixa de frequências licenciada no Brasil que vai de 915MHz a 928MHz, esta faixa representa a faixa australiana *AU915*, sendo esta por sua vez, utilizada no Brasil que não possui uma faixa exclusiva na especificação *LoRaWAN* até o momento deste trabalho. Na Figura 13, apresenta-se o módulo utilizado para a função dos dispositivos da malha.

Figura 13 – Módulo *Heltec ESP32 Wi-Fi LoRa 32*.

Fonte: Autor.

4.2.2 Módulo transceptor *LoRa SX1276*

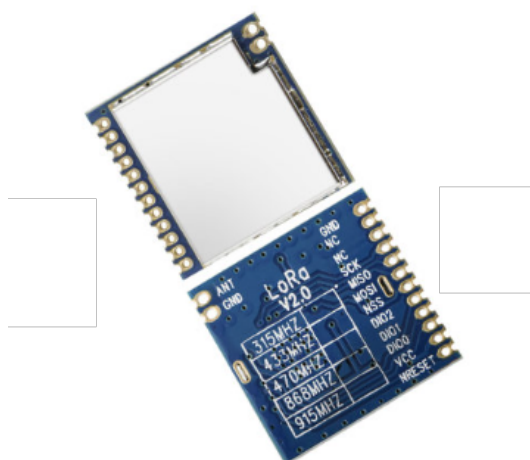
Módulo *NiceRF LoRa1276* (G-NICERF®, 2022), comercializado pela empresa *G-NiceRF*, sigla esta que nomeia os módulos *NiceRF* onde tem-se o rádio *LoRa SX1276*, produzido pela empresa *Semtech*®, algumas aplicações são indicadas, tais como: controle industrial, automação doméstica, rede de sensores, periféricos sem fio entre outras, na Figura 14, apresenta-se este módulo e suas principais características são:

- Banda de frequência: 915MHz;
- Sensibilidade *LoRa*: -139dBm;
- Potência máxima de saída: 20dBm;
- Taxa de transferência de dados: 1.2-300Kbps @FSK, 0.018-37.5Kbps @LoRa;
- Modos de modulação *LoRa*, FSK, GFSK;
- 256 bytes FIFO&CRC;
- Comunicação SPI.

4.2.3 Módulo *Espressif ESP32-DevKitC*

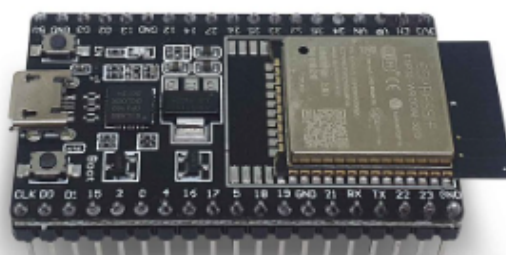
A placa de desenvolvimento *ESP32-DevKitC*, é um módulo para realização de protótipos que utilizam o microcontrolador *ESP32*, este módulo é comercializado pela empresa *Espressif* (ESPRESSIF SYSTEMS, 2022) e, é apresentado na Figura 15, o qual possui as seguintes características:

- Módulo microcontrolador *ESP32 Extensa LX6 dual core*;
- Porta micro USB para alimentação e gravação;
- Pinos 5V para alimentação externa;

Figura 14 – Módulo *NideRF SX1276*.

Fonte: Autor.

- Pinos 3V3 para alimentar periféricos.

Figura 15 – Módulo *Espressif DevKitC*

Fonte: Autor.

As características do microcontrolador *ESP32*, neste módulo, são idênticas às especificações do módulo *Heltec ESP32 Wi-Fi LoRa 32 V2*, supracitado.

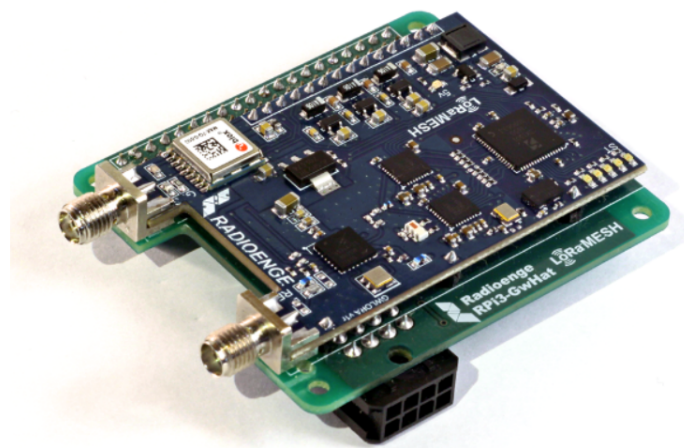
4.3 Infraestrutura *LoRaWAN*

Para a infraestrutura *LoRaWAN* foi utilizado o *Gateway LoRaWAN RD43HAT / RD43HATGPS* desenvolvido e comercializado no Brasil pela empresa Radioenge (RADIOENGE, 2020), facilmente encontrada no mercado e se mostra uma solução local e de baixo custo se comparado aos *gateways* multicanais profissionais, funciona conectado a

uma placa *Raspberry pi* com sistema operacional *Linux* tendo seu *firmware* disponibilizado na página da Radioenge, na Figura 16 pode ser visto o módulo para a montagem deste *gateway* e, abaixo estão algumas de suas características:

- compatível com protocolo *LoRaWAN*;
- 8 canais de recepção e 1 para transmissão;
- largura dos canais: 125kHz / 250KHz / 500KHz;
- potência de transmissão: 27dBm;
- sensibilidade de recepção: -137dBm;
- produto Certificado Anatel.

Figura 16 – *Gateway LoRaWAN RD43HAT / RD43HATGPS*.



Fonte: Radioenge (2020).

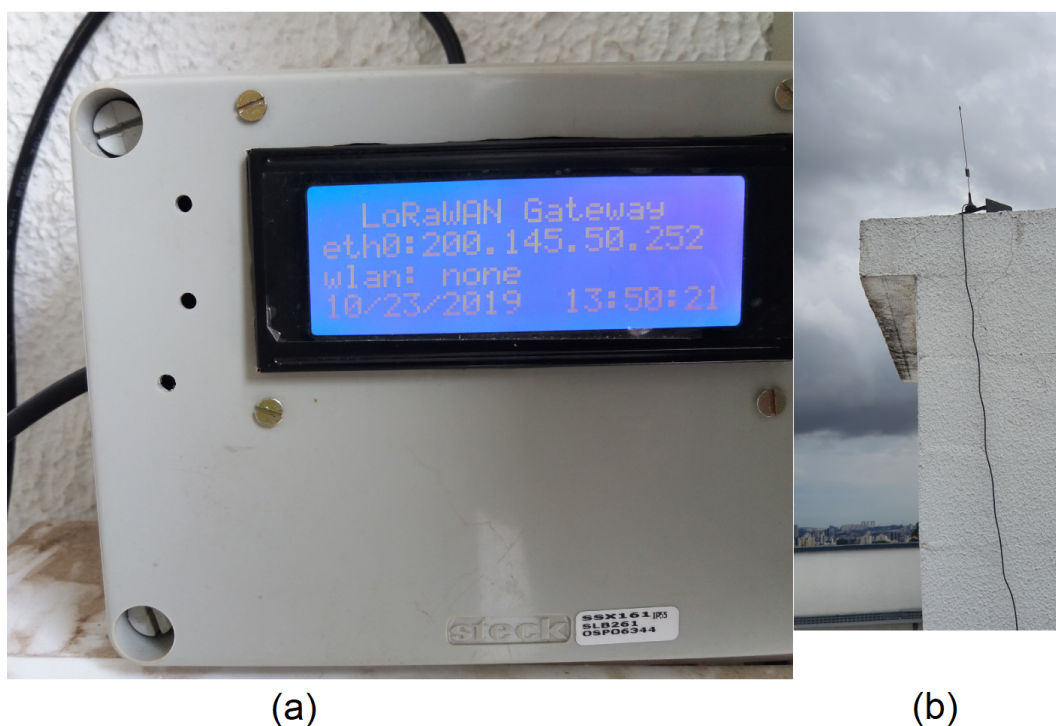
Para que o módulo *gateway* entre em funcionamento é necessário montá-lo em um local apropriado, instalar uma antena para recepção e transmissão dos dados e fazer a instalação e configuração de um Servidor de Rede e Aplicação, como por exemplo o *GatewayOS ChirpStack* (CHIRPSTACK..., 2022). Este *gateway* está montado no Laboratório Gasi (Grupo de Automação e Sistemas Integráveis) e sua antena está no ponto mais alto dentro do campus da Unesp e sua configuração é a que se segue:

- *Gateway LoRaWANRD43HAT / RD43HATGPS*;
- *Raspberry Pi*, 3 instalado com *GatewayOS ChirpStack*, que desempenha a função de Servidor de Rede *LoRaWAN* e Servidor de Aplicação;

- Antena 915MHz de 6dBi Radioenge.

Na Figura 17, pode ser visto em (a) o *gateway* operando e em (b) a antena externa para comunicação da rede *LoRaWAN*.

Figura 17 – *Gateway LoRaWAN* - Unesp Sorocaba: (a) *Gateway* instalado e (b) antena externa.



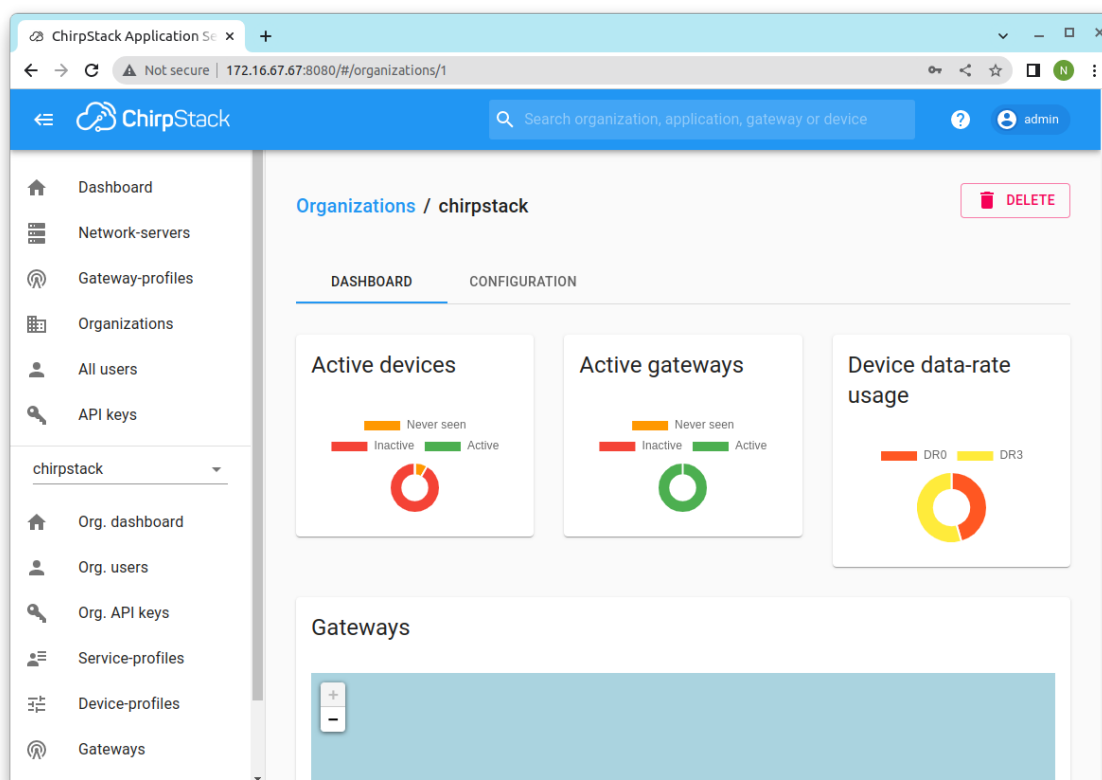
Fonte: (ALMEIDA N. C.; ROLLE, 2020).

4.4 Servidor de Rede e de Aplicação *ChirpStack* - *GatewayOS*

Para a utilização da Rede *LoRaWAN*, é necessário a utilização de um *software* gerenciador de Rede e um Servidor de Aplicação. Neste sentido, para este projeto foi utilizado o *software ChirpStack* que engloba estas funcionalidades e será descrito nesta seção.

ChirpStack é uma solução de *software* aberta que pode ser utilizado para configuração de uma rede *LoRaWAN* local, incluindo uma *interface web* para gerenciamento, Servidor de Rede *LoRaWAN* e de Aplicação, integração com os principais provedores de nuvem, banco de dados e serviços comumente usados no tratamento de dados em redes *LoRaWANs*, incluindo ainda uma *interface API* (*Application Programming Interface*) para extensão de suas características (CHIRPSTACK. . . , 2022). Na Figura 18, pode ser visto a página inicial de sua *interface* de controle, onde se apresentam alguns dados de seu uso.

Figura 18 – ChirpStack interface de controle.

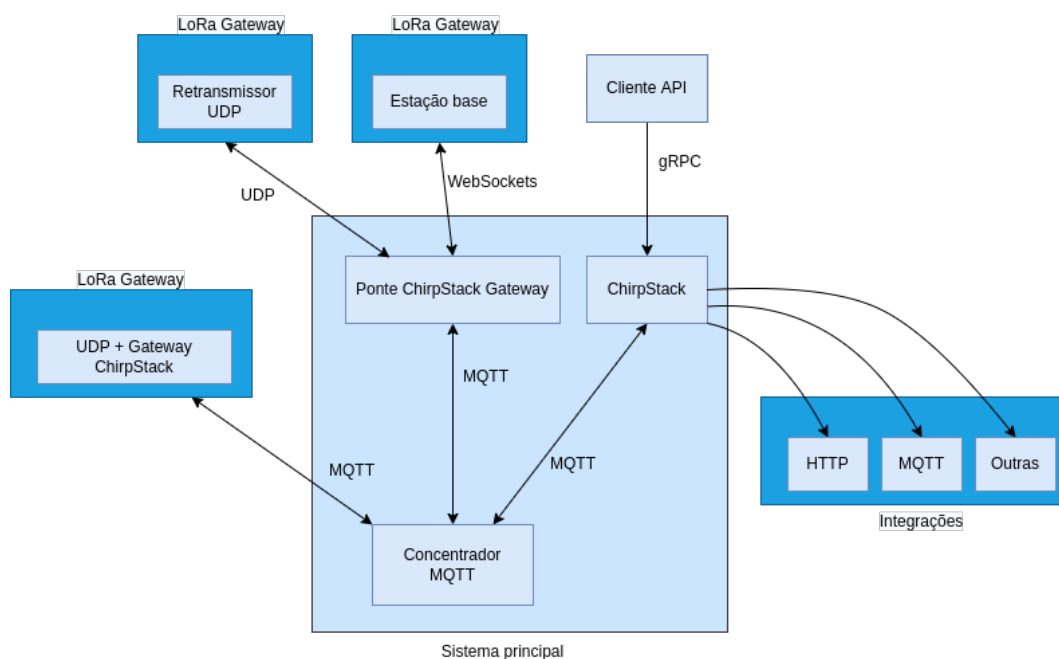


Fonte: Autor.

Com uma arquitetura baseada em serviços instalados em uma distribuição *Linux* denominada *GatewayOS*, que pode ser visto na Figura 19, suas principais características são:

- Suporte a Classe A de dispositivos *LoRaWAN* para comunicação bidirecional onde a cada envio (*uplink*) o dispositivo recebe uma transmissão de retorno (*downlink*) completando a comunicação *LoRaWAN* utilizando as duas janelas de comunicação presentes neste protocolo;
- Suporte a Classe B de dispositivos *LoRaWAN* que permite mais vias de recepção onde utiliza mais janelas de recepção se comparado à Classe A com tempo programado, utilizando um sinal sincronizador *Beacon* enviado pelo *gateway*;
- Suporte a Classe C onde os dispositivos *LoRaWAN* permanecem com a janela de recepção RX2 aberta após cada transmissão melhorando a latência, porém, consumindo mais energia para seu funcionamento;
- Taxa de velocidade de recepção adaptativa (*ADR - Adaptive Data Rate - Taxa de Dados Adaptativa*) onde o Servidor de Rede garante que o dispositivo utiliza a relação

- otimizada de potência de transmissão por velocidade de transmissão proporcionando melhor uso de energia e diminuindo as chances de colisão entre os pacotes *LoRaWAN*;
- Visualizador de dados ao vivo que provê um visualizador de dados separados por *gateways* e dispositivos e ainda apresenta todos os meta-dados e os dados puros de cada transmissão *LoRaWAN* recebida;
 - (Re)Configuração automática de canais que permite que se use as definições padrão ou estendidas de canais onde o Servidor de Rede assegura que os dispositivos estarão sempre sincronizados com a Rede;
 - Multiusuário o que permite que várias organizações trabalhem juntas no mesmo ambiente e gerencie suas aplicações individualmente;
 - Integração por *API gRPC (Google Remote Procedure Call)* e *REST (Representational State Transfer)* para interconexão com serviços externos, porém, por padrão é utilizado o concentrador *MQTT (Message Queuing Telemetry Transport - Transporte de Telemetria de Enfileiramento de Mensagens em tradução direta)*. As integrações utilizam os mais difundidos provedores de nuvem, banco de dados e plataformas de visualização;
 - Compatibilidade com as especificações *LoRaWAN 1.0* e *1.1* simultâneas incluindo especificações e parâmetros regionais de bandas.

Figura 19 – *ChirpStack* diagrama de blocos.

Fonte: Adaptado (CHIRPSTACK..., 2022).

