

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JOÃO PEDRO PASTÓRIO

**ALGORITMO DE GERENCIAMENTO DE RETRANSMISSÃO
EM NÍVEL DE APLICAÇÃO PARA REDES LORAWAN**

TOLEDO

2022

JOÃO PEDRO PASTÓRIO ✉

**ALGORITMO DE GERENCIAMENTO DE RETRANSMISSÃO
EM NÍVEL DE APLICAÇÃO PARA REDES LORAWAN**

**APPLICATION-LEVEL RETRANSMISSION MANAGEMENT
ALGORITHM FOR LORAWAN NETWORKS**

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Edson Tavares de Camargo ✉

Coorientador: Prof. Dr. Álvaro Ricieri Castro e Souza ✉

TOLEDO

2022



Este Trabalho de Conclusão de Curso está licenciado sob uma Licença Creative Commons Atribuição 4.0 Internacional.



TERMO DE APROVAÇÃO

ALGORITMO DE GERENCIAMENTO DE RETRANSMISSÃO EM NÍVEL DE APLICAÇÃO PARA REDES LORAWAN

por

João Pedro Pastório

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 14 de Junho de 2022 como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores listados na sequência. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Edson Tavares de Camargo
Presidente (Orientador)

Prof. Dr. Alexandre Huff
Membro Titular

Prof. Dr. Luis Carlos Mathias
Membro Titular

Prof. Dr. Daniel Cavalcanti Jeronymo
Responsável pelas Atividades de TCC

Prof. Dr. Maurício Zardo Oliveira
Coordenador(a) do Curso

Aos que conhecimento almejam, este
trabalho eu dedico.

AGRADECIMENTOS

Agradeço a minha família e amigos que sempre estiveram comigo e me apoiaram, mesmo nos momentos em que eu não pude estar presente. Em especial aos meus pais Walmir e Adriana, por me incentivar a sempre buscar conhecimento e sabedoria, mesmo nos momentos mais difíceis. E a minha irmã Gabriela por todo o carinho, me aconselhando durante toda a minha jornada.

A todos os meus professores, o conhecimento aqui expresso reflete o empenho e trabalho duro de vocês. Um agradecimento especial ao meu orientador Edson e coorientador Álvaro, por todos os ensinamentos compartilhados. Este trabalho não existiria sem vocês.

A última mensagem do golfinho foi mal interpretada [...], na verdade, a mensagem era esta: Até mais, e obrigado pelos peixes. (ADAMS, 1980, tradução)^a

^a “*The last ever dolphin message was misinterpreted [...], in fact the message was this: So long and thanks for all the fish.*” (ADAMS, 1980).

RESUMO

A Internet das Coisas (IoT) caracteriza objetos e dispositivos que apresentam a capacidade de se comunicar com a Internet. Dentre as tecnologias de rede sem fio utilizadas pelos dispositivos IoT, a modulação LoRa e o protocolo LoRaWAN ganham destaque em soluções urbanas de baixo consumo energético e ampla área de abrangência. Porém, nas redes LoRa, a troca de mensagens entre os dispositivos e o ponto de acesso costuma sofrer diversas interferências que prejudicam a qualidade do sinal e resultam na perda de dados. Como o protocolo LoRaWAN não apresenta nenhum mecanismo para garantir a entrega das mensagens, surge a proposta de um algoritmo para gerenciar as retransmissões das mensagens perdidas. Diferente de propostas anteriores, este trabalho propõe um algoritmo para operar em nível de aplicação através da comunicação bidirecional entre dispositivo e servidor de aplicação. O algoritmo é avaliado de forma prática em cenários reais de aplicações com o objetivo de verificar seu impacto ao garantir a entrega de dados. É realizada também uma comparação com o algoritmo CR-LoRaWAN, já desenvolvido para retransmissões em redes LoRa. Na comparação, evidenciam-se as principais diferenças entre os algoritmos e seus comportamentos mediante a diferentes testes. A partir dos resultados obtidos se evidencia a efetividade da abordagem adotada, revelando seu desempenho em diferentes execuções.

Palavras-chave: Internet das Coisas; redes LoRa; LoRaWAN; algoritmo de retransmissão.

ABSTRACT

The Internet of Things (IoT) characterizes objects and devices that have the ability to communicate with the Internet. Among the wireless network technologies used by IoT devices, LoRa modulation and the LoRaWAN protocol are highlighted in low-energy, wide-area urban solutions. However, in LoRa networks, the exchange of messages between the devices and the access point often suffers from various interferences that impair the signal quality and result in data loss. As the LoRaWAN protocol does not present any mechanism to guarantee the delivery of data, an algorithm is proposed to manage the retransmissions of lost messages. Different from previous proposals, this work proposes an algorithm to operate at the application level through bidirectional communication between device and application server. The algorithm is practically evaluated in real application scenarios in order to verify its impact when guaranteeing data delivery. A comparison is also made with the CR-LoRaWAN algorithm, already developed for retransmissions in LoRa networks. In the comparison, the main differences between the algorithms and their behavior through different tests are highlighted. From the results obtained, the effectiveness of the approach adopted is evidenced, revealing its performance in different executions.

Keywords: Internet of Things; LoRa networks; LoRaWAN; retransmission algorithm.

LISTA DE ILUSTRAÇÕES

Figura 1 – Tecnologias de rede	20
Figura 2 – Arquitetura de redes LoRaWAN	22
Figura 3 – Comunicação Classe A	23
Figura 4 – Placa de desenvolvimento - TTGO Tbeam	29
Figura 5 – Gateway RD43HAT Radioenge	30
Figura 6 – Diagrama de troca de mensagens com MQTT	31
Figura 7 – Fluxograma da troca de mensagens do dispositivo final	34
Figura 8 – Fluxograma do mecanismo de retransmissão	35
Figura 9 – Sequencia de recebimento de dados e retransmissões para N igual a 1	41
Figura 10 – Sequencia de recebimento de dados e retransmissões para N igual a 2	42
Figura 11 – Sequencia de recebimento de dados e retransmissões para N igual a 3	42
Figura 12 – Resultados obtidos pelas execuções realizadas com o ALR-LoRaWAN	45
Figura 13 – Resultados obtidos pelas execuções realizadas com o CR-LoRaWAN	46
Figura 14 – Mapa com a situação de recebimento dos dados	49
Figura 15 – Mapa com o intervalo de recebimento dos dados	50
Figura 16 – Mapa com o RSSI de recebimento dos dados	51
Figura 17 – Mapa com o SNR de recebimento dos dados	52

LISTA DE TABELAS

Tabela 1 – Taxa de transmissão e tempo de operação em função do fator de espalhamento SF	21
Tabela 2 – Resultado dos testes	39
Tabela 3 – Número de quadros recebidos por solicitações realizadas.	40
Tabela 4 – Comparação entre mensagens Downlink e dados retransmitidos. . .	47
Tabela 5 – Resultado dos testes	50

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

ACK	Mensagens de Confirmação, do Inglês <i>Acknowledgement</i>
ADR	Taxa de Dados Adaptável, do Inglês <i>Adaptive Data Rate</i>
ARQ	<i>Automatic Repeat Request</i>
CSS	<i>Chirp Spread Spectrum</i>
FCNT	Número de Sequência, do Inglês <i>Frame Count</i>
FSK	Modulação por chaveamento de frequência, do Inglês <i>Frequency-Shift Keying</i>
IDE	Ambiente de Desenvolvimento Integrado, do Inglês <i>Integrated Development Environment</i>
IoT	Internet das Coisas, do Inglês <i>Internet of Things</i>
LARC	Laboratório de Redes de Computadores
LPWAN	Rede de Longa Distância e Baixa Potência, do Inglês <i>Low Power Wide Area Network</i>
M2M	Máquina a Máquina, do Inglês <i>Machine to Machine</i>
MAC	Controle de Acesso ao Meio, do Inglês <i>Media Access Control</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
PAN	Rede de Área Pessoal, do Inglês <i>Personal Area Network</i>
RSSI	Indicador de Intensidade do Sinal Recebido, do Inglês <i>Received Signal Strength Indication</i>
RTO	Tempo Limite de Transmissão, do Inglês <i>Retransmission Time Out</i>
RTT	<i>Round Trip Time</i>
SBC	Sociedade Brasileira de Computação
SF	Fator de Espalhamento, do Inglês <i>Spreading Factor</i>
SMA	<i>Subminiature Version A</i>
SNR	Relação Sinal-Ruído, do Inglês <i>Signal-to-noise ratio</i>
TTN	<i>The Things Network</i>
UTFPR	Universidade Tecnológica Federal do Paraná
WTF	Workshop de Testes e Tolerância a Falhas

ACRÔNIMOS

ALR-LoRaWAN	Algoritmo de Retransmissão em Nível de Aplicação para Redes LoRaWAN, do Inglês <i>LoRaWAN Application Level Retransmission Algorithm</i>
CR-LoRaWAN	Algoritmo de Controle de Retransmissão em Redes LoRaWAN
LoRa	<i>Long Range</i>
LoRaWAN	<i>Long Range Wide Area Network</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
1.2	JUSTIFICATIVA	16
2	REFERENCIAL TEÓRICO	17
2.1	ALGORITMOS DE RETRANSMISSÃO CLÁSSICOS	17
2.2	INTERNET DAS COISAS	18
2.3	LORA	19
2.4	LORAWAN	21
2.4.1	Arquitetura de Rede LoRaWAN	21
2.4.2	Dispositivo Classe A	23
2.4.3	Mecanismo de Retransmissão	23
2.5	SERVIDOR DE REDE E APLICAÇÃO	24
2.6	TRABALHOS RELACIONADOS	25
3	MATERIAIS E MÉTODOS	28
3.1	MATERIAIS	28
3.1.1	TTGO T-Beam	28
3.1.2	Gateway Radioenge	29
3.1.3	LMIC	29
3.1.4	MQTT	30
3.1.5	PAHO	31
3.1.6	Arduino IDE	31
3.2	MÉTODOS	32
3.2.1	Algoritmo Proposto	32
3.2.1.1	Implementação do dispositivo final	33
3.2.1.2	Implementação do servidor de aplicação	33
3.2.2	Testes Práticos	36
3.2.3	Testes Comparativos	37
3.2.4	Testes em Campo	37
4	RESULTADOS E DISCUSSÃO	38
4.1	RESULTADOS PRÁTICOS	38
4.1.1	Ambiente de Realização das Execuções	38
4.1.2	Resultados Obtidos	39
4.2	RESULTADOS COMPARATIVOS	43
4.2.1	Ambiente de Realização das Execuções	43
4.2.2	Resultados Obtidos	44
4.3	RESULTADOS EM CAMPO	48
4.3.1	Ambiente de Realização do Teste	48
4.3.2	Resultados Obtidos	48
5	CONCLUSÕES	53
5.1	TRABALHOS FUTUROS	54
	REFERÊNCIAS	55

ÍNDICE REMISSIVO	59
-------------------------------	-----------

1 INTRODUÇÃO

A partir dos avanços em diversas áreas de tecnologia, dispositivos eletrônicos que antes eram voltados para uso técnico passaram a estar cada vez mais presentes em nosso cotidiano. Uma das principais características desses dispositivos é a sua capacidade de trocar informações, o que só foi possível após os avanços das tecnologias de comunicação (KHANNA; KAUR, 2019). Então, buscando ampliar a aplicabilidade de objetos do nosso dia a dia, utilizando tecnologia e conectando-os a uma rede de dispositivos, surgiu o conceito de Internet das Coisas, do Inglês *Internet of Things* (IoT). A IoT é uma área promissora, atualmente com muita pesquisa e desenvolvimento tanto na academia quanto na indústria. Conforme estimado por Framingham (2019), em 2025 haverá mais de 41 bilhões de dispositivos inteligentes em operação ao redor do mundo.

Esses dispositivos IoT podem utilizar diferentes infraestruturas de redes sem fio para se comunicar, a depender de suas necessidades. Uma rede muito utilizada por sensores é a Rede de Longa Distância e Baixa Potência, do Inglês *Low Power Wide Area Network* (LPWAN). Essa rede tem como característica sua ampla área de comunicação, em determinados cenários podendo chegar a um raio de cobertura de até 50 km e aplicações que visam baixo consumo energético (MEKKI *et al.*, 2017). Criada pela empresa Semtech, a modulação LoRa, derivação para *Long-Range*, é uma das tecnologias LPWAN e utiliza o protocolo de rede chamado LoRaWAN em sua comunicação (SEMTECH, 2019). Por essa tecnologia utilizar uma faixa de frequência não licenciada, possibilita a implantação de uma rede de baixo custo, flexível e em grande escala, sendo assim a opção adotada por várias cidades inteligentes (ROSSATO; SPANHOL, F.; CAMARGO, E., 2020).

Nesse âmbito, estão sendo desenvolvidas diversas aplicações que utilizam a rede LoRaWAN para comunicação, como, por exemplo, na agricultura de precisão, realizando análise de temperatura e umidade do solo (SINGH *et al.*, 2020). Ou ainda no controle de sistemas de irrigação inteligentes (FRAGA-LAMAS *et al.*, 2020). Em ambientes urbanos a rede LoRaWAN é empregada na detecção de níveis de material particulado e qualidade do ar (JOHNSTON *et al.*, 2019) e (PASTÓRIO, A.; SPANHOL, F.; CAMARGO, E., 2022), bem como no rastreamento em tempo real de objetos em movimento (CAMARGO, E. T.; SPANHOL, F. A.; SOUZA, 2021). Para algumas aplicações críticas, como no monitoramento do nível de gases tóxicos em áreas residenciais, se

torna fundamental a garantia de que os dados das aplicações sejam entregues aos servidores de rede. Entretanto, o desafio está na utilização do enlace sem fio para comunicação, atrelado ao impacto de diferentes níveis de obstruções presentes em ambientes urbanos, o que faz com que quadros de dados possam ser perdidos durante a transmissão (FERREIRA *et al.*, 2020).

Porém, o protocolo LoRaWAN não atua sobre os quadros perdidos, apenas fornecendo mensagens de confirmação do recebimento dos dados (LORA ALLIANCE, 2017). Passa a ser papel das aplicações implementar um algoritmo de retransmissão para lidar com casos de instabilidade na rede e perda de quadros. Entretanto, como mencionado por Capuzzo, Magrin e Zanella (2018), devem ser tomadas medidas de precaução quanto a utilização das mensagens de confirmação e retransmissão de sinal, uma vez que aumentar a frequência de dados no meio de propagação impacta na performance do sistema, podendo levar ao congestionamento da rede e um aumento na colisão dos sinais.