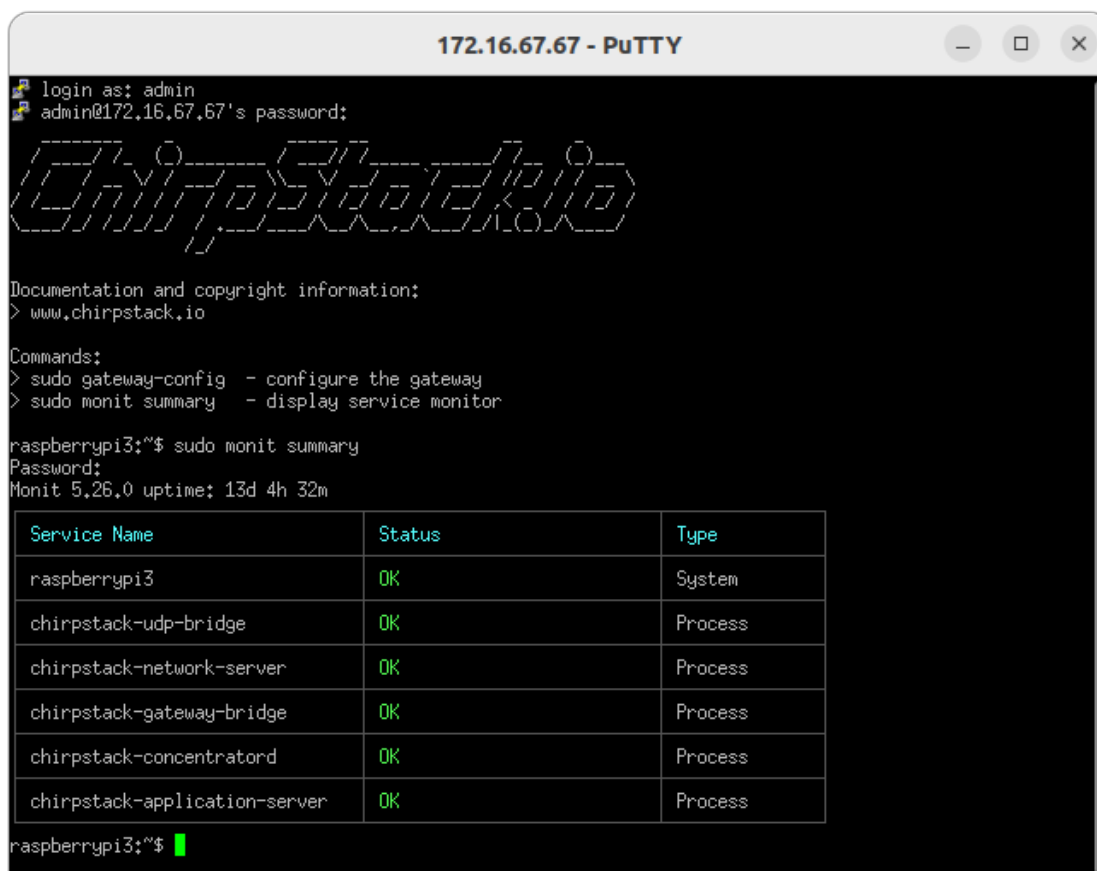
Este Sistema Operacional *GatewayOS* é uma distribuição *Linux* que funciona em todos os modelos de *SBCs Raspberry pi* e suporta vários modelos de concentradores *LoRaWAN* disponíveis no mercado de diversos fabricantes. Sua instalação é simples onde é disponibilizado uma imagem instalável no sítio do *ChirpStack*, feito o *download* desta imagem, grava-se em cartão de memória a ser inserido na *Raspberry pi*, que deve ter uma placa concentradora apropriada para *gateway LoRaWAN* conectada, como pode ser visto na Figura 20.

Figura 20 – Exemplo de *Gateway LoRaWAN ChirpStack*.



Fonte: Autor.

Após iniciado o sistema e fazendo as primeiras configurações, que podem ser vistas na documentação do *ChirpStack*, através de um terminal serial, é possível acessar o sistema do *GatewayOS* e verificar seus serviços em funcionamento, como mostrado na Figura 21.

Figura 21 – Estado dos serviços do *Gateway LoRaWAN ChirpStack* em funcionamento.

```
172.16.67.67 - PuTTY
login as: admin
admin@172.16.67.67's password:
ChirpStack

Documentation and copyright information:
> www.chirpstack.io

Commands:
> sudo gateway-config - configure the gateway
> sudo monit summary - display service monitor

raspberrypi3:~$ sudo monit summary
Password:
Monit 5.26.0 uptime: 13d 4h 32m
```

Service Name	Status	Type
raspberrypi3	OK	System
chirpstack-udp-bridge	OK	Process
chirpstack-network-server	OK	Process
chirpstack-gateway-bridge	OK	Process
chirpstack-concentrator	OK	Process
chirpstack-application-server	OK	Process

```
raspberrypi3:~$
```

Fonte: Autor.

5 Desenvolvimento

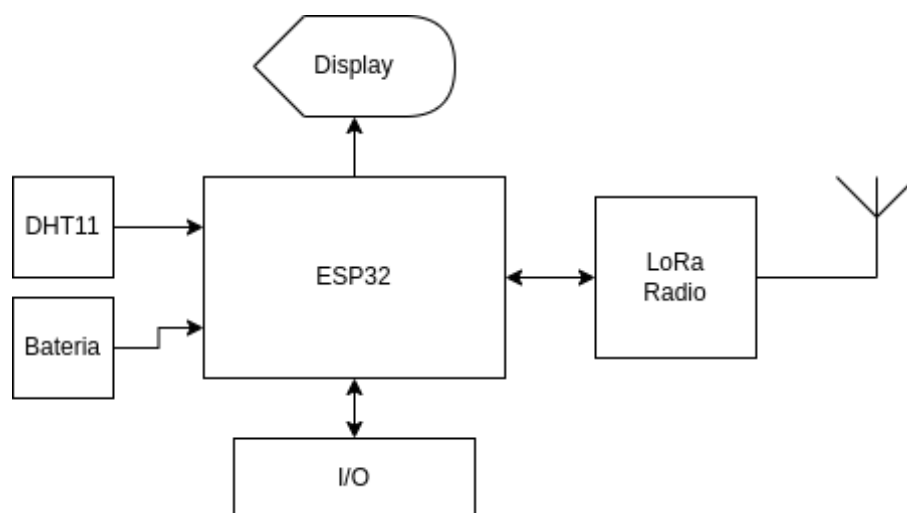
Para demonstrar o funcionamento da Rede Híbrida *LoRa* em Malha / *LoRaWAN*, será apresentada a preparação da Infraestrutura *LoRaWAN*, dos dispositivos *LoRa* em Malha, juntamente com os algoritmos desenvolvidos, o *Proxy* e o *software* da Rede Híbrida, que se divide em duas partes: o controle das rotas e o reencaminhamento dos pacotes de dados provenientes dos dispositivos da rede em malha em direção à rede *LoRaWAN*.

5.1 Dispositivos *LoRa* em malha

Para os dispositivos *LoRa* em malha, foram montados cinco dispositivos idênticos, a fim de demonstrar o funcionamento da malha. Na Figura 22, verifica-se o diagrama de blocos destes dispositivos e sua construção poder ser observada na Figura 23. Os dispositivos utilizam o seguinte *hardware*:

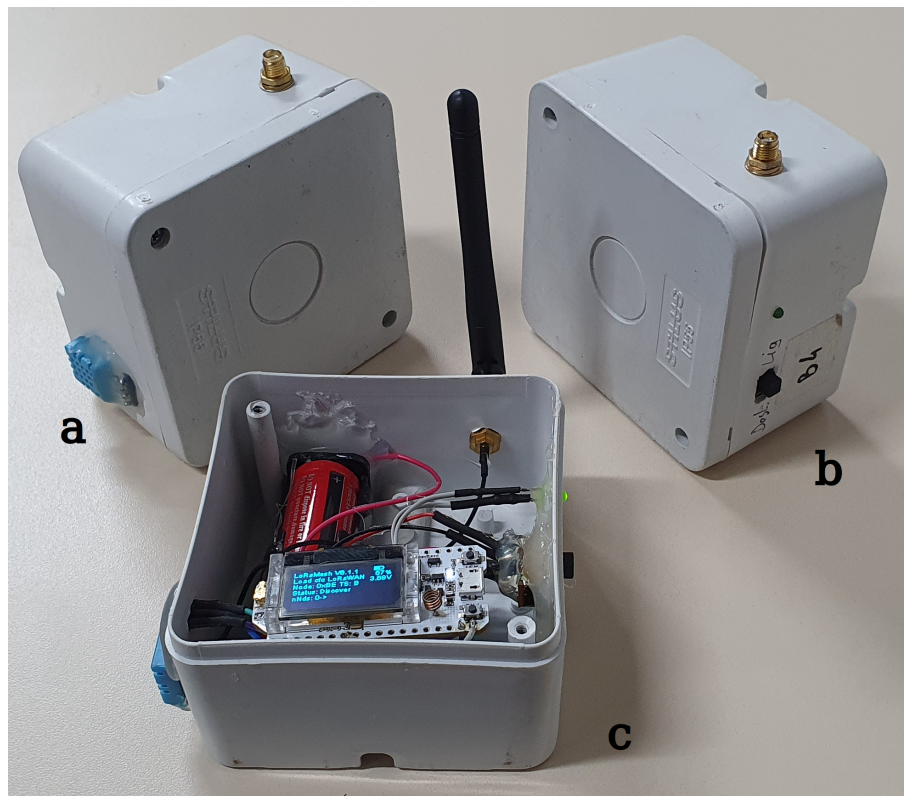
- *Heltec ESP32 Wi-Fi LoRa 32 V2*;
- Sensor de temperatura e umidade DHT11;
- Alimentação por bateria *LiPo* 3,7V 6800mAh.

Figura 22 – Diagrama de blocos dos dispositivos em malha.



Fonte: Autor.

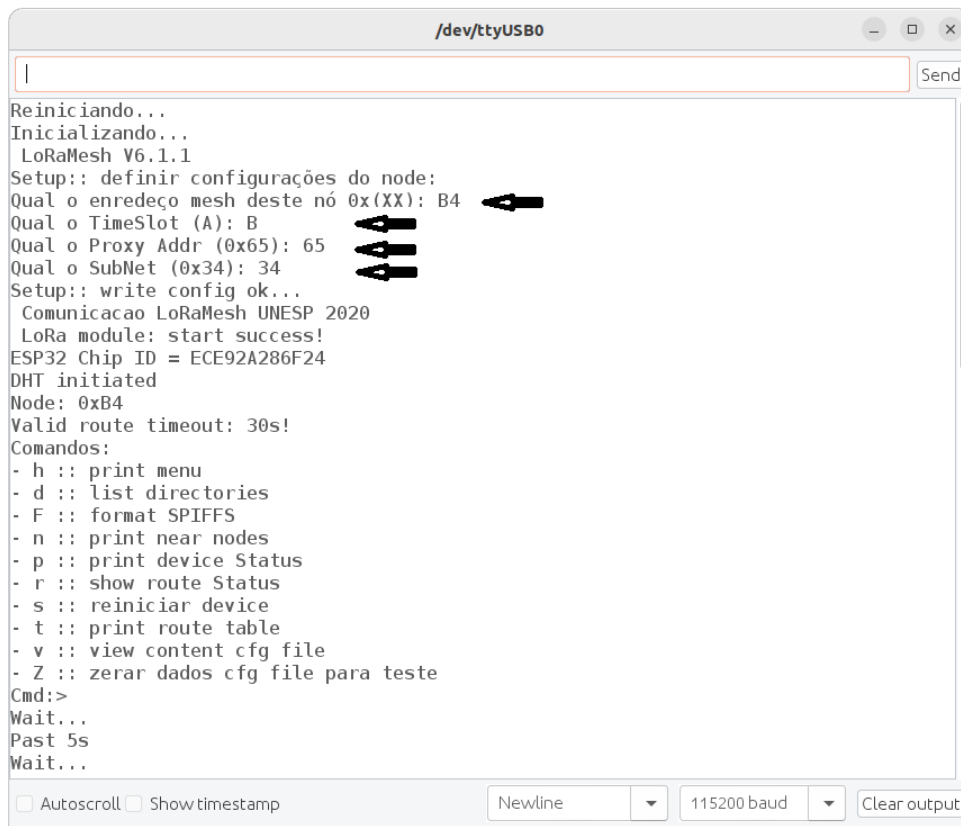
Figura 23 – Detalhes dos dispositivos *LoRa* em malha: a) sensor *DHT11*, b) identificador e c) conexões internas.



Fonte: Autor.

Os dispositivos *LoRa* em malha contém sensores *DHT11*, que são utilizados para enviar os dados de temperatura e umidade apenas para demonstrar o envio de dados dos dispositivos onde, podem ser inseridos outros sensores e atuadores através dos pinos de *I/O* digitais e analógicos disponíveis no módulo *ESP32*. Tais dados enviados devem ser tratados na Aplicação final *LoRaWAN* de forma idêntica, como é feito com um dispositivo individual *LoRaWAN*, sendo indicado aos dispositivos *LoRa* em malha que participem de uma mesma Aplicação cadastrada no Servidor de Aplicação utilizado.

Dentro da rede *LoRa* em malha, cada dispositivo deve receber um identificador ID, sendo que, este deve ser informado em sua primeira inicialização com o auxílio de um terminal serial conectado à porta *USB* do dispositivo, conforme observa-se na Figura 24. Este identificador é composto de um valor de oito bits em notação hexadecimal, como por exemplo: 0xB4. Desta forma, o dispositivo aguarda que seja informado na primeira inicialização e ainda outros parâmetros importantes, tais como: o *time stamp* (A, B, C ou D), necessário para que não ocorram envios simultâneos entre os dispositivos em malha, o método de ativação que, normalmente é ajustado para *ABP*, o ID do *Proxy* e o valor de sub-rede.

Figura 24 – Parâmetros dos dispositivos *LoRa* em Malha.

```
/dev/ttyUSB0
| Send
Reiniciando...
Inicializando...
LoRaMesh V6.1.1
Setup:: definir configurações do node:
Qual o enredoço mesh deste nó 0x(XX): B4
Qual o TimeSlot (A): B
Qual o Proxy Addr (0x65): 65
Qual o SubNet (0x34): 34
Setup:: write config ok...
Comunicacao LoRaMesh UNESP 2020
LoRa module: start success!
ESP32 Chip ID = ECE92A286F24
DHT initiated
Node: 0xB4
Valid route timeout: 30s!
Comandos:
- h :: print menu
- d :: list directories
- F :: format SPIFFS
- n :: print near nodes
- p :: print device Status
- r :: show route Status
- s :: reiniciar device
- t :: print route table
- v :: view content cfg file
- Z :: zerar dados cfg file para teste
Cmd:>
Wait...
Past 5s
Wait...
 Autoscroll  Show timestamp
Newline 115200 baud Clear output
```

Fonte: Autor.

Logo após estes parâmetros serem inseridos, o dispositivo inicia o primeiro processo de descoberta de novos vizinhos e depois a descoberta de rede que será mais bem apresentada na seção sobre o *software* em 5.1.1.

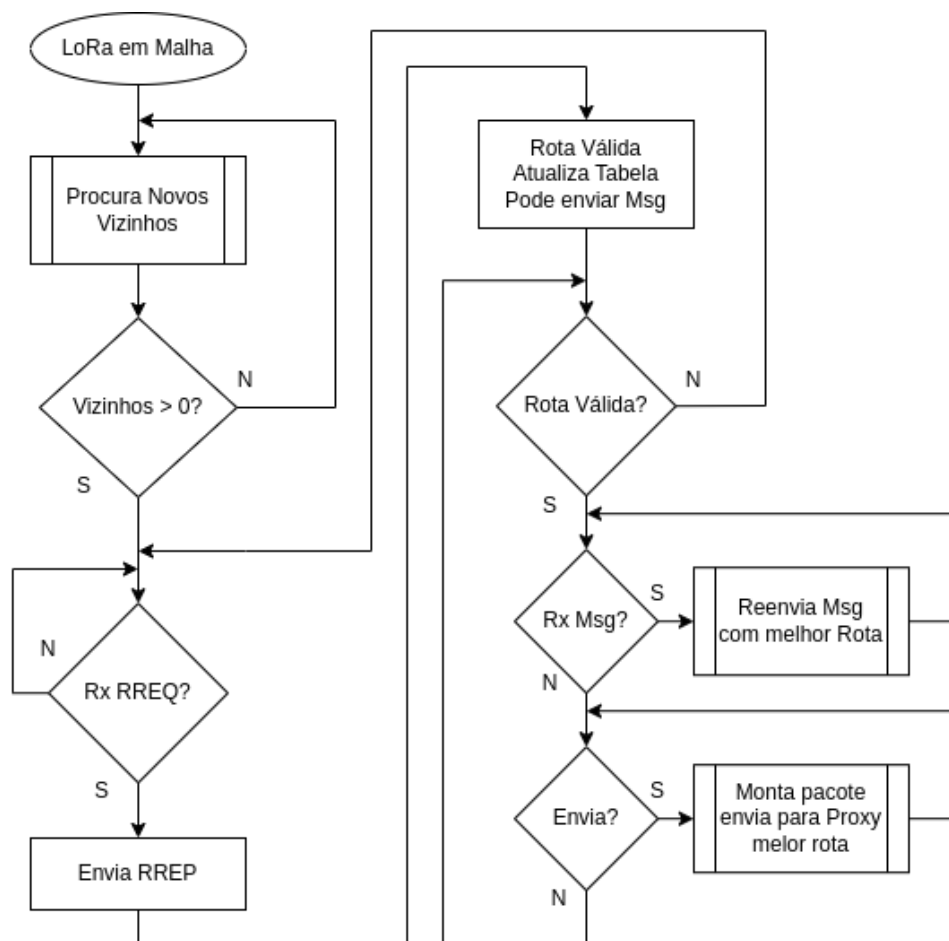
5.1.1 Algoritmo dos dispositivos *LoRa* em Malha

Após a inicialização, com todos os parâmetros do configurados, o dispositivo entra em modo de busca de novos vizinhos. Nesta busca, são verificadas algumas possibilidades: o *Proxy* está próximo, quais dispositivos estão próximos e, entre eles, existe algum dispositivo com uma rede válida que seja um caminho para o *Proxy*. A prioridade de busca entre os dispositivos é sempre pelo *Proxy* onde, se este for encontrado, o dispositivo envia um comando aos vizinhos indicando que está a um salto de distância do *Proxy*, o que caracteriza, um caminho para o envio dos dados.