O trabalho de Shanmuga Sundaram, Du e Zhao (2020) propõe uma política de retransmissão dinâmica pela variação do temporizador de retransmissão das aplicações. O objetivo é reduzir o número de retransmissões desnecessárias pelo atraso no recebimento da mensagem de confirmação. Enquanto que, Gomez e Crowcroft (2020) apresentam um algoritmo de retransmissão para redes LPWAN focando em um tempo limite de retransmissão adaptável ao *Round Trip Time* (RTT). O RTT é o tempo entre o envio de dados do emissor e o recebimento da mensagem de confirmação transmitida pelo receptor. Já o algoritmo Algoritmo de Controle de Retransmissão em Redes LoRaWAN CR-LoRaWAN proposto por Almeida *et al.* (2021) realiza o gerenciamento de retransmissões nos dispositivos finais de uma rede LoRaWAN. Os resultados apresentados tanto em testes simulados quanto práticos mostram que o algoritmo CR-LoRaWAN aumentou a porcentagem de quadros recebidos. Porém, como essa confirmação é realizada sempre na 5ª transmissão, alguns casos podem ser interpretados erroneamente como confirmados. Outros fatores que não são levados em consideração é o aumento no consumo energético dos dispositivos, no ciclo de trabalho e no congestionamento da rede. Esse algoritmo utiliza Mensagens de Confirmação, do Inglês *Acknowledgement* (ACK) para gerenciamento de retransmissão. Como levantado por Capuzzo, Magrin e Zanella (2018) a utilização excessiva dessa abordagem pode afetar negativamente uma rede LoRaWAN.

O objetivo desse trabalho consiste em desenvolver o Algoritmo de Retransmissão em Nível de Aplicação para Redes LoRaWAN, do Inglês *LoRaWAN Application Level Retransmission Algorithm* (ALR-LoRaWAN). Uma das vantagens da abordagem proposta é diminuir a sobrecarga no dispositivo final em lidar com os quadros perdidos, diminuindo assim o processamento e o consumo energético empregado com a gerencia da retransmissão. O algoritmo proposto gerencia as retransmissões pela comunicação bidirecional entre dispositivo final e servidor de aplicação. Nesta proposta o servidor de aplicação passa a ser responsável por gerenciar todo o tráfego de mensagens enviadas pelos dispositivos finais. Para realizar tal gerenciamento tem-se a utilização do Número de Sequência, do Inglês *Frame Count* (FCNT) presente no protocolo LoRaWAN. Sendo utilizado como base alguns conceitos de algoritmos de retransmissão clássicos apresentados em literatura (FOROUZAN, 2009), mas não se limitando aos mesmos. Resultados realizados em um ambiente real destacam o desempenho do algoritmo ALR-LoRaWAN, mostrando-se capaz de retransmitir com sucesso os quadros perdidos, assim como resultados comparativos com o CR-LoRaWAN apontam as suas vantagens.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Propor e implementar um algoritmo para gerenciamento de retransmissão em redes LoRaWAN visando obter um bom desempenho operacional dos dispositivos finais. O algoritmo será implementado em nível de aplicação a partir da comunicação bidirecional entre servidor de aplicação e dispositivo final.

1.1.2 Objetivos Específicos

- Realizar a análise de algoritmos clássicos de retransmissão em redes de comunicação.
- Propor um novo algoritmo de retransmissão.
- Implementar o algoritmo proposto em uma aplicação.
- Realizar experimentos com o algoritmo proposto.
- Avaliar o desempenho prático do algoritmo proposto.

- Comparar o algoritmo proposto com o algoritmo CR-LoRaWAN.

1.2 JUSTIFICATIVA

Diversas aplicações que utilizam redes IoT para comunicação, como no monitoramento de gases tóxicos em regiões urbanas (JOHNSTON *et al.*, 2019), apresentam a necessidade de garantir que os quadros de dados transmitidos pelos dispositivos finais sejam de fato recebidos pelos pontos de acesso, encaminhados para os servidores e consumidos pelas aplicações (PASTÓRIO, A.; RODRIGUES; CAMARGO, E. de, 2020). Nesse caso em específico, quando se ocorrem falhas durante a comunicação, as informações perdidas podem ocasionar uma análise errônea da qualidade de ar do ambiente, influenciando com isso a qualidade de vida da população presente nessas regiões. Já em aplicações de rastreamento de objetos em movimento, ao se perder dados de localização pode-se gerar a falta de informação do percurso realizado pelos dispositivos, prejudicando a realização de análises e o seu acompanhamento em tempo real.

A rede LoRaWAN, baseada na tecnologia LPWAN, vem sendo muito utilizada por aplicações urbanas. Essa rede, por sua vez, não gerencia as falhas que podem ocorrer durante a comunicação no enlace sem fio entre dispositivo final e ponto de acesso. Atrelado a isso, poucas pesquisas abordam a solução desse problema, muitas delas se voltando para a análise de viabilidade de se tratar o mesmo (MARAIS; ABU-MAHFOUZ; HANCKE, 2020). Com isso, a elaboração de um mecanismo para retransmissão de dados se torna de fundamental importância, garantindo assim confiabilidade para as aplicações desenvolvidas com a tecnologia LoRaWAN.

2 REFERENCIAL TEÓRICO

Esse capítulo aborda a base teórica necessária para se chegar aos objetivos do trabalho. Começando com a descrição de protocolos clássicos para gerenciamento de retransmissão. Na sequência se apresenta uma breve introdução sobre IoT e suas aplicações, explicando conceitos importantes sobre as redes LPWAN e uma tecnologia de implementação da mesma, a rede LoRaWAN, comentando sobre a modulação LoRa, o protocolo LoRaWAN e o serviço em nuvem *The Things Network* (TTN). A finalização do capítulo é realizada com a apresentação dos trabalhos relacionados ao tema.

2.1 ALGORITMOS DE RETRANSMISSÃO CLÁSSICOS

Em literatura clássica de redes de computadores, como em Kurose e Ross (2013) e Forouzan (2009), são abordados alguns protocolos para gerenciamento de retransmissão de dados como:

- ***Stop and Wait com Automatic Repeat Request (ARQ)***: O protocolo *Stop and Wait* com ARQ fornece ao receptor a possibilidade de verificar se pacotes recebidos apresentam algum erro, isso é feito com a utilização de bits de redundância. Caso o pacote de dados não apresente nenhum erro ao chegar no receptor, o mesmo responde ao emissor uma mensagem ACK o avisando que não precisa ser retransmitido. Já, se alguma inconsciência é percebida no pacote de dados, não é enviada nenhuma resposta ao emissor, o mesmo passando a retransmitir o dado quando encerrado o temporizador de retransmissão. Esse protocolo apresenta um grande problema relacionado a desempenho e utilização do canal de comunicação, uma vez que, ao enviar um pacote de dado, o remetente não transmite mais pacotes até o recebimento de um ACK ou finalização do temporizador, ficando, durante todo esse período, o canal sem ser utilizado (FOROUZAN, 2009).
- ***Go-back-N***: Para que o remetente utilize ao máximo a capacidade do link de comunicação, no protocolo *Go-back-N* o mesmo envia diferentes pacotes de dados simultaneamente, aguardando então a confirmação de algum desses pacotes enviados antes de voltar a transmitir um próximo pacote de dados. Para saber quais dos pacotes de dados já foram recebidos, esse algoritmo

utiliza um número de sequência associado a cada um dos pacotes enviado, com isso, o destinatário, ao receber um novo pacote não corrompido e que esteja na sequência de pacotes esperados, retorna ao remetente uma mensagem de confirmação contendo o número de sequência do último pacote confirmado. Já se o destinatário receber um pacote corrompido, ele envia o número de sequência do próximo pacote de dados esperado (KUROSE; ROSS, 2013).