No algoritmo dos dispositivos da Malha, que pode ser visto na Figura 25, a função mais importante é a de recepção que faz todo o controle, dependendo do tipo de mensagem que é recebido uma função específica é iniciada. Em linhas gerais, três ações são controladas e, uma máquina de estados se encarrega do gerenciamento da rede e encaminhamento das mensagens, esta máquina de estados trabalha em três estados ou ações:

- Descoberta: descoberta de novos vizinhos e de rede válida próxima;
- Gerenciamento de Rede: *Proxy* como vizinho e pedido de rota com os controles *RREQ* (*Route Requisition* - Requisição de rota) e *RREP* (*Route Reply* - Resposta de rota);
- Envio de mensagens: envio e reencaminhamento de mensagens com rota para o *Proxy*.

Figura 25 – Algoritmo dos dispositivos da Malha.



Fonte: Autor.

Uma rota é caracterizada pela soma, em módulo, dos níveis *RSSI* (*Received Signal Strength Indicator* - nível de recepção do sinal de RF) que são registrados e atualizados em cada recepção realizada pelo dispositivo. Através do comando *RREQ* recebido, uma rota pode ser formada atualizando-se uma tabela de dispositivos próximos e o valor *RSSI* recebido. Então, um comando *RREP* é retornado, indicando que esta rota foi aceita e criada na tabela de rotas interna. Com os dispositivos conhecidos e a tabela atualizada, identifica-se a melhor rota, sempre em direção ao *Proxy*, como dito, pela soma dos módulos dos valores de *RSSI* de cada dispositivo e então, o menor valor de *RSSI* identifica uma

rota, com menos saltos e válida. Na Figura 26, se verifica que o dispositivo 0xB4 apresenta uma rota para o *Proxy* (ID 0x65) e, este possui um valor *RSSI* de -31dbm, indicando uma rota para o *Proxy* com custo 31 (*RSSI* em módulo) com um salto, como indicado pelo valor "Hp".

Figura 26 – Rota do dispositivo 0xB4 para o *Proxy*.

```

:: vdat: 4.09V
Cost of route 1: 31
Best route: 1
Route 1: 0x65 Rx: -31 Hp: 1
Chg sts to LoRaMesh msg

```

Fonte: Autor.

Tão logo o dispositivo esteja em uma rede válida, inicia-se o processo de envio de mensagens em direção ao *Proxy*, continuamente e a cada tempo pré-determinado, o dispositivo faz o envio do pacote de dados que inclui o *ID* do dispositivo, um valor único que identifica cada pacote denominado *seq* (*sequence*), o valor da carga da bateria em porcentagem (*bt*), um contador (*ct*), que é atualizado a cada envio, o valor da temperatura (*tp*), umidade (*hd*) e um campo de rota (*rt*). Na Figura 27, pode-se observar um pacote de dados a ser enviado pelo dispositivo 0xB4.

Figura 27 – Pacote de dados do dispositivo 0xB4.

```

Chg sts to LoRaMesh msg
Data to send: {"id":"B4","sq":80,"bt":85,"ct":1,"tp":19.00,"hd":21.00,"rt":"B465"}
Snd LoRaMesh msg... (Node 0xB4 sq 80)

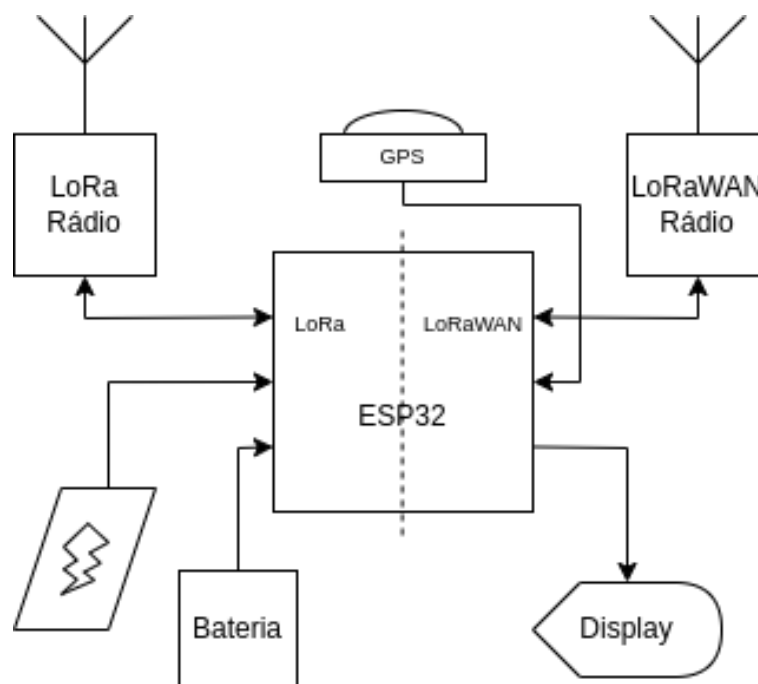
```

Fonte: Autor.

O reencaminhamento do pacote de dados é iniciado ao receber um pacote de um dispositivo vizinho, verificando seu destino, através de parâmetros de controle interno. Uma vez identificado que o pacote não é para o dispositivo local e, sendo para o *Proxy*, o dispositivo reenvia este pacote segundo a "melhor rota" em direção ao *Proxy*, incluindo a informação de rota do dispositivo local que faz o reenvio.

5.2 Dispositivo *Proxy*

O dispositivo *Proxy*, torna possível o presente trabalho pois, executa a função de gerenciamento da rede em malha e o reencaminhamento dos pacotes *LoRa*, provenientes dos dispositivos da malha, em direção à rede *LoRaWAN* através do processo de virtualização dos dispositivos da malha. A virtualização é proporcionada, configurando-se os parâmetros *LoRaWAN* dos dispositivos da malha diretamente no *Proxy*. Desta forma, o dispositivo *Proxy* está dividido em duas partes, como mostra o diagrama de blocos da Figura 28.

Figura 28 – Diagrama de blocos do *Proxy*.

Fonte: Autor.

Dois rádios *LoRa* são utilizados no dispositivo *Proxy*, sendo um para comunicação na rede *LoRa* em malha e o outro para comunicação na rede *LoRaWAN* e virtualização dos dispositivos da malha. Ambas as redes, *LoRa* e *LoRaWAN*, necessitam de configurações específicas contidas no *software* do *Proxy*. O rádio *LoRa* desempenha a função de recepção e transmissão dos controles da rede em malha e, utiliza a configuração de um canal com frequência fixa para todos os dispositivos da rede em malha sendo que, o *SF* configurado determina este parâmetro para toda a rede em malha. Para os testes, foi especificado *SF7*, por se tratar de um valor que permite a "menor" distância *LoRa* entre os dispositivos, o que facilita os testes da rede, porém, outros valores de *SFs* fazem com que maiores distâncias sejam alcançadas aumentando-se o alcance da malha, fazendo com que, os dispositivos transmitam os dados para o *Proxy* com apenas um salto, o que dificulta o funcionamento da rede em malha com vários saltos.

O rádio *LoRaWAN*, é responsável pelo envio dos pacotes dos dispositivos da malha para a rede *LoRaWAN*, sendo que, a cada pacote recebido de um dispositivo da malha, o *software* de controle faz o processo de virtualização do dispositivo *LoRaWAN*, previamente configurado na Aplicação e no *Proxy*, fato este, que permite que os dispositivos da rede em malha transmitam seus dados para a rede *LoRaWAN* como parte integrante desta rede e utilizando a ativação *ABP*. Outro fator importante, para a configuração com dois rádios, é que a comunicação *LoRaWAN* trabalha com vários canais de frequências diferentes e que são ajustados dinamicamente, através do protocolo *LoRaWAN*, de acordo com as configurações regionais, sendo valores diferentes de frequência para os canais de *uplink*

e *downlink*, e desta forma, tendo um rádio para cada função, faz com que o protocolo *LoRaWAN* seja aplicado corretamente.

O hardware do dispositivo *Proxy* conta com um módulo microcontrolador *ESP32 DevKitC* pois, neste módulo, tem-se os terminais *I/O* disponíveis para a conexão de todos os periféricos necessários, que são:

- Rádio *NiceRF* para comunicação *LoRa* em malha, em porta *SPI1*;
- Rádio *NiceRF* para a rede *LoRaWAN*, conectado à porta *SPI2*;
- *Display OLED* para visualização de mensagens internas;
- Receptor *GPS (Global Position System)* para registro dos eventos de transmissão;
- Alimentação por bateria *LiPo 26650* de 3,7V 6800mAh;
- Tomada para ligação de painel solar, para funcionamento em campo.

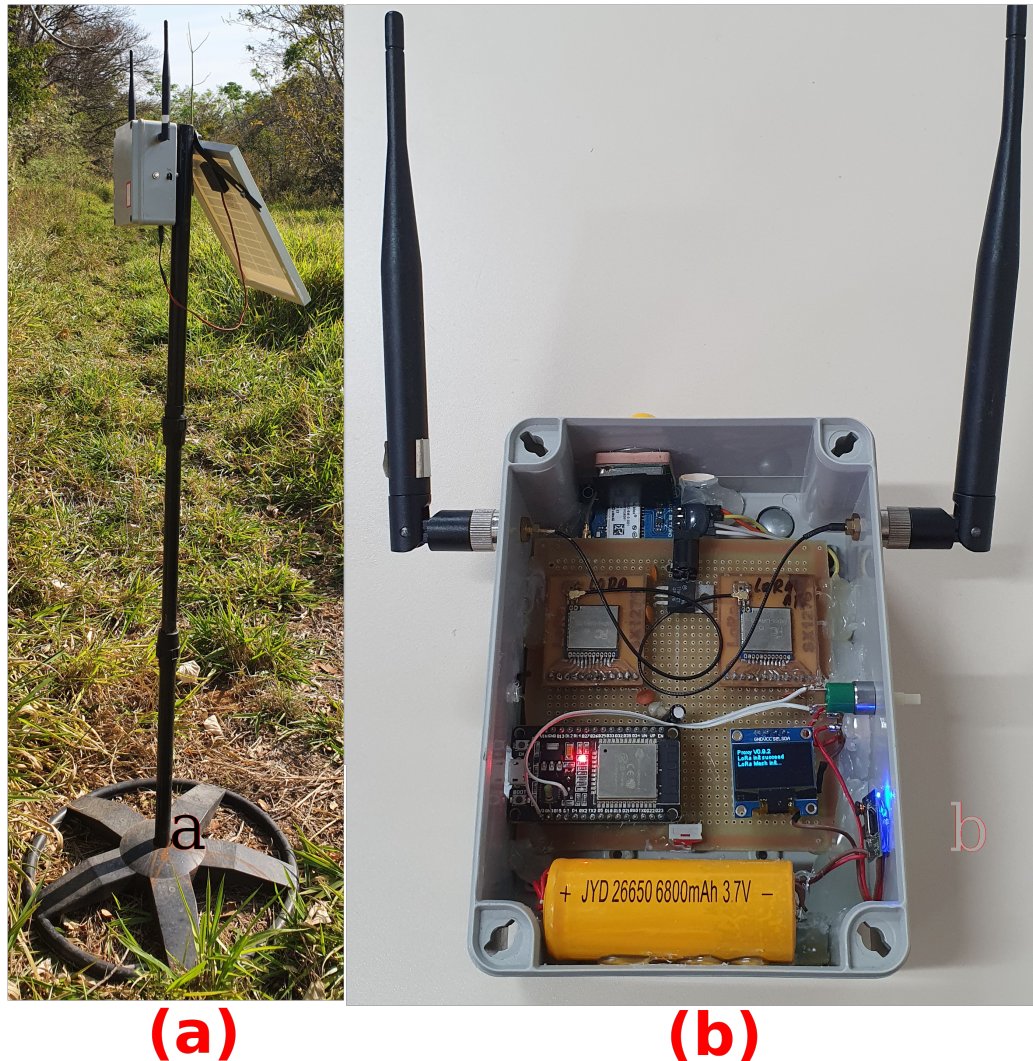
Os aspectos construtivos e o posicionamento do *Proxy* em campo podem ser vistos na Figura 29, bem como, as antenas e a caixa que abriga o equipamento.

5.3 Integração *LoRa* em Malha – *LoRaWAN*

Para a integração da rede *LoRa* em malha com a infraestrutura *LoRaWAN*, o *Proxy* possui duas funções distintas controladas pelo seu algoritmo, estas funções são:

- **Gerenciador da rede em Malha:** esta função tem a tarefa de gerenciar todas as ações da rede em malha e, permite que todos os dispositivos conectados em malha possam enviar seus pacotes de dados para o *Proxy*, onde este os recebe e agenda sua retransmissão para a rede *LoRaWAN*, segundo uma fila de envio;
- **Redirecionador *Proxy*:** os dispositivos da rede em malha são virtualizados, pelo *Proxy*, este reencaminha o pacote de dados para a rede *LoRaWAN* e, desta forma, cada dispositivo da malha representa um único dispositivo identificável na rede *LoRaWAN*, estes dispositivos devem ser previamente configurados no Servidor de Rede *LoRaWAN* e Servidor de Aplicação.

Como redirecionador, o uso de uma integração baseada em um *Proxy*, onde se habilita a virtualização dos dispositivos operando na mesma *interface* rádio possibilita que vários dispositivos compartilhem este mesmo rádio. Do ponto de vista operacional, o *Proxy* deve se comportar como vários dispositivos virtuais alternadamente, mas a segurança da rede *LoRaWAN* não deve ser violada (SISINNI et al., 2019).

Figura 29 – (a) *Proxy* em campo e (b) aspectos construtivos.

Fonte: Autor.

Para manter a segurança desejada, o *Proxy* desenvolvido e os dispositivos da rede em malha apresentam algumas funcionalidades em comum, com as seguintes características:

- cada dispositivo da malha, identificado por seu *ID*, deve enviar seu pacote de dados utilizando uma "função de encriptação", dentro da rede em malha, para o *Proxy*;
- o *Proxy* deve manter configurado as chaves *LoRa WAN*, únicas de cada dispositivo;
- o *Proxy*, deve decodificar o pacote de dados e reencaminhar para a rede *LoRa WAN*, segundo as chaves configuradas.

O *Proxy* implementa todo o gerenciamento da camada de enlace *LoRa WAN*, enquanto a parte *MAC* das mensagens dos dispositivos da malha é obtida através dos parâmetros *LoRa WAN* dos dispositivos, cadastrados no *Proxy*, quando for inserido um

novo dispositivo da malha, devem ser cadastrados os novos parâmetros referentes ao novo dispositivo.

O desenvolvimento deste *Proxy*, permite manter a confidencialidade em ambos os casos: rede em malha e dados *LoRaWAN*. Na situação do *Proxy* sofrer um ataque a segurança da rede *LoRaWAN* não fica comprometida, desde que somente o *Proxy* detenha as chaves de sessão de rede dos dispositivos conectados.

Vale a pena mencionar que, regulações locais podem impor restrições quanto ao uso de bandas de rádio e limitar no número de transmissões, este fator é conhecido como *Duty Cycle (DC)*. O *DC* é definido como sendo a relação entre a duração da transmissão e o tempo entre duas transmissões consecutivas. Sendo assim, cada *Proxy* deve respeitar os limites entre todas as mensagens enviadas pela rede em malha e toda as mensagens enviadas para a rede *LoRaWAN*. Com a arquitetura do *Proxy* proposta, o impacto de limites rigorosos de *DC* pode ser mitigado adotando-se estratégias de gerenciamento de transmissões, conectando-se vários *Proxies* para dividir as estatísticas de transmissão entre a rede *LoRa* em malha e *LoRaWAN* (este tópico será discutido em trabalhos futuros), entre outras opções.

Quanto ao roteamento, a importância de cenários dinâmicos força o uso de protocolos de roteamento reativos, tais como *AODV* e *DSR (Dynamic Source Routing)*, discutidos por *Varghese et al. (2019)*. Em particular, para o presente trabalho, o uso do protocolo *AODV* para gerenciamento da rede em malha é preferido e motivado por ser um protocolo já bastante utilizado em redes de sensores sem fio e apresenta uma excelente performance se comparado com o protocolo *DSR* (VARGHESE et al., 2019).

5.3.1 Algoritmo gerenciador da rede *LoRa* em Malha

Para a programação destes módulos, foi utilizado a *IDE (Integrated Development Environment) Arduino*, pois apresenta a possibilidade de programar diversos microcontroladores e sistemas embarcados, dentre os quais, o microcontrolador *ESP32* se encontra. Também por possuir uma grande comunidade ativa de desenvolvedores permitindo que o trabalho seja direcionado para o projeto em desenvolvimento, sendo que, muitas bibliotecas já encontram se desenvolvidas pela comunidade, bem como as bibliotecas do módulo *ESP32* e do rádio *LoRa*, disponibilizadas pelo fabricante *Heltec*.

O gerenciamento da rede em malha segue algumas características do protocolo de roteamento *AODV*, sendo as que se seguem:

- mensagens "*Hello*": quando um dispositivo é iniciado, este envia mensagens do tipo "*Hello message*" para encontrar dispositivos próximos. Quando um vizinho recebe esta mensagem, responde com uma mensagem "*Hello Ok*", indicando o recebimento e retornando alguns dados, como por exemplo, seu *ID*, *RSSI* e um identificador

numérico aleatório denominado "seq" (*sequence* - sequência), que funciona como identificador único desta mensagem. Este identificador *seq* é utilizado para evitar que uma mensagem seja respondida mais de uma vez e, que será visto com mais detalhes mais a frente. Este processo é responsável por criar a tabela de dispositivos vizinhos e se repete até que os dispositivos encontrem o *Proxy*;

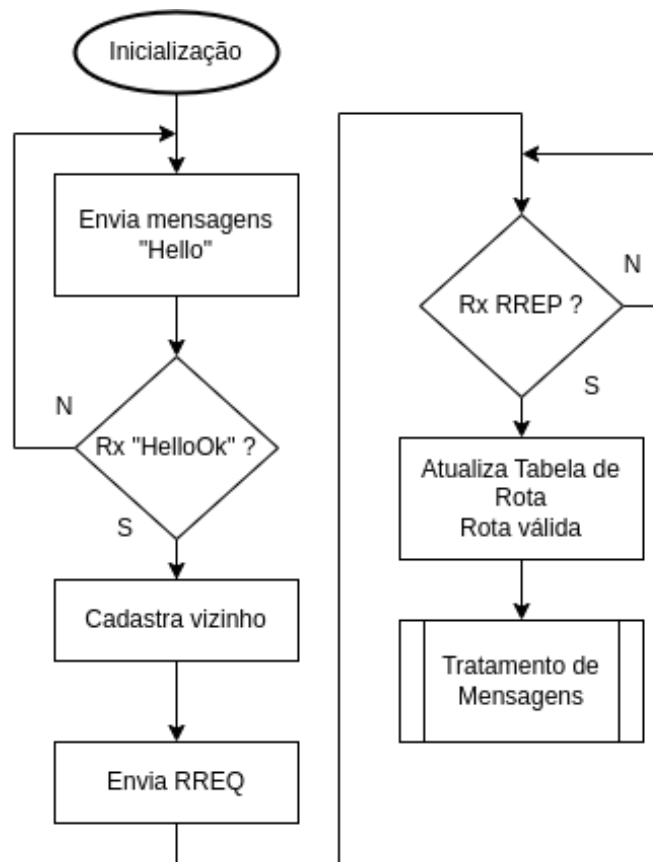
- mensagens *RREQ* e *RREP*: após o processo de "Hello", inicia-se o processo de requisição de rota entre os dispositivos da malha: *RREQ*, o *Proxy* então, envia aos dispositivos mais próximos o pedido de requisição *RREQ* e aguarda uma resposta *RREP*. Nesta resposta existem alguns parâmetros tais como: o *ID* do dispositivo, seus vizinhos e suas posições na malha (saltos para o *Proxy*). Recebido esta resposta *RREP*, indica que já temos uma ligação de rede com este dispositivo. Este processo se repete até que todos os dispositivos da rede recebam uma requisição *RREQ* e respondam o *RREP* correspondente. As mensagens de *RREP* que retornam trazem registrados a rota de retorno e, se um dispositivo tem mais de um vizinho esta mensagem *RREP* retorna de vários caminhos, apresentando as várias rotas possíveis para um determinado dispositivo na malha.
- parâmetro "seq" (*sequence*): todas as mensagens trocadas pelos dispositivos utilizam um parâmetro denominado "seq", que é o mecanismo que identifica de forma única cada mensagem. Desta forma o algoritmo é capaz de responder apenas uma vez às mensagens e evita um fato indesejado no gerenciamento de uma rede em malha, que se chama "flood" (inundação), onde as mensagens ficam indo e voltando entre os dispositivos e não seguem o caminho correto em direção ao *Proxy*, por exemplo, e ainda este mecanismo é utilizado para diminuir a sobrecarga de mensagens que é um problema recorrente no gerenciamento das redes em malha.

O algoritmo das fases de "Hello" e *RREQ/RREP* é apresentado na Figura 30, onde pode-se observar, em linhas gerais, o desenvolvimento da parte de gerenciamento da rede em malha, que é parte integrante do *software* do *Proxy* e nos dispositivos da malha.

5.3.2 Algoritmo redirecionador: *Proxy*

O dispositivo *Proxy*, pelo fato de gerenciar a rede em malha, deve ser compatível com a rede *LoRaWAN* em Classe C, a fim de garantir que os dados dos dispositivos da malha sejam reencaminhados para a rede *LoRaWAN*. Para que esta função se torne possível, o módulo do *software* do *Proxy* responsável por reencaminhar os pacotes para a rede *LoRaWAN* e a virtualização, utiliza a biblioteca *LMIC* (*LoRaWAN Made In C*) (LORAWAN-NODE, 2019), bastante difundida entre os desenvolvedores *LoRaWAN* para *Arduino*, sendo esta biblioteca totalmente compatível com o rádio *LoRa SX1276* e a rede *LoRaWAN*.

Figura 30 – Algoritmo de gerenciamento da rede em malha.



Fonte: Autor.

Após o processo de configuração da rede em malha, com as rotas conhecidas entre todos os dispositivos, o *Proxy* alterna para o tratamento de mensagens dos dispositivos da malha para fazer o reencaminhamento para a rede *LoRaWAN* através da virtualização. Para o funcionamento correto, todos os dispositivos da rede em malha devem ser cadastrados na rede *LoRaWAN* e suas chaves devem ser configuradas no *Proxy*, a fim deste poder fazer o reencaminhamento dos pacotes de dados para a Aplicação a ser utilizada. Mais detalhes desta Aplicação serão vistos outro tópico, neste capítulo.

Os parâmetros dos dispositivos da malha, devem ser cadastrados no *Proxy*, em momento de compilação, por se tratar de um projeto experimental e, para próximas versões serão adicionadas funções para cadastrar estes dados via terminal serial, com armazenamento em memória de dados, presente no microcontrolador *ESP32*. As chaves *LoRaWAN* dos dispositivos que, devem ser configurados no *Proxy* são:

- o *ID* do dispositivo na malha;
- o nome do dispositivo "*Device name*";
- o *Network Session Key* do dispositivo (*NwkSKey*);

- o *Application Session Key* do dispositivo (*AppSKey*);
- o *Device Address* do dispositivo (*DevAdr*).

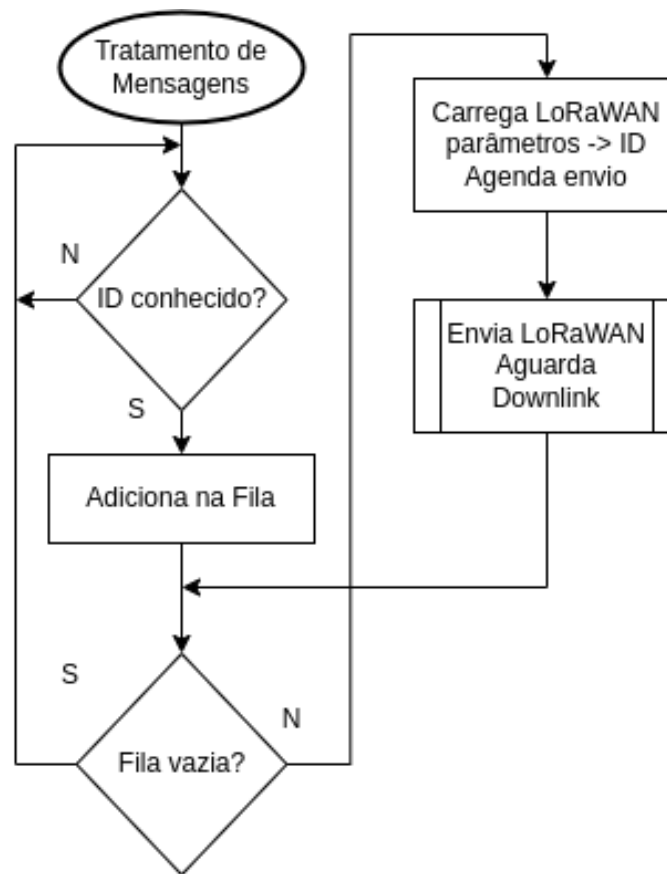
Estes dados são cadastrados no Servidor de Aplicação, por exemplo, o *ChirpStack*, que será demonstrado na próxima seção. Assim que cadastrados os parâmetros *LoRaWAN*, estes são configurados no *Proxy*, tornando os dispositivos da malha conhecidos e autorizados no *Proxy*. Se algum dispositivo não cadastrado tentar enviar algum pacote para o *Proxy*, este não será aceito e os dados serão descartados.

Com a rede em malha configurada e o *Proxy* na fase de tratamento de mensagens, o processo de reencaminhamento destas mensagens pelo *Proxy* ocorre da seguinte forma: ao receber um pacote de dados de um dispositivo da malha, sendo um dispositivo conhecido, este pacote é registrado em uma fila *FIFO* (*Fist In First Out*) e aguarda para ser enviado para a rede *LoRaWAN*, esta fila existe para gerenciar os processos internos e os vários pacotes recebidos pelo *Proxy* em um curto período de tempo, coordenar a virtualização e garantir o funcionamento do protocolo *LoRaWAN* de forma integral, diminuindo assim problemas de o microcontrolador estar ocupado e gerenciando melhor o tempo.

Os pacotes da fila são enviados por ordem de chegada e, para não acontecer atrasos, a prioridade do algoritmo é enviar todos os dados que estão em fila no momento em que tiver um dado disponível, sendo que o *Proxy* utiliza os dados cadastrados dos dispositivos para que, de forma virtual, a cada pacote a ser transmitido para a rede *LoRaWAN*, o *Proxy* assume a "identidade" do dispositivo, utilizando os parâmetros pré-cadastrados e fazendo o envio representando este dispositivo na rede *LoRaWAN*.

Uma situação importante a ser observada é que, o *Proxy* está preparado apenas para o método de ativação *ABP*, até o presente desenvolvimento, e devem-se cadastrar os dispositivos com a função de desabilitar a validação de contagem de pacotes de dados pois, mesmo com a ativação em *ABP*, se um dispositivo parar e voltar a retransmitir, o contador de pacotes se reinicia e o dispositivo consegue reiniciar a transmissão sem problemas.

O módulo do algoritmo referente ao tratamento de mensagens, do *Proxy* para a rede *LoRaWAN*, é apresentado na Figura 31, onde pode-se observar o processo de virtualização dos parâmetros, dos dispositivos da rede em malha que, ocorre a cada envio para a rede *LoRaWAN* através do carregamento destes parâmetros individuais dos dispositivos da malha.

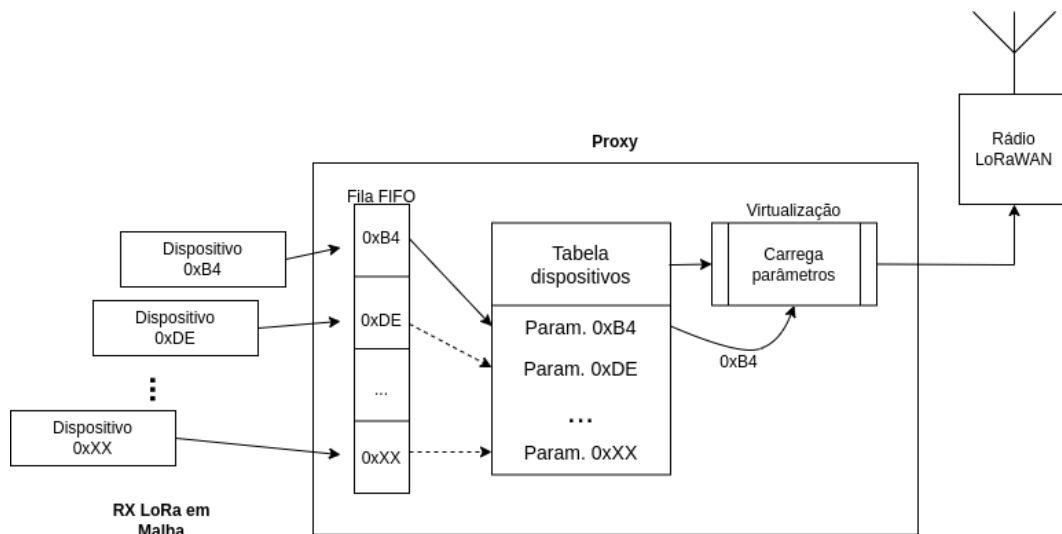
Figura 31 – Algoritmo de reencaminhamento do *Proxy*.

Fonte: Autor.

5.3.3 Virtualização dos dispositivos da malha para a rede *LoRaWAN*

O processo de integração, entre as redes em malha e a rede *LoRaWAN*, apresenta alguns desafios, tais como: quantos dispositivos serão ligados, o processo de ativação, quantos saltos cada dispositivo está em relação à rede *LoRaWAN* e ainda outros. O presente desenvolvimento, possibilita que vários dispositivos, ligados em malha em uma rede *LoRa* enviem pacotes de dados para uma rede *LoRaWAN* através de um *Proxy* que está "entre" a rede *LoRa* em Malha e a rede *LoRaWAN* e dispostos em campo, nas áreas de sombra, onde a rede *LoRaWAN* não está presente. Para que o dispositivo *Proxy*, este sim posicionado em área com cobertura *LoRaWAN*, faça o reenvio dos pacotes de dados da malha para a rede *LoRaWAN*, o processo interno ao *Proxy* que permite esta operação é a virtualização dos dispositivos da malha, que faz com que o *Proxy* seja identificado na rede *LoRaWAN* como um dispositivo "comum" nesta rede.

Para que esta virtualização ocorra, os dispositivos devem ter suas chaves, já citadas, cadastradas no Servidor de Aplicação e em uma tabela de dispositivos conhecidos no *Proxy*, comum a todos estes dados está o *ID* de cada dispositivo da malha. Na Figura 32, pode-se observar o fluxo de dados dos dispositivos para a virtualização.

Figura 32 – Virtualização de dispositivos no *Proxy*.

Fonte: Autor.

A virtualização *LoRaWAN* ocorre da seguinte forma: cada dispositivo da malha envia seu pacote de dados pela rede *LoRa* em malha, quando o *Proxy* reconhece como um dispositivos da sua rede este pacote fica referenciado na fila, a cada envio a fila é atualizada (*FIFO*), o módulo de virtualização carrega os dados do dispositivo segundo a tabela de dispositivos cadastrados, altera os parâmetros *LoRaWAN* através de comandos específicos da biblioteca *LMIC* e executa o envio através de uma *callback* de envio *LoRaWAN*, esta *callback* de envio trabalha sobre o protocolo *LoRaWAN* o que permite garantir a compatibilidade com os padrões da rede *LoRaWAN*.

Este processo tem-se mostrado efetivo para o método de ativação *ABP* e, necessita que no Servidor de Aplicação, cada dispositivo tenha habilitado o parâmetro de *Disable frame-counter validation* (Desabilitar a validação de contagem de pacotes), pois na troca de parâmetros dos dispositivos para uma nova transmissão virtual, o contador de pacotes não é restaurado de forma eficaz.

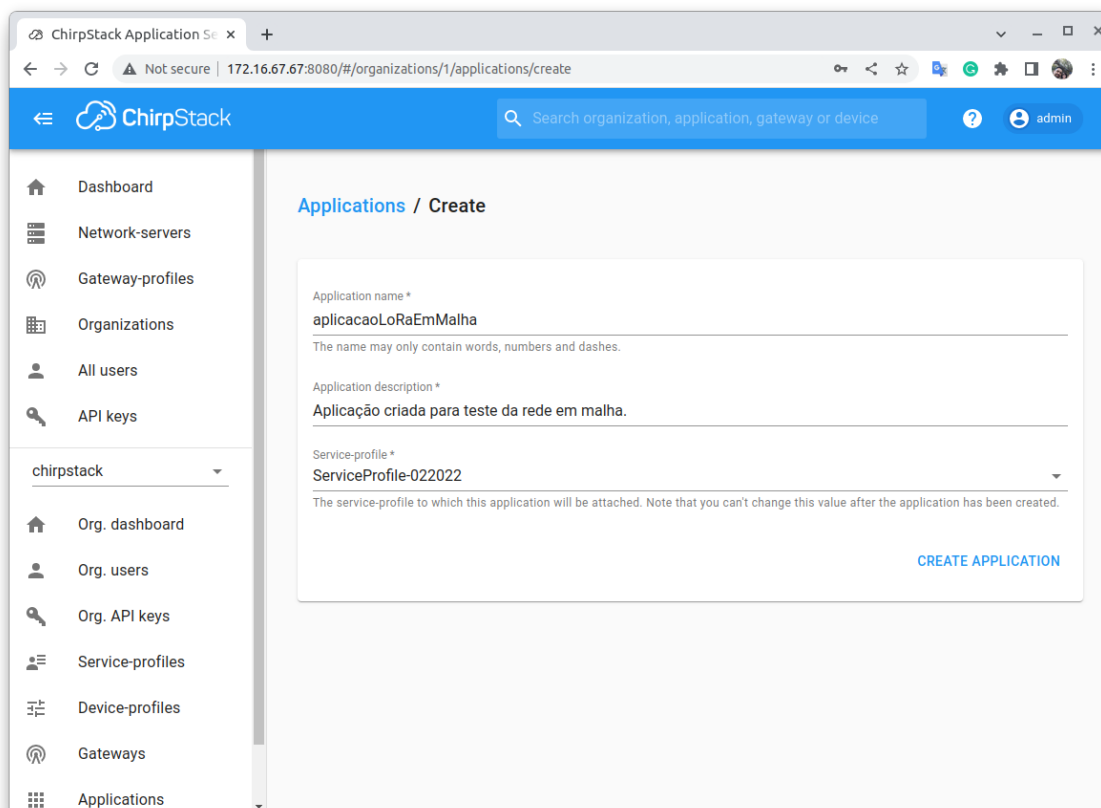
Esta virtualização se mostra funcional para o método de ativação *OTAA*, porém, esta função está presente como trabalhos futuros.

5.4 Infraestrutura *LoRaWAN* (*ChirpStack*)

Para o funcionamento da rede Híbrida *LoRa* em malha / *LoRaWAN*, a configuração dos dispositivos da malha deve ser feita primeiramente no Servidor de Aplicação, por exemplo, no *ChirpStack*, onde cada dispositivo da malha e o *Proxy* devem ser adicionados em uma Aplicação, e os dados de configuração *LoRaWAN* devem ser configurados no *Proxy* juntamente com o *ID* dos dispositivos da malha.