- **Repetição seletiva:** Esse protocolo adiciona uma maior complexidade no lado receptor da comunicação, onde o mesmo passa a armazenar o número de sequência dos pacotes recebidos fora de ordem em um *buffer*, dessa forma, serão enviadas mensagens de confirmação negativa apenas para os pacotes desejados, reduzindo o número de mensagens duplicados enviados pelo emissor. Como em enlaces com ruído há uma alta chance dos dados chegarem fora de ordem, esse algoritmo se torna mais eficiente do que o *Go-back-N* (FOROUZAN, 2009).

2.2 INTERNET DAS COISAS

Cunhado pelo cientista da computação Kevin Ashton em 1999, o termo Internet das Coisas caracteriza objetos e dispositivos com capacidade de se conectar a Internet e a partir dela trocar informações (KRANENBURG; BASSI, 2012). O que tornou a comunicação entre dispositivos eletrônicos possível foram os avanços realizados em diversas áreas de tecnologia, como em sistemas embarcados e nas tecnologias de comunicação (SANTOS *et al.*, 2016).

O conceito de IoT se abrange para diferentes áreas, estando relacionado às aplicações com capacidade de se conectar e trocar informações pela Internet, uma rede de computadores globalmente interconectados (FOROUZAN, 2009). Isso permite que objetos possam realizar tarefas além das que haviam sido primordialmente projetados a fazer, como é o caso de fechaduras inteligentes (MARTÍNEZ; ERAS; DOMÍNGUEZ, 2018), que podem ser acessadas de forma remota por seus usuários, verificando quando as mesmas estão sobre risco de violação. Também surgem aplicações que passam a ser viáveis por conta da capacidade de comunicação e autonomia, como na utilização de sensores para monitoramento do solo em áreas rurais (SINGH *et al.*,

2020). Com isso, por possibilitar inovações em diferentes áreas, as aplicações que envolvem IoT estão ganhando destaque tanto no campo acadêmico quanto comercial (SANTOS *et al.*, 2016).

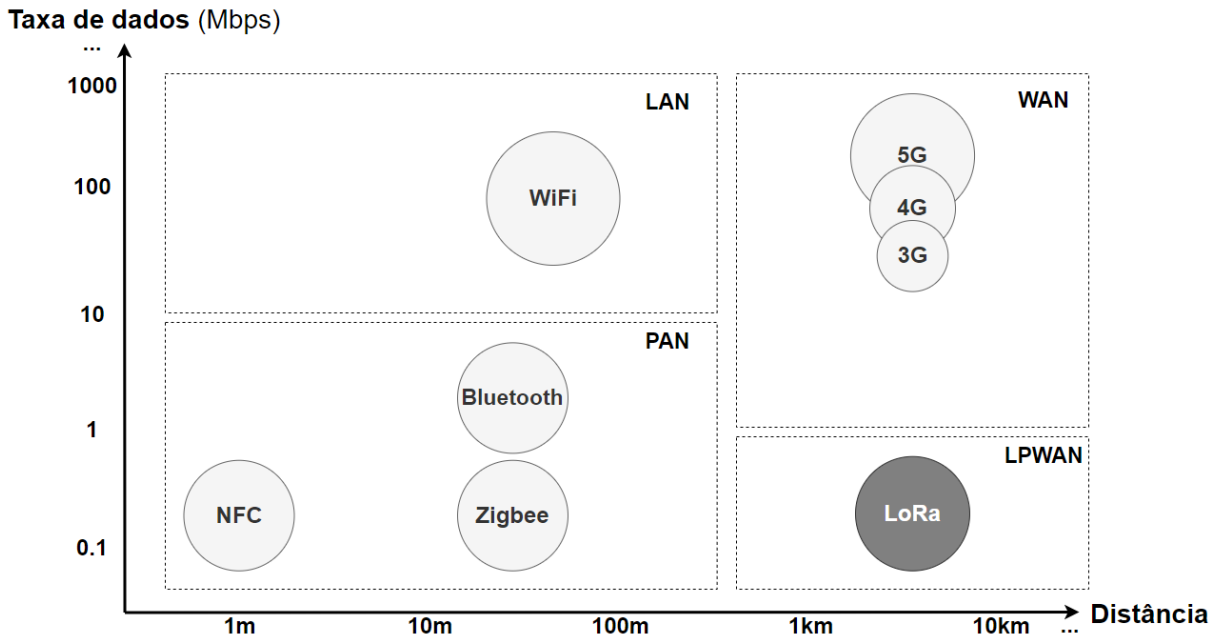
As aplicações que utilizam por base a comunicação entre objetos estão presentes em cenários urbanos, rurais, residenciais e industriais. Cada uma dessas diferentes áreas necessitam de uma infraestrutura de rede específica para atender as suas demandas. No caso de aplicações residenciais são muito utilizadas as Rede de Área Pessoal, do Inglês *Personal Area Network* (PAN), como o protocolo de comunicação Bluetooth, por permitir que os dispositivos troquem informações por meio do enlace sem fio de forma segura, com uma área de abrangência em torno de dezenas de metros e por apresentar eficiência energética em sua versão *Bluetooth Low Energy* (SANTOS *et al.*, 2016).

Quando se busca uma maior área de abrangência durante a comunicação, podendo chegar a mais de 10 km de distância (RAJ; RAMAN, 2017), duas tecnologias de rede se fazem presente, as tecnologias baseadas em WAN e LPWAN. Diferente das redes WAN que tem por objetivo uma alta taxa de transferência de dados, as redes LPWAN buscam o melhor desempenho energético de seus dispositivos em uma ampla área de comunicação. Dessa forma, as redes LPWAN passam a ser muito utilizadas por sensores, esses realizam troca de informações com poucos quadros de dados e entre um maior intervalo de tempo, permanecendo em hibernação por longos intervalos, sendo ativados apenas quando precisam trocar suas informações com a rede (RAJ; RAMAN, 2017).

2.3 LORA

Concebida pela Semtech, LoRa abreviação para *Long Range* opera na camada física da comunicação entre os dispositivos IoT, sendo essa uma tecnologia de modulação de rádio frequência desenvolvida para redes LPWAN (SEMTECH, 2019). Dessa forma, os dispositivos que a utilizam para comunicação conseguem trocar informações em uma ampla área de abrangência, e apresentam um baixo consumo energético, estando associado à baixa transferência de dados. Na Figura 1 temos uma comparação ao que rege os aspectos de distância e transferência de dados entre essa tecnologia e demais tecnologias utilizadas por dispositivos IoT (CHAUDHARI; ZENNARO, 2020).