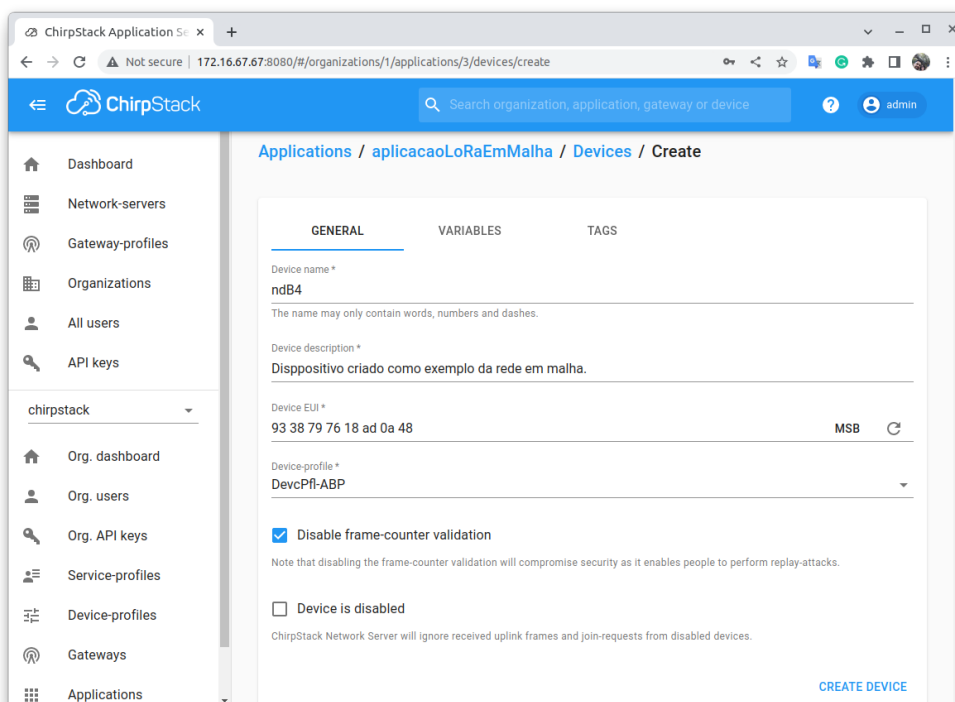
A seguir, apresenta-se a configuração de uma Aplicação inserida no *ChirpStack* e as configurações pertinentes dos dispositivos a serem conectados onde, inicialmente faz-se o cadastro de uma Aplicação na *interface* do *ChirpStack*, como é apresentado na Figura 33.

Figura 33 – Criando uma Aplicação no *ChirpStack*.



Fonte: Autor.

Dentro da nova Aplicação criam-se os dispositivos, como exemplo, será criado um dispositivo com o *ID* 0xB4 da rede em malha e a este será dado o "*Device name*": "ndB4" que é o identificador do dispositivo na Aplicação, como mostra a Figura 34, salienta-se a geração automática das chaves "*Device EUI*", a indicação de um "*Device-profile*" que indica um perfil de dispositivo e a ativação da função "*Disable frame-count validation*", explicada anteriormente que, juntamente com o método de ativação *ABP*, permite que os dispositivos se comuniquem de forma assíncrona com o Servidor de Rede.

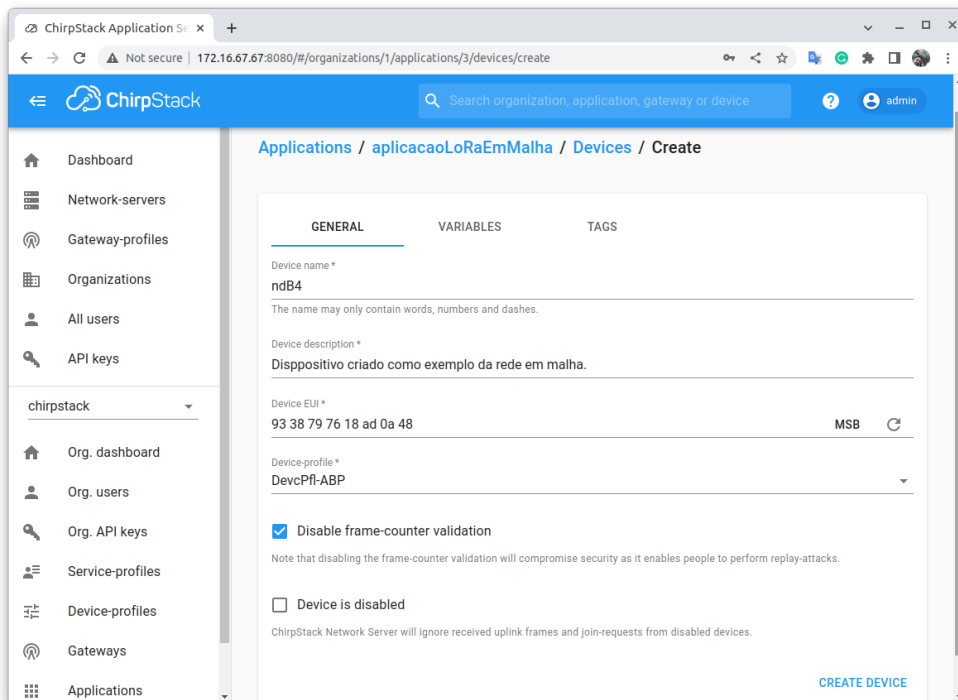
Figura 34 – Adicionando um dispositivo da malha no *ChirpStack*.

Fonte: Autor.

Na sequência, o *ChirpStack* orienta a continuação do cadastro dos dados de ativação do novo dispositivo, sendo estes dados muito importantes para que a comunicação *LoRaWAN* ocorra, também é interessante frisar que estes dados devem estar anotados pois é necessário que sejam cadastrados no *Proxy* para este poder fazer o reenvio dos dados do dispositivo ora cadastrado, a Figura 35 mostra o término do cadastro do dispositivo 0xB4 no *ChirpStack*.

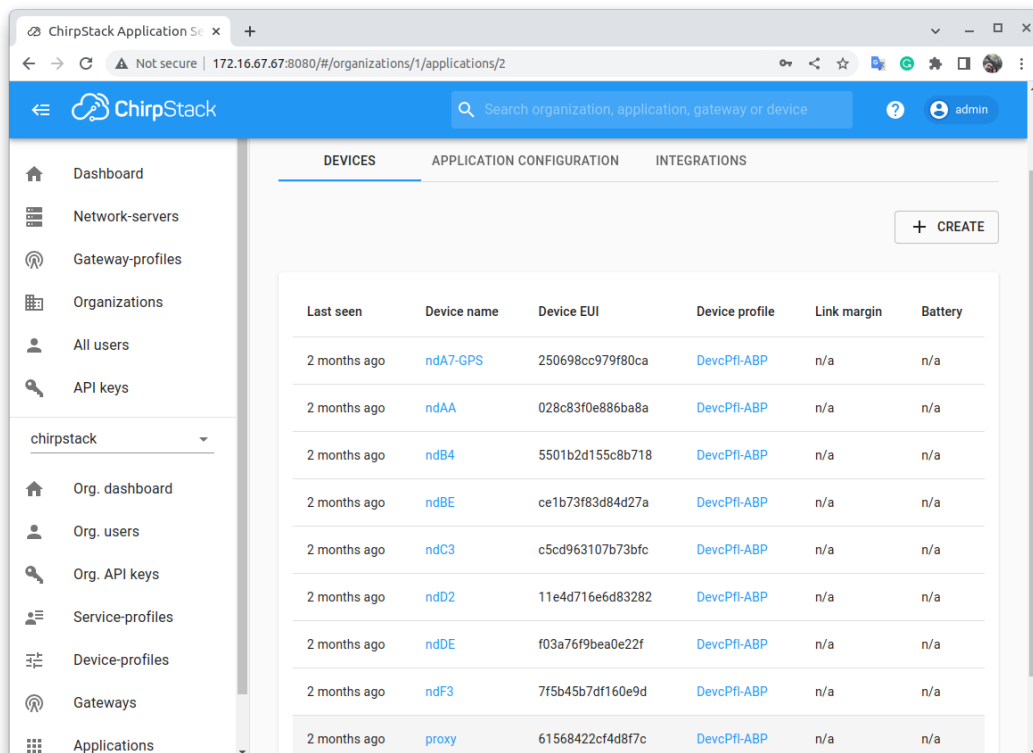
De forma análoga, é indicado a criação de um dispositivo *LoRaWAN* com o "Device name" *proxy* que será utilizado para verificação de seu funcionamento, isto é opcional, mas se mostra interessante para a verificação do funcionamento da rede pois, sabendo-se que o *Proxy* se encontra em funcionamento é possível verificar alguns dados juntamente com o nível de bateria dele. Na figura 36, observa-se o dispositivo "ndB4", o *proxy* e os outros dispositivos cadastrados que irão operar na malha, para o teste de funcionamento, foram cadastrados outros dispositivos de forma semelhante à exemplificada anteriormente.

Figura 35 – Cadastro das credenciais LoRaWAN no *ChirpStack*.



Fonte: Autor.

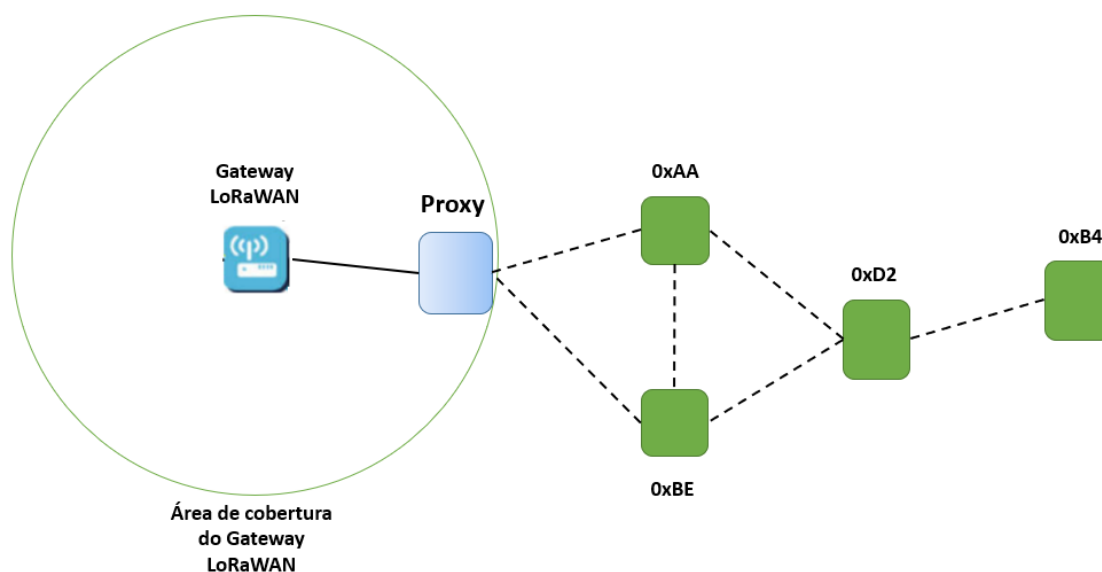
Figura 36 – Dispositivos cadastrados no *ChirpStack* para o teste da rede em malha.



Fonte: Autor.

Com vários dispositivos cadastrados e a rede em malha trabalhando, um cenário proposto é a ligação de alguns dispositivos e o *Proxy*, como mostra a Figura 37, onde o *Proxy* sendo instalado em uma área de cobertura *LoRaWAN* válida e os outros dispositivos posicionados em áreas de sombra, a rede em malha resultante permite que dispositivos até três "saltos de distância" se comuniquem com a rede *LoRaWAN*, conforme pode ser visto na próxima seção onde se apresentam os dados dos testes realizados, sendo este o cenário utilizado.

Figura 37 – Dispositivo *Proxy* interligando a rede *LoRa* em malha a um *gateway LoRaWAN*.



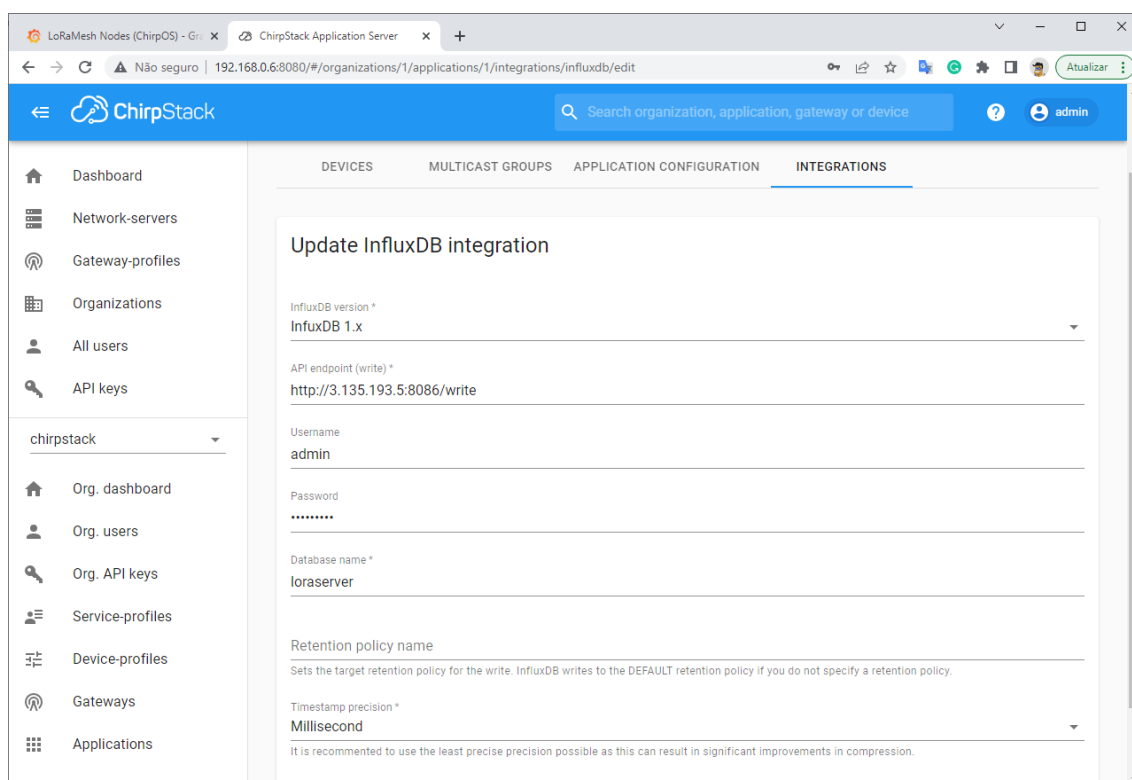
Fonte: Autor.

5.5 Aplicação para visualização dos dados da Rede em Malha

Para a utilização dos dados, provenientes de qualquer dispositivo *LoRaWAN* e para demonstração do uso dos dados da rede em malha, propõe-se a criação desta Aplicação que apresenta os dados. O termo Aplicação é um termo genérico que pode englobar várias soluções e indica qual ferramenta o usuário ou cliente irá utilizar para "consumir" os dados desta Aplicação. Para a presente Aplicação da rede em malha, será utilizado o Banco de Dados *InfluxDB*, aliado a um *script* em *Python* para captar e organizar os dados, já para a visualização será utilizado o *framework Grafana*, que possibilita o acesso e visualização de forma completa dos dados em um único ambiente, todas estas ferramentas estão alocadas em uma instância *VPS (Virtual Private Server)* no provedor da *Amazon Web Services (AWS)*. Os códigos para esta Aplicação estão abertos em um repositório no *Github* para esta demonstração (ALMEIDA, 2022).

Através do recurso de Integração, presente no *ChirpStack*, é possível enviar os dados de uma Aplicação *LoRaWAN* para um Banco de Dados e, na Figura 38, é apresentada esta configuração que pode ser acessado no *ChirpStack* no menu *Application* e aba *Integration*, acessar o tópico *InfluxDB* onde, os parâmetros configurados são: *API endpoint*, que indica o endereço da instância do Banco de Dados, *Username* e *Password* que são as credenciais para acessar o Banco de Dados, *Database name* que é o nome da tabela utilizado e *Timestamp precision* que foi setado para *Millisecond* (milissegundo).

Figura 38 – Configuração da Integração do *LoRaWAN* ao Banco de Dados *InfluxDB*.



Fonte: Autor.

Os pacotes de dados, que são recebidos pela Aplicação *LoRaWAN*, necessitam de um decodificador dentro do *ChirpStack*, este pode ser acessado pelo menu *Device-profiles* (perfil do dispositivo), onde é criado um perfil para os dispositivos da Aplicação e permite a edição do *Codec* de decodificação que tem a função de acessar o pacote, extrair os dados e enviar para o usuário ou cliente que irá consumir/utilizar os dados da Aplicação, neste caso, dos dispositivos da malha. Na Figura 39, observa-se um trecho deste código que foi editado para extrair os dados para serem enviados ao Banco de Dados *InfluxDB*. Nesta configuração a função `Decode(fPort, bytes)` realiza as tarefas de extrair os dados do pacote recebido, em formato *Json* (*JavaScript Object Notation* - Notação de Objetos em *JavaScript*), esta função separa os dados relevantes para o projeto e os prepara para serem gravados no Banco de Dados, em um formato *Json*, próprio da Aplicação.

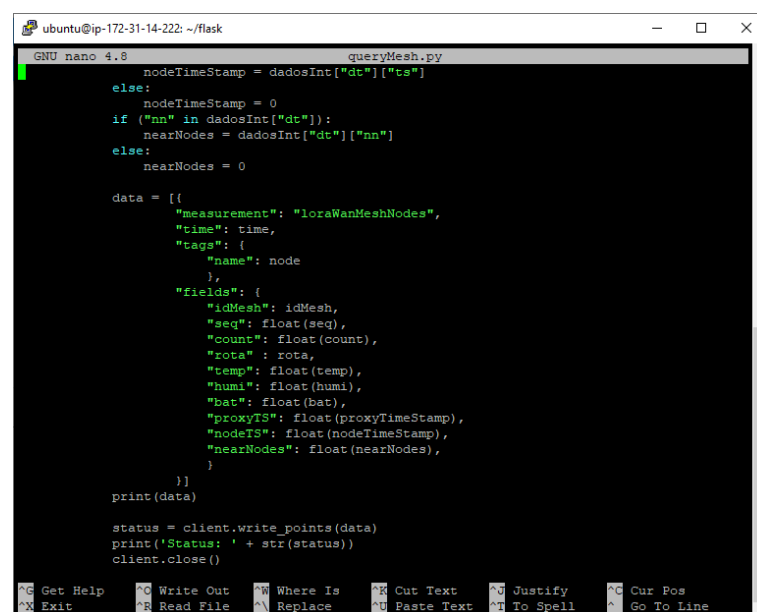
Figura 39 – Configuração da Integração do *LoRaWAN* ao Banco de Dados *InfluxDB* e indicação da função *Decode*.