Figura 1 – Tecnologias de rede



Fonte: autoria própria.

Os dispositivos baseados em LoRa convertem seus dados digitais em sinais de rádio frequência utilizando a técnica *Chirp Spread Spectrum* (CSS). A técnica de modulação CSS consiste na utilização do máximo da largura de banda disponível para transmissão do sinal por meio da alteração em frequência, semelhante ao Modulação por chaveamento de frequência, do Inglês *Frequency-Shift Keying* (FSK). A diferença do CSS para o FSK está na forma de modulação baseada em frequências, enquanto o FSK utiliza duas ou mais frequências de modo discreto, uma para cada símbolo, o CSS varia continuamente entre as frequências. Essa variação em frequência pode representar pulsos de *Upchirp*, alteração de uma frequência alta para baixa ou *Downchirp*, mudança de uma frequência baixa para alta (SENEVIRATNE, 2019).

Os dispositivos que utilizam a modulação LoRa em sua camada física apresentam também o mecanismo de Fator de Espalhamento, do Inglês *Spreading Factor* (SF). Esse mecanismo realiza a alteração do número de *Chirps* transmitidos por segundo, o que impacta na distância que um sinal pode ser detectado, mas aumenta o tempo de transmissão do mesmo (SEMTECH, 2019). O fator de espalhamento pode assumir 6 valores diferentes, como mostra a Tabela 1, começando em SF7, com o maior número de bits/s, o que reduz o tempo gasto para transmissão do sinal e por consequência aumentando a vida útil da bateria dos dispositivos, indo até SF12, com uma menor taxa de bits/s, porém aumentando a distância em que o sinal pode ser detectado (SEMTECH,

2019).

Tabela 1 – Taxa de transmissão e tempo de operação em função do fator de espalhamento SF.

SF	Taxa de dados (bits/s)	Tempo de operação (ms)
7	5470	56
8	3125	103
9	1758	205
10	977	371
11	537	741
12	293	1483

Fonte: autoria própria (2022), adaptação Semtech (2019).

2.4 LORAWAN

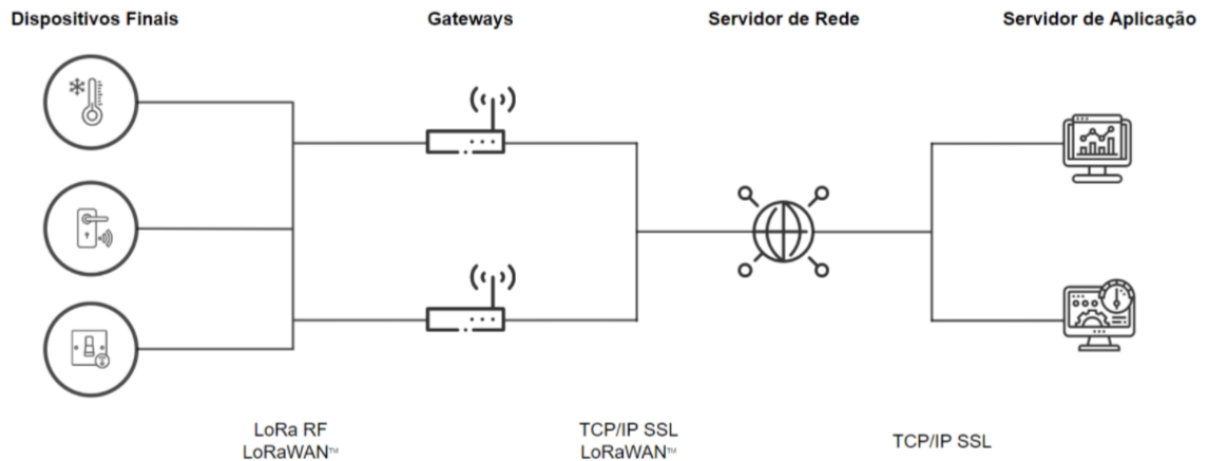
Para lidar com demais aspectos de uma rede de comunicação de dispositivos IoT, a LoRa Alliance desenvolveu o protocolo de rede LoRaWAN, abreviação para *Long Range Wide Area Network*. Esse protocolo rege a topologia da arquitetura de uma rede LoRa e o processo de comunicação entre dispositivos, provendo, na troca de informações, qualidade de serviço, segurança e melhor desempenho energético dos dispositivos (SENEVIRATNE, 2019).

2.4.1 Arquitetura de Rede LoRaWAN

Na Figura 2 é apresentada a arquitetura das redes LoRa. A arquitetura segue uma topologia de rede em estrela, onde os nós são os dispositivos finais, sensores e atuadores interagindo com o ambiente. Como ponto central, responsável por gerenciar o redirecionamento das mensagens, tem-se os pontos de acesso ou *Gateways*. As redes LoRa apresentam um processo de comunicação bidirecional, podendo haver transmissão de quadros dos dispositivos finais para os pontos de acesso ou dos pontos de acesso para os dispositivos finais (SEMTECH, 2019).

Os dispositivos finais, presentes nas redes LoRa enviam seus dados para os *Gateways* por meio do enlace sem fio utilizando a modulação CSS. Esses dados, após serem recebidos por pelo menos um ponto de acesso, são direcionados para o servidor de rede relativo a aplicação que o dispositivo final está registrado, por meio de uma conexão IP (SEMTECH, 2019). Como os *Gateways* agem apenas como encaminha-

Figura 2 – Arquitetura de redes LoRaWAN



Fonte: autoria própria.

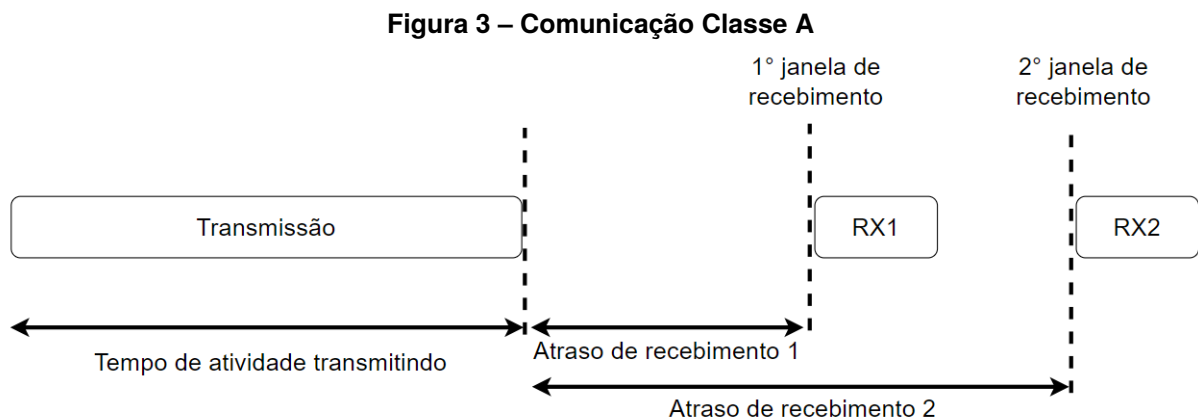
dores, os servidores de rede são responsáveis por gerenciar toda a comunicação nas redes LoRaWAN, lidando com a segurança dos dados e garantia de entrega. Então os servidores de rede encaminham os dados recebidos para serem consumidos pelos servidores de aplicação, o que é realizado por meio das chaves de acesso fornecidas pelo dispositivo (ALLIANCE, 2017). Nas aplicações se realiza todo o tratamento dos dados enviados pelos dispositivos, podendo também ocorrer envios de mensagens em resposta a esses dados. Todo o processo de comunicação que parte do dispositivo final para as aplicações é referenciado como *Uplink* (SEMTECH, 2019).

O processo de transmissão inverso ao *Uplink* é denominado *Downlink*, o que ocorre partindo das aplicações, ou servidor de rede, para os dispositivos finais. Como múltiplos pontos de acesso podem receber o mesmo dado de um dispositivo final, o ponto de acesso com maior Indicador de Intensidade do Sinal Recebido, do Inglês *Received Signal Strength Indication* (RSSI) é escolhido para transmitir as mensagens de *Downlink* (SEMTECH, 2019).

Como os dispositivos finais em uma rede LoRa permanecem a maior parte do tempo desativados para economizar bateria, o protocolo LoRaWAN dispõe três modos de operação com o objetivo de coordenar a comunicação entre dispositivos finais e servidores (LAVRIC; POPA, 2017), descritos a seguir.

2.4.2 Dispositivo Classe A

A classe A foi projetada para atingir o maior desempenho energético dos dispositivos. Nessa classe o dispositivo final permanece desativado até precisar enviar alguma informação para a rede, como mostra a Figura 3. Assim que essa informação é completamente encaminhada, o mesmo aguarda um certo tempo para abrir a primeira janela de recebimento de dados, onde, em sincronia com o servidor de rede, pode receber mensagens de *Downlink*. Caso nenhuma mensagem seja recebida pelo dispositivo em sua primeira janela de recepção, o mesmo aguarda um segundo intervalo e abre uma segunda oportunidade de recebimento de dados. Após as duas tentativas, o dispositivo é desativado até precisar enviar um novo quadro de dados (SEMTECH, 2019).



Fonte: autoria própria.

Diferente da classe A, a classe B utiliza mensagens *Beacon* para sincronizar com o servidor intervalos periódicos de abertura de janela de recepção, por necessitar de ativações periódicas a utilização dessa classe implica em um maior consumo energético. Já os dispositivos de classe C permanecem a todo o momento ativos, com uma janela de recepção aberta, esse caso é utilizado para aplicações que independem de eficiência energética (SEMTECH, 2019).

2.4.3 Mecanismo de Retransmissão

Como mencionado por Almeida *et al.* (2021), o protocolo LoRaWAN não fornece um mecanismo para gerenciamento de retransmissões e tolerância a falhas durante

a comunicação, apresentando apenas a possibilidade de se solicitar mensagens de confirmação (ACK) em cada quadro de dados enviado. Com o ACK é possível saber apenas se o quadro de dado atual foi recebido pelo ponto de acesso em casos de *Uplink* ou pelo dispositivo final em casos de *Downlink*.

Como documentado em Alliance (2017), no protocolo LoRaWAN as mensagens trocadas entre dispositivo final e ponto de acesso apresentam em sua estrutura de cabeçalho um Número de Sequência, do Inglês *Frame Count* (FCNT). O FCNT é um número incremental para mensagens *Downlink* e *Uplink*, esse é utilizado como identificação particular de cada mensagem. Os números de sequência são muito utilizados em protocolos de redes clássicos, como no protocolo TCP, onde o mesmo é empregado para gerenciar a troca de mensagem entre emissor e receptor. No protocolo TCP, esse número é utilizado para ordenar as mensagens no destinatário, uma vez que os quadros de dados enviados podem assumir diferentes caminhos durante a sua propagação pela rede (FOROUZAN, 2009).

O gerenciamento dos números de sequência *Downlink* pode ser realizado pelo dispositivo final, já, para as mensagens *Uplink* isso é realizado pelos servidores de rede e aplicação. Passando a ser uma estrutura considerada para validar se mensagens de dados foram perdidas durante a comunicação. O que se torna objeto de estudo desse trabalho.

2.5 SERVIDOR DE REDE E APLICAÇÃO

Todos os dados que trafegam em uma rede LoRaWAN são repassados para o servidor de rede assim que recebidos pelos *Gateways*. Sendo então o servidor de rede responsável por gerenciar toda a comunicação que ocorre em uma rede LoRaWAN. Alguns dos papéis desempenhados pelo servidor está em autenticar o dispositivo emissor das mensagens, realizar a filtragem de mensagens duplicadas recebidas pelos pontos de acesso, enviar respostas ACK aos dispositivos finais, adequar a taxa de dados do envio das mensagens para melhor desempenho da rede e encaminhar os dados para a aplicação correta (INDUSTRIES, 2020).

Desenvolvido pela *The Things Industries*, o *The Things Stack* é um servidor de código aberto e colaborativo de implementação da camada de rede do protocolo LoRaWAN. Com o *The Things Stack* executando em nuvem, é possível conectar os

pontos de acesso em uma rede colaborativa para ser utilizado por demais aplicações presentes em sua região de cobertura. Além disso, o mesmo proporciona mecanismos para gerenciamento da camada de rede das redes LoRaWAN. Conseguindo com ele administrar todos os componentes presentes na rede, como os pontos de acesso, dispositivos finais, aplicações e encaminhar o tráfego de dados para integrações pré-definidas (INDUSTRIES, 2020).

O *The Things Stack MQTT Server* é uma integração possível com a utilização do servidor de rede *The Things Stack*. Como o MQTT é um protocolo em camada de aplicação, com ele é possível gerenciar as mensagens da rede LoRaWAN por um cliente MQTT a partir dos tópicos de publicar e subscrever (HILLAR, 2018):

- **Publicar:** Com o mecanismo de publicação é possível enviar mensagens de *Downlink* para a rede com destino aos dispositivos finais.
- **Subscrever:** Com o mecanismo de subscrever é possível registrar o cliente para ter acesso a todos os dados de *Uplink* recebidos pelo servidor de rede, podendo assim configurar um servidor de aplicação próprio.

2.6 TRABALHOS RELACIONADOS

Relacionado ao tema de retransmissão em redes LoRaWAN, o trabalho de Capuzzo, Magrin e Zanella (2018) fornece uma visão do impacto da utilização de mensagens de confirmação, e consequentemente retransmissões, por dispositivos finais. Os resultados do estudo foram obtidos a partir da utilização do simulador de código aberto NS-3. O cenário simulado consistia em um *Gateway* atuando como receptor de diferentes números de dispositivos finais, esses com capacidade de enviar mensagens com e sem confirmação, assim como retransmissões quando os quadros com confirmação forem perdidos. Para as interferências do meio de propagação o estudo utilizou perdas por distância logarítmica, fatores de sombreamento e construções.

Os resultados da pesquisa realizada por Capuzzo, Magrin e Zanella (2018) mostram que o aumento no número de dispositivos finais utilizando mensagens de confirmação impacta negativamente o desempenho da rede. Com isso, expõem o risco de se utilizar mensagem de confirmação excessivamente. Para contornar tal problema, algumas soluções são elencadas pelos autores, como a utilização de estratégias para adaptar parâmetros do sistema e o aumento no número de *Gateways* para balancear a

carga de mensagens necessárias e o tempo que os mesmos passam transmitindo.