Fonte: Autor.

Concluída esta parte, é necessário ter instalado no *VPS* o Banco de Dados *InfluxDB*, o *framework Grafana* e o *Python*. Um *script*, em *Python*, irá fazer acesso ao Banco de Dados, o tratamento e extração dos mesmos e o envio para a tabela que guarda os dados para apresentação no visualizador *Grafana*. Na figura 40, observa-se o trecho que tem a função de enviar os dados para a tabela final de dados.

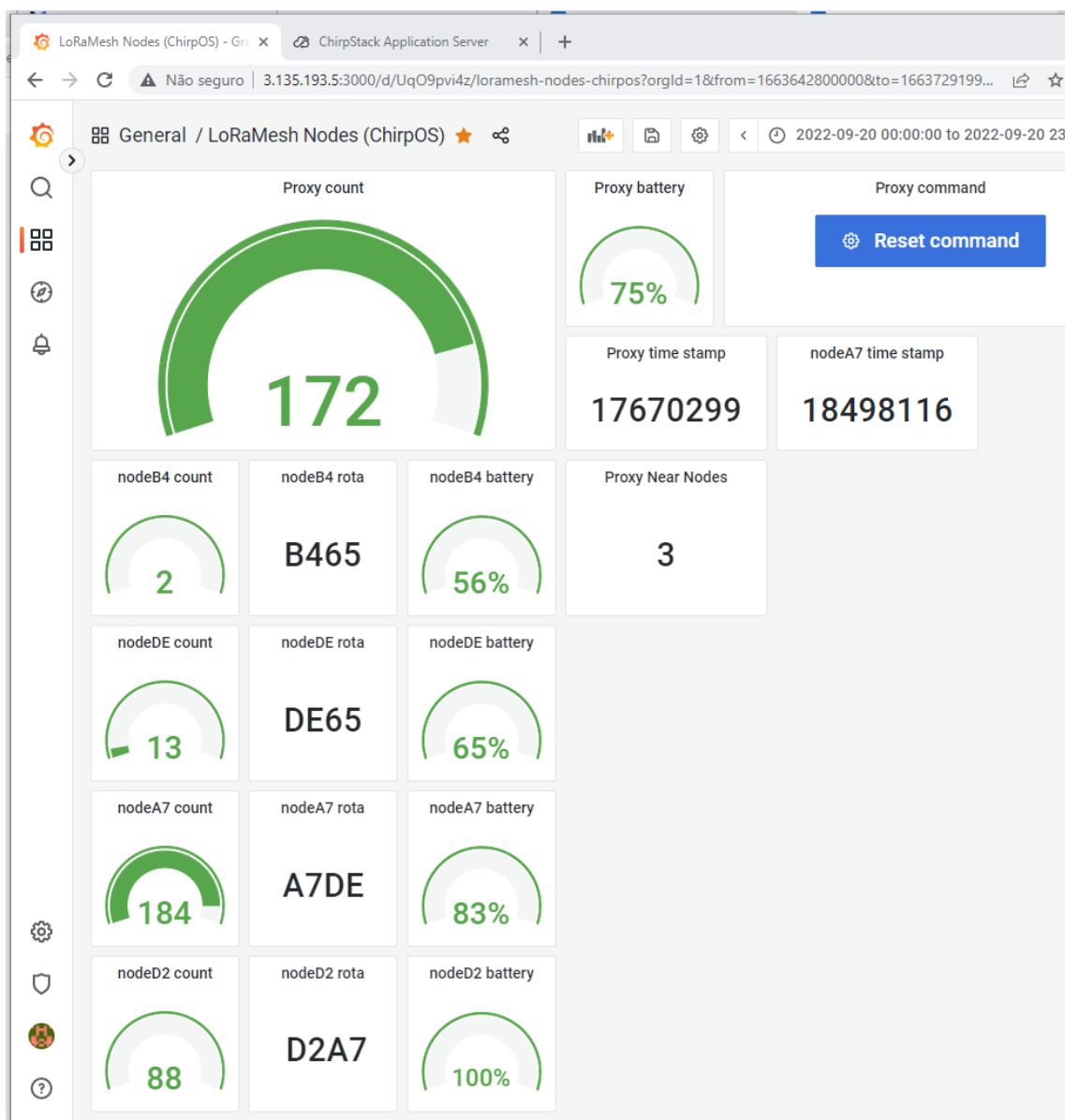
Figura 40 – *Script* em *Python* que envia os dados ao Banco de Dados *InfluxDB*.



Fonte: Autor.

Para a visualização dos dados, no *framework Grafana*, foi configurada uma *query* (consulta) individual para cada dispositivo, de forma a permitir o acesso aos dados, ainda, foi utilizado o formato de visualização em medidor para a contagem de pacotes recebidos e o nível de bateria dos dispositivos da malha, inclusive do *Proxy*, e o mostrador de texto para a rota dos dispositivos. Na Figura 41, observam-se os dados apresentados na Aplicação de Visualização no *Grafana*.

Figura 41 – Visualização dos dados apresentado pelo *Grafana*.

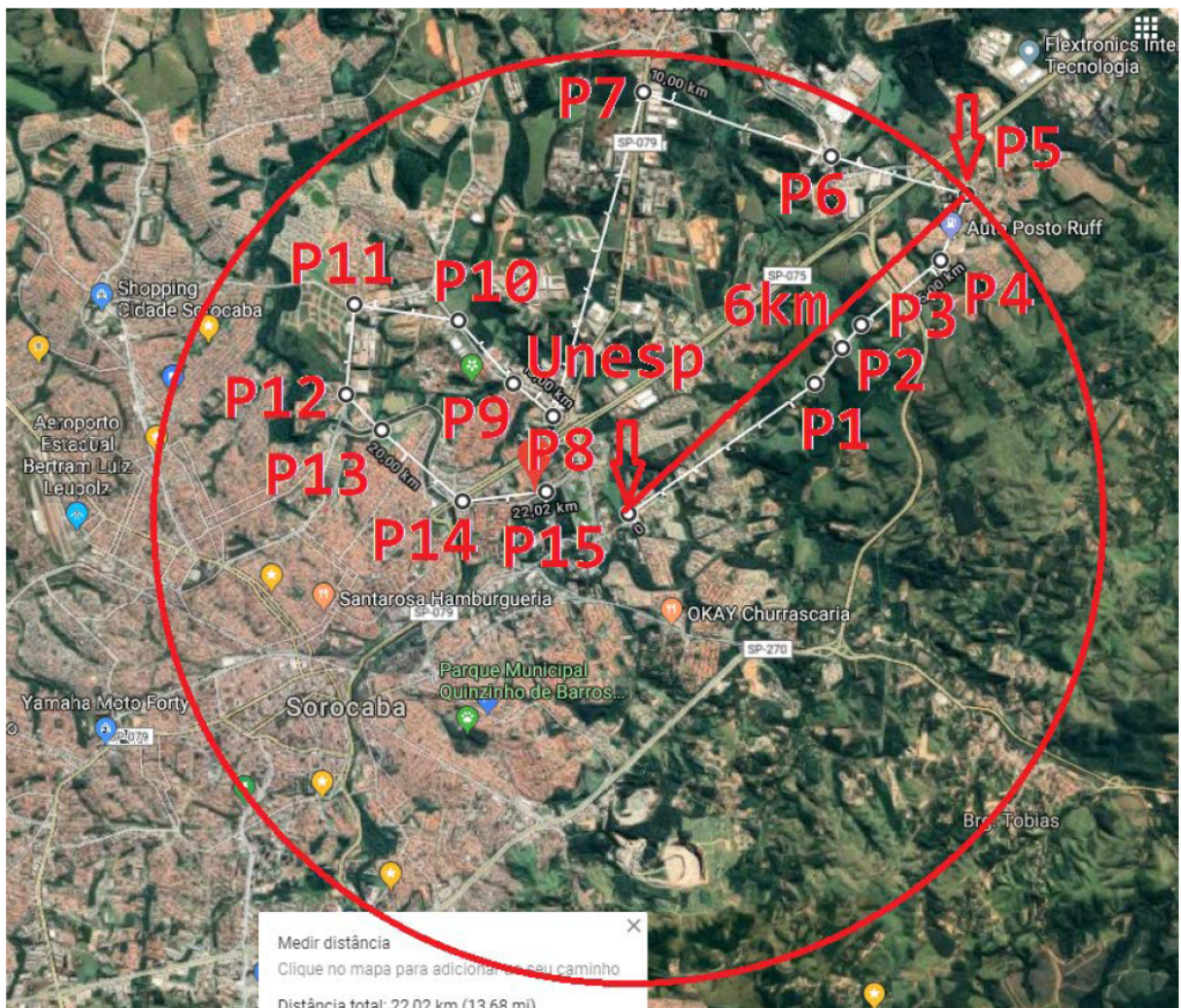


Fonte: Autor.

6 Resultados

Em testes preliminares, discutidos no artigo (ALMEIDA; ABUD; GODOY, 2019), ficou conhecido que, para o *gateway* LoRaWAN presente na Unesp Campus Sorocaba, existem áreas de sombra para a rede LoRaWAN onde, foi verificado através de um dispositivo *mapper* os pontos em que há a comunicação com o *gateway* e outros onde a comunicação não ocorre, devido a fatores como a topografia e obstáculos para a referida comunicação. Na Figura 42, observa-se que entre os pontos P3 e P4, não é possível para o dispositivo *mapper* se comunicar com a rede LoRaWAN, constatando que esta é uma área de sombra.

Figura 42 – Teste de cobertura da rede LoRaWAN e o posicionamento do *Proxy*.



Fonte: Autor.

Aumentando a imagem entre os pontos P3 e P4, pode-se ver que entre estes pontos existe uma grande área em que não há a cobertura *LoRaWAN* e, nesta área então, que serão realizados os testes. Na Figura 43, observa-se o ponto ideal para a ligação do *Proxy* é o ponto P3 pois, está mais próximo do *gateway* da Unesp a uma distância de 4,25km e permite a cobertura desta área de sombra com a rede em malha.

Figura 43 – Posicionamento do *Proxy* para teste da Rede Híbrida.



Fonte: Autor.

Para a validação da Rede Híbrida *LoRa* em Malha / *LoRaWAN*, os dispositivos da rede em malha foram alocados nesta área de sombra, sendo que, esta área se encontra em um vale e o cenário para os testes foi o seguinte: o *Proxy* foi posicionado em um ponto alto, P3 como referido anteriormente onde, inicialmente, as transmissões *LoRaWAN* ocorrem normalmente. Na Figura 44, é apresentado em (a) marcado em verde, a área onde tem-se cobertura *LoRaWAN*, em (b) marcado em cinza, a área de sombra para a rede *LoRaWAN* e o *Proxy* posicionado e indicado pelo círculo vermelho.

Figura 44 – Áreas de cobertura (a), sombra (b) para a rede *LoRaWAN* e o posicionamento do *Proxy*.



Fonte: Autor.

Para verificar a área de cobertura *LoRaWAN*, pode-se utilizar o aplicativo *web TTNMapper* (<https://ttnmapper.org/heatmap/>), que mostra graficamente em um mapa onde é possível, para os dispositivos *LoRaWAN*, atuarem e nas áreas onde não tem comunicação, caracteriza-se como uma área de sombra. Para este teste, se utiliza um dispositivo *LoRaWAN* com *GPS* e um *firmware* específico denominado *TTNMapper* instalado.

Neste exemplo, é utilizando o Servidor de Aplicação *ChirpStack*, porem, outros servidores de Aplicação podem ser utilizados, como por exemplo, o *The Things Network*, a Rede *ATC LoRaWAN* entre outros que não são discutidos neste trabalho.

Os experimentos a serem realizados vão utilizar o cenário proposto na Figura 45, onde os dispositivos da rede em Malha estão posicionados na área de sombra *LoRaWAN* e os parâmetros para os testes são:

- taxa de entrega de pacotes (*Packet Delivery Ratio - PDR*) dos dispositivos da malha (PDR_{malha}): que é calculado como sendo a relação entre os pacotes enviados pelos dispositivos e o número de pacotes recebidos na rede *LoRaWAN*, dado em porcentagem;
- taxa de entrega de pacotes *PDR* da Rede Híbrida *LoRaWAN* (PDR_{LW}): dado pela relação entre o número total de pacotes enviados pelos dispositivos e o número de total de pacotes *LoRaWAN* recebidos;
- média de tempo entre o dispositivo mais distante da malha e o *Proxy* (*End-to-end delay*): dado pela média de tempo de transmissão do último dispositivo até o *Proxy*, passando pela rede em Malha.

Um fator importante a ser observado, para o efetivo teste da rede em malha, é a distância entre os dispositivos, visto que, as características da rede *LoRa* dizem respeito a comunicação de longa distância o que interfere na formação da rede em malha, devido aos dispositivos encontrarem vizinhos mesmo em situações mais difíceis e, para a validação do teste da rede em malha necessita impor obstáculos para direcionar a criação dessa rede para a validação. Nos testes, observou-se que uma área de vegetação densa pode atrapalhar a comunicação *LoRa* mesmo a poucas centenas de metros e, áreas urbanas também reduzem a distância entre os dispositivos, o que pode ser contornado elevando a antena nos pontos verificados.

Com relação ao *SF*, foi utilizado *SF7* nos dispositivos da malha, no sentido de reduzir a distância entre eles e utilizar a criação de rotas usando a característica de menor distância de comunicação a favor do teste, sendo que, se outro *SF* maior for utilizado a criação de rotas aliado a uma maior distância de alcance da rede *LoRa* faz com que seja

criado uma rede com apenas um salto entre os dispositivos e o *Proxy*, isto é, todos os dispositivos conseguem se comunicar com o *Proxy* no cenário proposto. Sendo assim, a utilização de *SF7* proporcionando um bom ganho na área de alcance do cenário proposto.

Figura 45 – Posicionamento dos dispositivos para o teste da rede *LoRa* em malha.



Fonte: Autor.

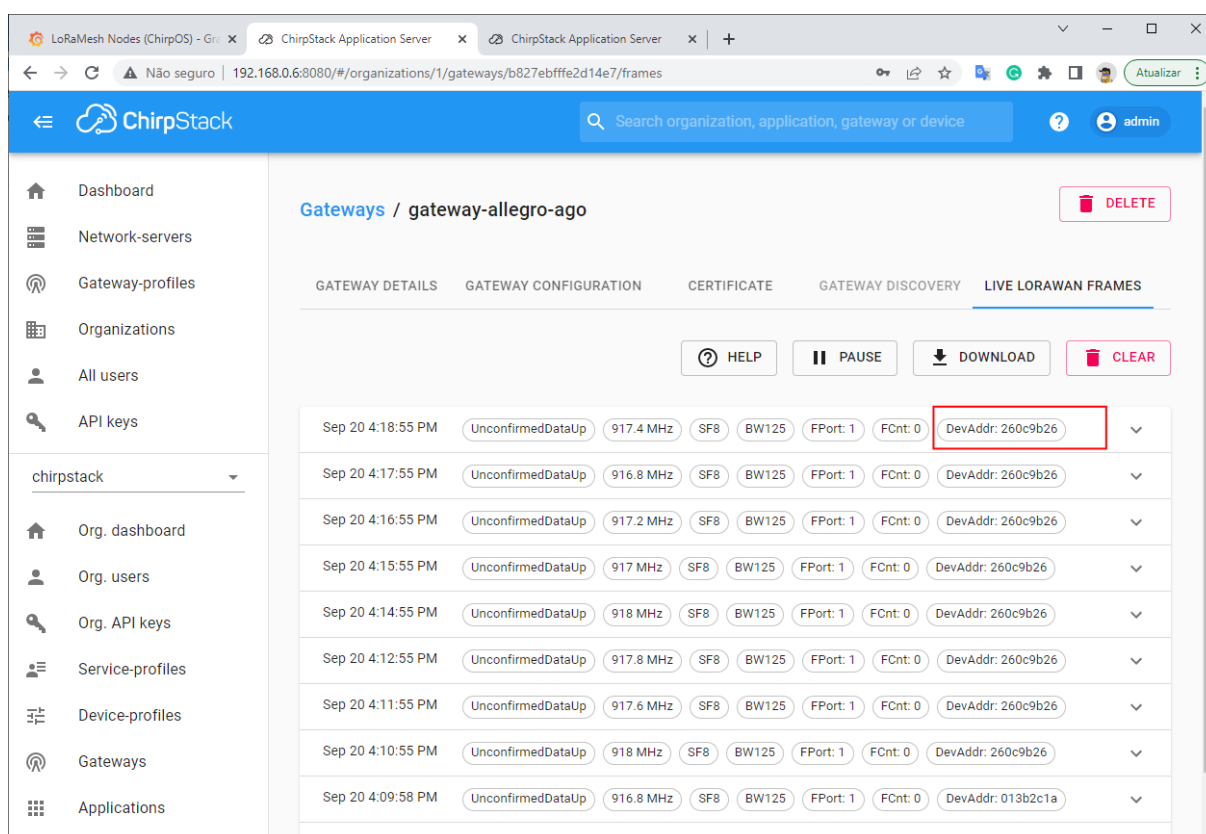
Com estes parâmetros será possível verificar o efetivo funcionamento e rendimento da Rede Híbrida *LoRa* em Malha / *LoRaWAN*, validando assim os processos de comunicação utilizados.