Já, o trabalho de Marais, Abu-Mahfouz e Hancke (2020) analisa diferentes cenários que utilizam mensagem de confirmação, verificando a viabilidade da sua utilização e propondo, com base em demais pesquisas científicas no segmento, alternativas para o melhor desempenho da rede quando essas mensagens forem necessárias. Uma das propostas é a utilização da estratégia de retransmissão por *back-off*, como abordado em Alliance (2017). Nessa estratégia é utilizado um intervalo de tempo aleatório para a tentativa de retransmissão dos dispositivos finais, evitando que ocorra transmissões simultâneas de mensagens de confirmação e sobrecarregue os *Gateways*. Nessa estratégia, caso a retransmissão não seja realizada, passa a ser papel do servidor de rede gerenciar o que deve ser feito com o quadro de dados não recebido.

Outra abordagem, realizada por Abdelfadeel *et al.* (2020) é a utilização de um esquema de agendamento, nesse caso os dados são armazenados no dispositivo final e apenas transmitidos de forma agrupada em um mesmo *payload*, enviando assim quadros de dados maiores, porém com uma menor frequência. Com essa ação as mensagens de ACK são válidas para todo o grupo de dados enviado, reduzindo o número de mensagens de confirmação solicitadas pelos dispositivos finais.

Em Gomez e Crowcroft (2020) outro aspecto é levado em consideração ao lidar com retransmissão em redes LPWAN. Esse trabalho propõe a utilização do Tempo Limite de Transmissão, do Inglês *Retransmission Time Out* (RTO) de forma multimodal, considerando diferentes cenários para o RTT, que calcula o tempo entre ida e volta de uma mensagem trocada entre o emissor e receptor. Nos resultados do trabalho são levantadas as diferenças entre a utilização de um algoritmo padrão RTO e de sua versão multimodal. Verifica-se ainda como a utilização multimodal reduz a latência das mensagens de confirmação e a entrega de quadros em momentos de perda de sinal.

Como concluído no trabalho de Pastório, J. P., Souza e Camargo, E. T., apesar do acréscimo no número de *Gateways* auxiliares na entrega de quadros em redes LoRa, as obstruções presentes no meio sem fio impactam diretamente na redução da porcentagem de quadros recebidos, restringindo a confiabilidade da entrega. Surge com essa limitação a necessidade de se utilizar outros mecanismos para auxiliar a entrega de mensagens em redes LoRa, como pelo emprego de algoritmos para gerenciamento de retransmissões.

Em Almeida *et al.* (2021) é proposto um algoritmo de retransmissão para redes

LoRaWAN, o CR-LoRaWAN. Nesse algoritmo as mensagens de confirmação são interpretadas para grupos de cinco quadros de dados, dessa forma caso o quinto quadro enviado com a mensagem ACK não seja recebido, o dispositivo final retransmite todos os cinco quadros referenciados pela mesma. O mesmo ocorre quando a mensagem de confirmação é recebida, podendo ocorrer a interpretação errônea de quadros de dados como confirmados, quando os mesmos podem ter sofrido alguma interferência e não chegaram a ser recebidos pelos pontos de acesso. Outro detalhe desse algoritmo surge caso seja utilizado um curto intervalo entre mensagens de confirmação, podendo ocorrer o mesmo problema abordado por Capuzzo, Magrin e Zanella (2018).

3 MATERIAIS E MÉTODOS

Nesse Capítulo serão apresentados os materiais utilizados para se chegar ao objetivo da pesquisa, como dispositivos, bibliotecas e ferramentas de programação. Também serão demonstrados os métodos empregados para o seu desenvolvimento.

3.1 MATERIAIS

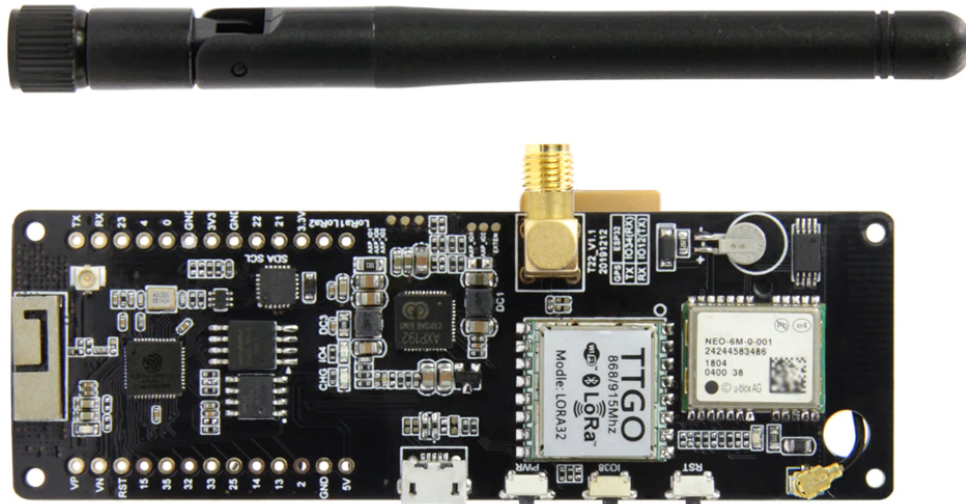
São elencados então os materiais utilizados para desenvolvimento dos objetivos propostos, incluindo os dispositivos, como a placa de desenvolvimento TTGO TBeam e Gateway LoRaWAN RD43HATGPS utilizado juntamente com *Raspberry Pi*. Também é apresentada a biblioteca LMIC para comunicação em redes LoRaWAN e ferramentas de programação utilizadas.

3.1.1 TTGO T-Beam

O TTGO T-Beam, mostrado na Figura 4, é uma placa de desenvolvimento voltada para aplicações de redes IoT de baixo consumo energético. Essa placa apresenta os seguintes componentes (LILYGO, 2021):

- **Chip ESP32:** Microcontrolador *dual-core* com 4MB de memória flash, integrado com WiFi, Bluetooth e modulação LoRa, muito utilizado em aplicações IoT.
- **Semtech SX1276/78:** Transceptor responsável por comunicar em rádio frequência
- **Antena Subminiature Version A (SMA):** Antena para transmissão de sinais de rádio frequência operando a 915MHz.
- **Modulo NEO-6 GPS:** Antena de cerâmica para recebimento dos dados geográficos.
- **Pinos de entrada e saída:** 26 pinos de entrada e saída, incluindo GPIO, 3.3V, 5V, SPI, ADC e DAC.
- **3 Botões:** *Power, Reset* e GPIO38

Figura 4 – Placa de desenvolvimento - TTGO Tbeam



Fonte: Lilygo (2021).