6.1 Verificação da comunicação *LoRaWAN* utilizando o *ChirpStack*

Para confirmar a comunicação de um determinado dispositivo *LoRaWAN*, tendo este dispositivo devidamente configurado em um Servidor de Aplicação, como por exemplo

o *ChirpStack*, deve-se observar a aba "LIVE LORAWAN FRAMES" (Pacotes *LoRaWAN* ao vivo em tradução direta) onde, é possível observar os pacotes recebidos por este *gateway*. Esta aba pode ser acessada através do menu *Gateways*, na *interface* do *ChirpStack*, o que pode ser visto na Figura 46, como um exemplo desta comunicação onde se observa um dispositivo identificado pelo endereço "DevAddr: 260c9b26" destacado.

Figura 46 – Visualização dos pacotes *LoRaWAN* dentro da *interface* do *ChirpStack* destacando em vermelho o *DevAddr* recebido.



Fonte: Autor.

Neste caso, o Servidor de Rede está recebendo os dados deste dispositivo, contudo, podem existir posições em campo que fazem com que esta comunicação não aconteça, apontando assim que temos nestes locais ou áreas as sombras para a comunicação *LoRaWAN* em que, os pacotes de dados dos dispositivos não sejam recebidos pelo *gateway* próximo. Nestes pontos é que se indica instalar um dispositivo *LoRa* em Malha. A área de cobertura do *gateway* instalado no Campus da Unesp foi discutida em (ALMEIDA; ABUD; GODOY, 2019), onde são identificados pontos ou áreas em que a comunicação dos dispositivos com o *gateway* ocorre normalmente, mas pontos de sombra aparecem entre os pontos verificados no estudo, não sendo possível a comunicação nestes locais devido a topologia da região verificada ou outras possíveis circunstâncias.

6.2 Testes de Comunicação e Validação da Rede *LoRa* em Malha

Para verificação do funcionamento da rede em malha, os dispositivos foram posicionados em uma área de sombra para o *gateway* que está no Campus da Unesp. Os dispositivos *LoRa* estão organizados, segundo o cenário proposto, da seguinte forma: utilizando 5 (cinco) dispositivos e o *Proxy* onde cada um recebeu um valor hexadecimal para sua identificação na malha e um nome, são eles: 0xB4 (ndB4), 0xBE (ndBE), 0xDE (ndDE), 0xA7 (ndA7) e 0xD2 (ndD2), o *Proxy* possui o endereço 0x65 na rede em malha. Vale ressaltar que, o gerenciamento da rede em malha permite diversas outras possibilidades de posicionamento dos dispositivos, formado configurações diferentes da apresentada, respeitando-se o alcance de comunicação dos dispositivos *LoRa* e a presença de obstáculos entre eles.

Para verificar os tempos da rede *LoRa* em Malha, observe a Figura 47, onde são mostrados recortes do terminal serial conectado ao *Proxy*, a fim de demonstrar o processo de descoberta de novos vizinhos e a formação da rede com dois dispositivos próximos.

Ligando os dispositivos, inicia-se o processo de descoberta de novos vizinhos (a) e de criação da rede em malha (b) e (c) com dois vizinhos que, pode variar dependendo do número de dispositivos próximos e, para visualizar esta função, observe o tempo utilizado pelo *Proxy* (iniciado às 15:50:03) o processo de descoberta de novos vizinhos, logo após, *RREQ/RREP* até chegar no passo identificado como "Valid route: True", com rota válida (terminado em 15:50:48), destacado em (d), com dois vizinhos descobertos, este tempo foi de aproximadamente 45 segundos tendo em vista que o processo de *RREQ/RREP* foi feito inicialmente para o dispositivo ndB4 e em seguida ndDE, sendo que os dispositivos foram ligados em tempos diferentes e nesta sequência. E por fim, em (e) pode-se observar o custo das rotas criadas para os dois vizinhos do exemplo.

Figura 47 – Demonstração do tempo utilizado pelo *Proxy* para gerenciar a criação de uma rede com 2 vizinhos. (a) início do processo de criação da rede, (b) descoberta do primeiro vizinho: 0xB4, (c) descoberta do segundo vizinho: 0xDE, (d) final da descoberta de vizinhos e rotas válidas e (e) custo das rotas criadas.

```

/dev/ttyUSB0
15:50:02.585 -> LoRaMesh Unesp Proxy V0.9.2
15:50:02.618 -> Starting LoRa...
15:50:03.117 -> LoRa init succeeded.
15:50:03.615 -> ESP32 Chip ID = 8485FFBF713C
15:50:03.615 -> Node: 0x65
15:50:03.649 -> Loading AU915/AU921 Configuration...
15:50:03.649 -> Setup OK, discover neighbourhoods... (a)
15:50:03.649 -> Upd queue(1): 1[65,105]
15:50:03.649 -> Status STPDISCOVER
15:50:08.698 -> Status STPREPREQ
15:50:08.698 -> Start... Sending RREQ message for 1 nodes!
15:50:08.732 -> -- 0x65 nodes --
15:50:08.732 -> #1: 0xB4 hp 1 Rx -49 NP 1 (b)
15:50:08.732 -> -- Tabela de RREQs para 1 nodes --
15:50:08.732 -> -- Send RREQ #1 -> 0xB4 hp 1 sa 54 sts 0
15:50:38.766 -> Status STPREPREQ
15:50:38.766 -> Start... Sending RREQ message for 2 nodes!
15:50:38.766 -> -- 0x65 nodes --
15:50:38.766 -> #1: 0xB4 hp 1 Rx -53 NP 1 (c)
15:50:38.799 -> #2: 0xDE hp 1 Rx -45 NP 1
15:50:38.799 -> -- Tabela de RREQs para 2 nodes --
15:50:38.799 -> -- Send RREQ #1 -> 0xB4 hp 1 sq 168 sts 0
15:50:38.799 -> SendRREQ->0xB4 rota(1): 65
15:50:38.799 -> RREQ sent, waiting 1 RREPs
15:50:40.693 -> SendRREQ->0xDE rota(1): 65
15:50:40.693 -> RREQ sent, waiting 1 RREPs
15:50:48.109 -> Rx RREP destin me!
15:50:48.109 -> Upd rt-> 0xDE with RX: -51 hp 1 rt: DE->65
15:50:48.109 -> #2: 0xDE hp 1 Rx -45 NP 1
15:50:48.109 -> -- Finish RREQ #1 of 0xDE in 3s with seq: 20
15:50:48.142 -> Route 1: 0xB4 Rx: -49 Hp: 1 >
15:50:48.142 -> Route 2: 0xDE Rx: -51 Hp: 1 >
15:50:48.142 -> :: Node: 0x65 TS: A
15:50:48.142 -> :: Version firm : 0.9.2
15:50:48.142 -> :: Valid route : True (d)
15:50:48.142 -> :: Near nodes : 2 -> 0xB4, 0xDE
15:50:48.142 -> :: Node position : 0 -> Proxy
15:50:48.142 -> :: Status machine: 0x21 => STPREPREQ
15:50:48.142 -> Route cost 1: 49 (e)
15:50:48.142 -> Route cost 2: 51
15:50:48.142 -> Finding best route to 2 nodes.
15:50:48.175 -> Best route: 1
Autoscroll Show timestamp Newline 115200 baud Clear output

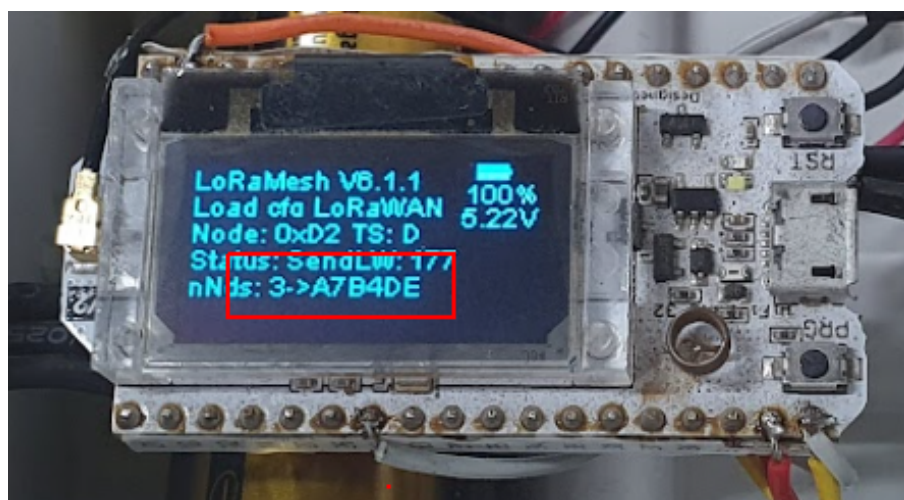
```

Fonte: Autor.

Após iniciados os outros dispositivos e, tendo a rede toda formada, puderam-se observar que os dispositivos ndB4, ndBE e ndDE fazem a transmissão dos pacotes para o *Proxy* com apenas 1 salto, o dispositivo ndD2 alcança o *Proxy* com 2 saltos, através dos dispositivos ndDE ou ndB4, configurando assim duas rotas possíveis para ndD2, o que pode ser verificado no *display* do dispositivo mostrado na Figura 48, já por sua vez, o dispositivo ndA7, consegue enviar seus dados através de ndD2 aproveitando as rotas deste

dispositivo e se comunicando com 3 saltos até o *Proxy*. Neste cenário, os pacotes de dados dos dispositivos chegam até o *Proxy* que está posicionado em um local onde é possível a comunicação com o *gateway LoRaWAN* do campus da Unesp onde, medindo pelo mapa, encontra-se a 4,25km de distância em linha reta, sendo este o único *gateway LoRaWAN* disponível na localidade. Outro fator importante é que, o *Proxy* está localização em uma região alta e a área onde estão dispostos os dispositivos é uma região baixa, criando assim esta região de sombra para a rede *LoRaWAN*.

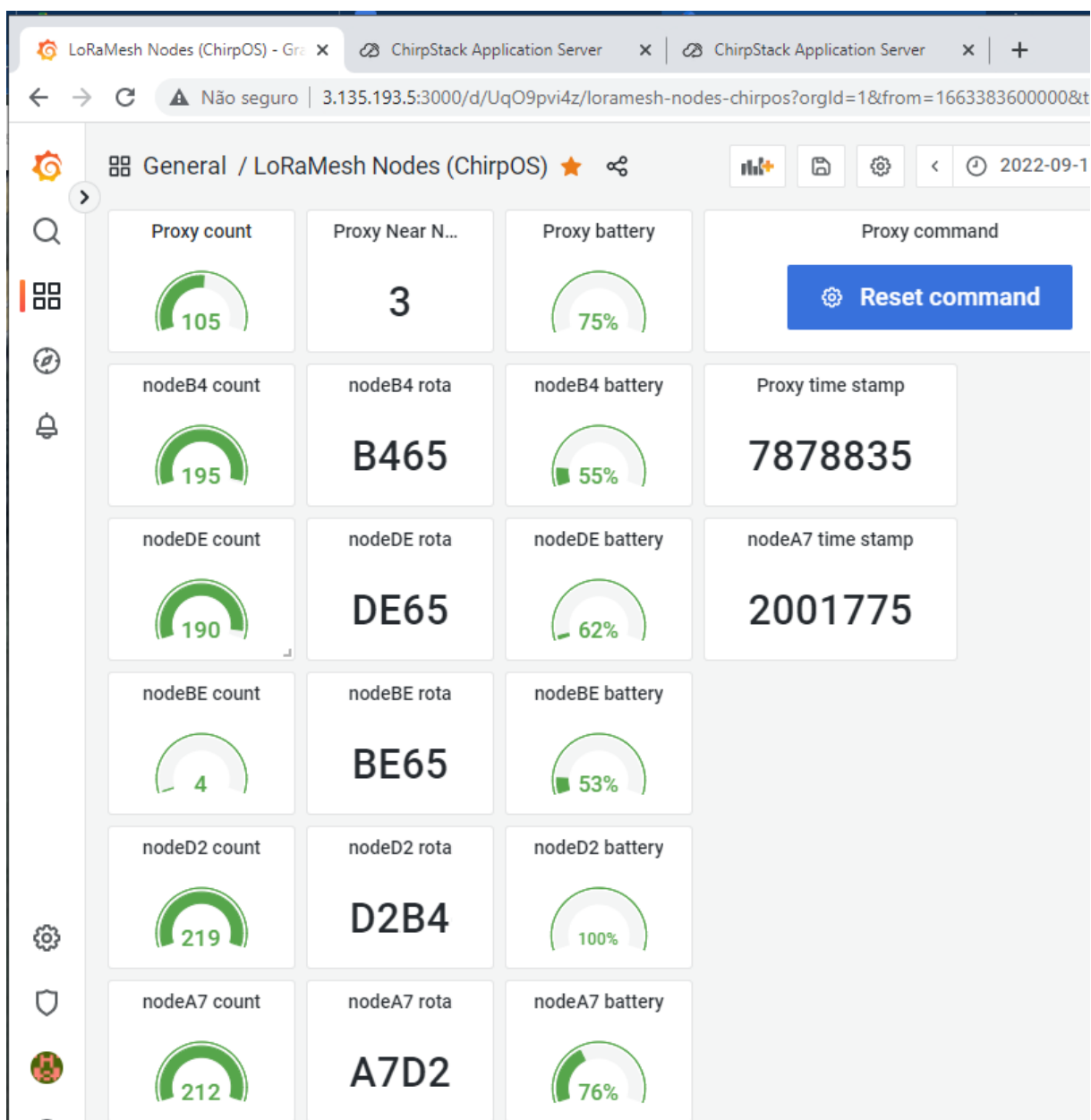
Figura 48 – No detalhe pode ser visto os dispositivos vizinhos do dispositivo ndD2.



Fonte: Autor.

Conhecidos estes passos, a Figura 49 apresenta o painel criado no *framework Grafana*, onde podem ser vistos os vários dados captados e destacam-se alguns detalhes:

- o *Proxy* apresentou 3 dispositivos próximos no campo "*Proxy Near Nodes*";
- a contagem de pacotes na primeira coluna indicada por "*nodeB4 count*", por exemplo;
- as rotas dos dispositivos, na coluna central, indicada por "*nodeB4 rota*", onde os dispositivos indicam o vizinho mais próximo ao qual está transmitindo seus pacotes;
- o valor da bateria dos dispositivos é indicado por "*nodeB4 battery*".

Figura 49 – Aplicação de teste da rede *LoRa* em Malha apresentada pelo *Grafana*.

Fonte: Autor.

6.3 Testes de Comunicação da Rede Híbrida *LoRa* em Malha / *LoRaWAN*

Com a rede *LoRa* em Malha em funcionamento, para o teste da Rede Híbrida, os testes de comunicação foram feitos da seguinte forma: após o processo de formação de rotas, os dados transmitidos pelos dispositivos da malha são arquivados no banco de dados *InfluxDB* e, através de uma consulta (*query*) ao banco de dados é possível acessar os dados para realizar a análise da rede *LoRa* em Malha e do funcionamento do *Proxy*. Para

esta análise, no pacote de dados, foi inserido um campo denominado *count* (contador) em formato numérico crescente que é reiniciado a cada inicialização dos dispositivos da malha e ajuda a entender quais pacotes foram recebidos pelo *Proxy* e quais foram reencaminhados para a rede *LoRaWAN*, desta forma é possível executar as seguintes análises: rendimento dos dispositivos, taxa de entrega de pacotes (*Packet Delivery Ratio - PDR*) e a média de tempo entre o dispositivo mais distante da malha e o *Proxy* (*End-to-end delay*).

6.3.1 Taxa de entrega de pacotes - *PDR* dos dispositivos da malha

A taxa de entrega de pacotes dos dispositivos da malha (PDR_{malha}), na Rede Híbrida, é calculado como a porcentagem de pacotes entregues pela rede e, será calculada pela relação entre o número de pacotes enviados pelos dispositivos e o número de pacotes recebidos no *Proxy* de forma individual para cada dispositivo.

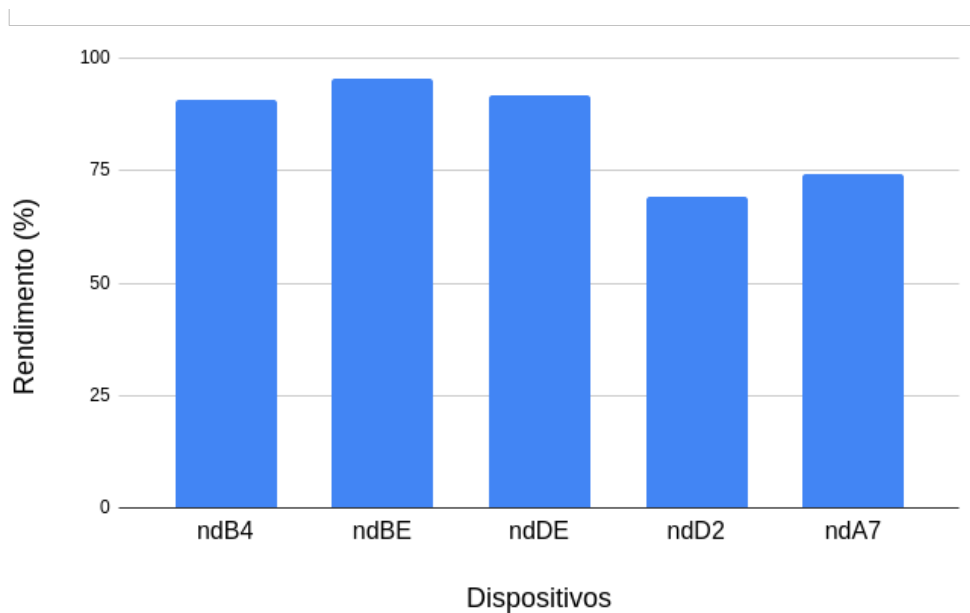
$$PDR_{malha} = \frac{\text{número_de_pacotes_enviados}}{\text{número_de_pacotes_recebidos}} * 100\% \quad (6.1)$$

Para esta análise, foi criado um *script* em *python*, no *VPS* que roda a Aplicação, este *script* acessa o banco de dados e faz a contagem dos pacotes recebidos onde é calculada a relação de rendimento utilizando o contador presente nos pacotes de cada dispositivo da malha. Os valores são apresentados na Tabela 2 e no gráfico da Figura 50, observa-se que os dispositivos mais próximos do *Proxy* (1 salto) apresentaram melhor rendimento, já os dispositivos mais distantes ndD2 (2 saltos) e ndA7 (3 saltos), começam a apresentar um rendimento menor onde, o dispositivo ndD2 foi quem apresentou o menor valor devido, possivelmente, a erros de transmissão e perdas de pacotes por checagem de *CRC*. Também existe uma perda maior quanto mais saltos são inseridos na rede em Malha como foi observado no artigo (SOLé et al., 2022), sendo esta uma característica importante das redes em malha.

Tabela 2 – Taxa de rendimento dos dispositivos da malha.

Dispositivos	número de pacotes enviados	número de pacotes recebidos LoRaWAN	PDR_{malha} (%)
ndB4	563	512	90,94
ndBE	575	550	95,65
ndDE	545	500	91,74
ndD2	542	376	69,37
ndA7	523	388	74,19

Fonte: Autor.

Figura 50 – Rendimento dos dispositivos da Rede *LoRa* em Malha.

Fonte: Autor.

O processo de contagem e verificação foi feito contabilizando-se os pacotes recebidos e descartando-se os pacotes faltantes, analisando o campo contador presente no pacote de dados dos dispositivos. É importante salientar que, os resultados são referentes a um dia de testes e mais testes são necessários, porém, o fato de os dispositivos serem alimentados por bateria limita a um teste por dia e então, aguardar o tempo de recarga das baterias para novos testes, isto limitou a geração de mais dados sobre os dispositivos visto que, estes estavam em campo e a distâncias relativamente grandes.

6.3.2 Taxa de entrega de pacotes - *PDR* da Rede Híbrida *LoRaWAN*

A taxa de entrega de pacotes da Rede Híbrida *LoRaWAN* (PDR_{LW}), é calculada obtendo-se a porcentagem do valor total de pacotes de todos os dispositivos da malha em relação ao número de pacotes recebidos pela Aplicação *LoRaWAN*, através do *Proxy* e, indica a quantidade total de pacotes que a rede está entregando. Na Tabela 3, observam-se os dados e o valor de *PDR* calculado.

Tabela 3 – Taxa de *PDR* da rede Híbrida *LoRa* em Malha / *LoRaWAN*.

Total de pacotes enviados	Total de pacotes recebidos <i>LoRaWAN</i>	PDR_{LW} (%)
2748	2326	84,64

Fonte: Autor.

7 CONCLUSÕES

Este trabalho apresentou e descreveu o desenvolvimento de uma Rede Híbrida *LoRa* em Malha / *LoRaWAN*, através do desenvolvimento do dispositivo *Proxy*, que gerencia a rede *LoRa* em Malha e atua como ponte entre a rede em malha e a rede *LoRaWAN* formando, assim, a Rede Híbrida *LoRaWAN*. Este dispositivo *Proxy* é compatível com as especificações *MAC* versão 1.0.3 da rede *LoRaWAN* em Classe C e permite futuras atualizações. O desenvolvimento desta solução se deve ao fato de que a rede *LoRaWAN*, através dos *gateways*, que fazem a comunicação dos dispositivos *LoRaWAN* com Servidores de Rede e de Aplicação, apresenta um bom alcance nas transmissões de dados, porém, fatores externos, relacionados ao local onde estes *gateways* são instalados e a presença de obstáculos para a transmissão dos pacotes *LoRaWAN*, podem fazer com que apareçam áreas com sombra, para os pacotes transmitidos o que faz com que estes locais de sombra não sejam cobertos pela rede *LoRaWAN*.

O presente desenvolvimento do dispositivo *Proxy*, proporciona a interconexão da Rede *LoRa* em Malha à rede *LoRaWAN*, sendo que, não é necessária nenhuma alteração na infraestrutura já existente de *gateways LoRaWAN*, devido ao fato de que o dispositivo *Proxy* é totalmente compatível com os *gateways* já disponíveis virtualizando os dispositivos da malha para a rede *LoRaWAN* utilizando o método de ativação *ABP*, ainda apenas para envio (*uplink*) de pacotes, sendo o *downlink* da rede *LoRaWAN* não implementado. É indicado para futuras versões deste projeto, a implementação do método de ativação *OTAA*, para melhor segurança das aplicações e da função de *downlink* na rede Híbrida *LoRaWAN*.

Sendo a rede *LoRaWAN*, uma das *LPWANs* que mais crescem no cenário mundial, vários trabalhos recentes apontam possíveis soluções para as áreas de sombra tanto em zonas rurais, no setor industrial, doméstico ou híbridos. Neste caso, a utilização de uma rede Híbrida *LoRa* em Malha / *LoRaWAN* permite que, locais onde não possa ocorrer a transmissão *LoRaWAN*, mesmo dentro da área de cobertura de um *gateway LoRaWAN*, um dispositivo ligado em malha possa transmitir e receber dados até uma aplicação utilizando a rede em malha aqui apresentada e, enviar os dados para a rede *LoRaWAN* através do referido *Proxy*, sendo este posicionado dentro da área de cobertura de um *gateway LoRaWAN* devidamente operante.

A rede em malha permite, através do seu gerenciamento de rede, que se algum dispositivo venha a desligar, outra rota seja utilizada, adicionando assim, um fator de confiabilidade na transmissão dos pacotes de dados, porém, uma confirmação de envio/recebimento se faz necessário, mas poderia causar um atraso na rede ou maior dificuldade de

gerenciamento das transmissões, tanto na criação da rede, quanto no envio dos dados com a rede em funcionamento. Esta rede em malha desenvolvida não apresenta confirmação de transmissão de pacotes de dados.

Quanto a utilização de memória: um fator relevante na seleção do microcontrolador utilizado foi: o microcontrolador *ESP32* possui 520KB de memória *RAM*, o que facilitou o desenvolvimento e gerenciamento das filas de transmissão e tabelas de rotas. A quantidade de memória utilizada supera a disponibilidade de memória de outros microcontroladores de mercado atuais. Outro fator importante é que, a plataforma *ESP LoRa 32 WiFi V2* apresenta *hardware* de baixo custo e com todas as características essenciais para este projeto.

O consumo de energia não foi abordado nos dispositivos da malha, mas observa-se que para manter a rede em malha operando é interessante que o microcontrolador esteja funcionando integralmente, não possibilitando que este entre em modos de economia de energia, sendo classificado como de Classe C, pelo fato de enviar dados de forma assíncrona, esteja sempre em modo de recepção para os pacotes de dados de um dispositivo vizinho, assim sendo, deve por estar sempre em modo de interrupção por recepção, para reencaminhar os pacotes recebidos de um vizinho para o *Proxy*. Uma possibilidade seria, o gerenciamento de tempos de transmissão agendados, onde os dispositivos acionem em intervalos de tempo pré-definidos e retornem ao estado de economia de energia.

Uma situação constatada é que, se os dispositivos estiverem próximos, a formação de redes com 1 salto é mais constante, o que para os testes foi fator determinante tendo em vista que a distância entre os dispositivos necessitou ser bem estudada, já que a rede *LoRa* é conhecida pela transmissão em longas distâncias, desta forma optou se por utilizar *SF7* que possui o menor alcance para a realização dos testes. Uma possível solução para os testes é a limitação do valor de *RSSI*, onde se verificou que, para valores menores que -102dBm, a quantidade de pacotes perdidos aumenta consideravelmente, inviabilizando o teste da rede em malha, já que com poucos saltos a comunicação em malha não é validada.

Quanto às características do protocolo *AODV*, tais características facilitaram o desenvolvimento do presente projeto, mas se torna interessante o teste de novas variantes deste mesmo protocolo, pois podem adicionar novas funcionalidades ao gerenciamento de rotas de formas mais eficientes em relação à rede em malha.

Os dispositivos instalados na área de sombra, demonstram que a solução apresentada é uma boa opção para sensoriamento remoto, cidades inteligentes e controles autômatos, facilitados pela instalação da Rede Híbrida *LoRa* em Malha / *LoRaWAN*.

Outra possibilidade que se apresenta é, a extensão da rede *LoRaWAN*, sendo que os *links* da rede *LoRa* em Malha podem chegar a vários quilômetros de distância e apresentam uma solução com um rendimento superior a 80%, aproveitando este bom indicador, porém,

não sendo indicada para situações críticas de comunicação.

Atendendo aos itens propostos, este projeto demonstrou a utilização da Rede em Malha, utilizando o protocolo *LoRa*, comunicando com a rede *LoRaWAN*, sendo esta uma rede bastante importante em aplicações de *IoT*. A Rede em Malha, aliada ao longo alcance da comunicação *LoRa*, permitem uma área de cobertura muito grande sendo uma excelente opção às redes mais tradicionais, tais como, *Wi-Fi*, *Bluetooth* ou *3G/4G*. Outros pontos favoráveis são: os fabricantes disponibilizam bibliotecas para programação, e ainda, a existência de uma comunidade ativa de desenvolvedores que oferecem um bom suporte a atualizações de *software* para vários sistemas de microcontroladores do mercado.

Os resultados apresentados validam o desenvolvimento do presente projeto, aqui apresentado, como uma possível solução para a problemática da comunicação em áreas de sombra e a extensão do alcance da rede *LoRaWAN*. O dispositivo *Proxy*, principal elemento da rede Híbrida *LoRa* em Malha / *LoRaWAN*, atuando como gerenciador da Rede em Malha, constitui um elemento que permite que mais dispositivos se conectem à rede *LoRaWAN*, estendendo seu alcance e melhorando a conectividade em áreas de sombra.

7.1 Proposta de melhorias

A seguir, apresentam-se algumas propostas de melhorias verificadas:

- melhorar o gerenciamento da rede em malha;
- adicionar suporte a ativação *OTAA*;
- adicionar suporte a mensagens de *downlink* para os dispositivos da malha através do *Proxy*;
- criptografia entre os dispositivos da malha;
- contador de pacotes salvo em memória *flash*;
- análise de consumo de energia e melhora na duração da bateria;
- possibilidade de limitar o *RSSI* na descoberta de novos vizinhos, a fim de otimizar a rede em malha.

Referências

- ABRARDO, A. et al. Black Powder Flow Monitoring in Pipelines by Means of Multi-Hop LoRa Networks. In: *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)*. [s.n.], 2019. p. 312–316. Disponível em: <<https://doi.org/10.1109/METROI4.2019.8792890>>.
- AKYILDIZ, I. F.; WANG, X. A survey on Wireless Mesh Networks. *IEEE Communications Magazine*, v. 43, n. 9, p. S23–S30, 2005. Disponível em: <<https://doi.org/10.1109/MCOM.2005.1509968>>.
- ALMEIDA, N. C.; ABUD, R. B.; GODOY, E. P. Infraestrutura LoRaWAN para Aplicações didáticas de Internet das Coisas. *4º Congresso Pós-Graduação do IFSP - 2019*, 11 2019.
- ALMEIDA, N. C. de. *Hybrid LoRa Mesh*. 2022. Acesso em 20 out. 2022. Disponível em: <<https://github.com/ncamilo/hybrid-lora-mesh>>.
- ALMEIDA N. C.; ROLLE, R. G. E. P. F. P. S. E. Proposal of a hybrid lora mesh / lorawan network. In: *2020 IEEE International Workshop on Metrology for Industry 4.0 IoT*. [S.l.: s.n.], 2020. p. 702–707.
- ANATEL. *ATRIBUIÇÃO DE FAIXAS DE FREQUÊNCIAS NO BRASIL*. [S.l.], 2014. Acesso em 22 jun. 2020. Disponível em: <<https://www.anatel.gov.br/Portal/verificaDocumentos/documentoVersionado.asp?numeroPublicacao=&documentoPath=radiofrequencia/qaff.pdf&Pub=&URL=/Portal/verificaDocumentos/documento.asp>>.
- ANGRISANI, L. et al. LoRa protocol performance assessment in critical noise conditions. In: *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*. [s.n.], 2017. p. 1–5. Disponível em: <<https://doi.org/10.1109/RTSI.2017.8065952>>.
- ASLAM, M. S. et al. Exploring Multi-Hop LoRa for Green Smart Cities. *IEEE Network*, v. 34, n. 2, p. 225–231, 2020. Disponível em: <<https://doi.org/10.1109/MNET.001.1900269>>.
- AZHARI, R. Y.; FIRMANSYAH, E.; BEJO, A. Simple Protocol Design of Multi-Hop Network in Lora. In: *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. [s.n.], 2019. p. 177–181. Disponível em: <<https://doi.org/10.1109/ISRITI48646.2019.9034662>>.
- CHIRPSTACK, open-source LoRaWAN®Network Server. 2022. Acesso em: 01 set. 2022. Disponível em: <<https://www.chirpstack.io/>>.
- DIAS, J.; GRILO, A. Multi-hop LoRaWAN uplink extension: specification and prototype implementation. In: *Journal of Ambient Intelligence and Humanized Computing*. [s.n.], 2020. p. 945–959. Disponível em: <<https://doi.org/10.1109/ISRITI48646.2019.9034662>>.
- DUONG, C. T.; KIM, M.-K. Reliable Multi-Hop Linear Network Based on LoRa. *International Journal of Control and Automation*, v. 11, p. 143–154, 04 2018. Disponível em: <<http://dx.doi.org/10.14257/ijca.2018.11.4.13>>.

- ESPRESSIF SYSTEMS. *Espressif ESP32 DevKitC*. 2022. Acesso em 20 set. 2022. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>>.
- G-NICERF®. *G-NiceRF® LoRa127X module*. 2022. Acesso em 20 set. 2022. Disponível em: <<https://www.nicerf.com/products/detail/sx1276-lora-module-lora1276-1.html>>.
- GEMALTO; ACTILITY; SEMTECH. *LoRaWAN Security, Full End-to-End Encryption for IoT Application Providers*. 2017. Acesso em: 01 set. 2020. Disponível em: <https://lora-alliance.org/sites/default/files/2019-05/lorawan_security_whitepaper.pdf>.
- LORA-ALLIANCE. 2020. Acesso em: 01 set. 2020. Disponível em: <<https://www.lora-alliance.org>>.
- LORA-ALLIANCE. *RP2-1.0.1 LoRaWAN® Regional Parameters*. [S.l.], 2020. Acesso em 22 jun. 2020. Disponível em: <https://lora-alliance.org/sites/default/files/2020-06/rp_2-1.0.1.pdf>.
- LORAWAN-NODE. 2019. Acesso em: 01 set. 2019. Disponível em: <<https://github.com/ncamilo/LoRaWAN-node>>.
- LUNDELL, D. et al. A routing protocol for lora mesh networks. In: *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. [s.n.], 2018. p. 14–19. Disponível em: <<https://doi.org/10.1007/s12652-019-01207-3>>.
- LUVISOTTO, M. et al. On The Use Of LoRaWAN For Indoor Industrial IoT Applications. *Wireless Communications and Mobile Computing*, Hindawi, 2018. Disponível em: <<https://doi.org/10.1155/2018/3982646>>.
- MEKKI, K. et al. A Comparative Study Of LPWAN Technologies For Large-Scale IoT Deployment. *Wireless Communications and Mobile Computing*, ICT Express. Elsevier BV., 2018. Acesso em 11 abr. 2020. Disponível em: <<https://doi.org/10.1016/j.ict.2017.12.005>>.
- RADIOENGE. *Gateway LoRaWAN RD43HAT*. 2020. Disponível em: <<https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html>>. Acesso em: 15 set. 2020.
- SANCHEZ-IBORRA, R.; CANO, M.-D. State of the art in LP-WAN solutions for industrial IoT services. In: _____. [s.n.], 2017. p. 241 – 254. ISBN 9783038423706. Disponível em: <<https://doi.org/10.3390/s16050708>>.
- SEMTECH. *LoRa Modulation Basics*. [S.l.], 2015. Acesso em 15 jun. 2020. Disponível em: <<http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf>>.
- SEMTECH CORPORATION. *LoRa Modulation LoRaWAN A Technical Overview*. [S.l.], 2019. Acesso em 20 jun. 2020. Disponível em: <https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf>.
- SINHA, R. S.; WEI, Y.; HWANG, S.-H. A survey on LPWAN technology: LoRa and NB-IoT. In: . [s.n.], 2017. v. 3, p. 14–21. Disponível em: <<https://doi.org/10.1016/j.ict.2017.03.004>>.

SISINNI, E. et al. Lorawan range extender for industrial iot. *IEEE Transactions on Industrial Informatics*, v. 16, n. 8, p. 5607–5616, 2019. Disponível em: <<https://doi.org/10.1109/TII.2019.2958620>>.

SOLÉ, J. M. et al. Implementation of a lora mesh library. *IEEE Access*, v. 10, p. 113158–113171, 2022.

TUDO CONECTADO. Revista da Sociedade Brasileira de Computação. Sociedade Brasileira de Computação, n. 4, 2015. Acesso em 10 abr. 2020. Disponível em: <https://www.sbc.org.br/images/flippingbook/computacaobrasil/computa_29_pdf/comp_brasil_2015_4.pdf>.

USMAN, R.; KULKARNI, P.; SOORIYABANDARA, M. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys Tutorials*, v. 19, n. 2, p. 855–873, 2017. Disponível em: <<https://doi.org/10.1109/COMST.2017.2652320>>.

VARGHESE, S. G. et al. Comparative study of zigBee topologies for IoT-based lighting automation. *IET Wireless Sensor Systems*, v. 9, n. 4, p. 201–207, 2019. Disponível em: <<https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-wss.2018.5065>>.

ZHANG, Y.; LUO, J.; HU, H. *Wireless Mesh Networking: Architectures, Protocols and Standards*. [S.l.]: CRC Press, 2006. (Wireless Networks and Mobile Communications). ISBN 9781420013542.