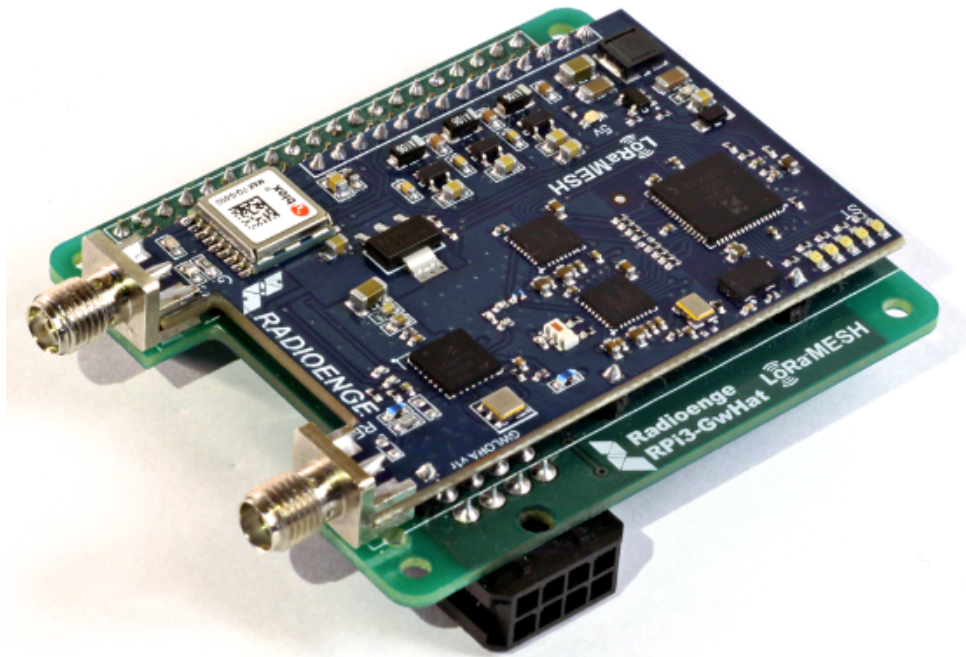
3.1.2 Gateway Radioenge

Como ponto de acesso será utilizado o *Shield RD43HAT / RD43HATGPS*, apresentado na Figura 5 em conjunto com *Raspberry Pi 3* modelo B+. O RD43HAT, desenvolvido pela Radioenge (2018), consiste em um *Gateway* para redes LoRaWAN, apresentando 8 canais de recebimento e transmissão de dados por modulação CSS e utilizando antena de radio frequência que opera na banda de frequência não licenciada de 902,5 a 928MHz (RADIOENGE, 2018).

3.1.3 LMIC

A biblioteca LMIC, desenvolvida pela IBM Corporation em 2014, implementa a camada de Controle de Acesso ao Meio, do Inglês *Media Access Control* (MAC) do protocolo LoRaWAN versão 1.0.2, disponibilizada por Moore e MCCI (2018). Essa biblioteca foi desenvolvida em linguagem de programação C, suportando dispositivos classe A e B, assim como transceptores SX1272 e SX1276. Com sua utilização é possível abstrair toda a configuração necessária para realizar a comunicação entre dispositivo final e *Gateway*, utilizando eventos para tornar esse um processo direto (MCCI, 2018). As chamadas desses eventos e suas funções correspondentes são implementadas no presente trabalho. Algumas das funções fornecidas pela biblioteca são:

Figura 5 – Gateway RD43HAT Radioenge



Fonte: Radioenge (2018).

- *onEvent*: Podendo gerenciar alterações em eventos, como quando ocorre a recepção de dados, e gerar gatilhos para outras funções.
- *LMIC_startJoining*: Realiza o processo de *join* na rede LoRaWAN, executando a primeira troca de mensagens com o servidor de rede.
- *LMIC_setTxData2*: Utilizado para enviar quadros de dados na frequência configurada, podendo solicitar mensagens de confirmação.

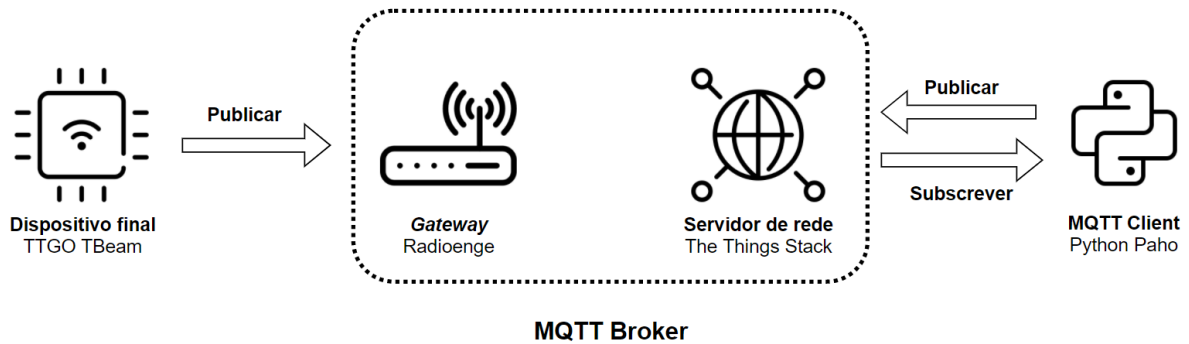
3.1.4 MQTT

O *Message Queuing Telemetry Transport* (MQTT) é um protocolo de comunicação que executa sobre o TCP/IP. Esse protocolo foi desenvolvido para comunicação entre dispositivos Máquina a Máquina, do Inglês *Machine to Machine* (M2M), buscando utilizar pouca largura de banda e mensagens leves, sendo muito utilizado em dispositivos IoT que visam um baixo consumo energético, como é o caso de sensores presentes em redes LoRa (HILLAR, 2018).

Esse protocolo utiliza o mecanismo de publicar e subscrever para realizar o processo de troca de mensagens, como mostra a Figura 6. Os dispositivos, ao enviar quadros de dados para os pontos de acesso, que são encaminhados para o servidor de rede, estão publicando uma nova mensagem ao MQTT Broker. Os dados recebidos

pelo MQTT Broker podem ser coletados por aplicação com a utilização do *subscreever*, para isso é necessário que o servidor de aplicação se registre como um cliente MQTT. Essa mesma aplicação pode publicar dados ao MQTT Broker, esses dados publicados muitas vezes são mensagens *downlink* que serão encaminhadas ao dispositivo final.

Figura 6 – Diagrama de troca de mensagens com MQTT



Fonte: autoria própria.

3.1.5 PAHO

Paho é um projeto que busca a implementação de bibliotecas de código aberto voltadas a comunicação entre dispositivos utilizando o protocolo MQTT. Essas bibliotecas são desenvolvidas para diferentes linguagens de programação, facilitando a criação de aplicações e sua integração com as redes IoT a partir de um cliente MQTT. Para a linguagem de programação Python é disponibilizada a biblioteca *paho-mqtt* para operar um cliente MQTT por um servidor de aplicação. Ao subscreever a aplicação no servidor de rede, que está executando o MQTT Server, o cliente MQTT passa a ser capaz de receber todo o tráfego de dados *uplink* da rede destinado à aplicação registrada na subscrição. Também se torna possível publicar mensagens em tópicos específicos, sendo esses tópicos os dispositivos finais registrados no servidor de rede (HILLAR, 2018).

3.1.6 Arduino IDE

Um Ambiente de Desenvolvimento Integrado, do Inglês *Integrated Development Environment* (IDE) é utilizado para auxiliar no desenvolvimento de software pela in-

tegração de diversas ferramentas em uma única plataforma. Para programação das placas de desenvolvimento foi utilizado o Arduino IDE, uma vez que, o mesmo fornece a possibilidade de se implementar e carregar códigos utilizando as diversas bibliotecas e módulos suportados pelo chip ESP32 presente na placa TTGO Tbeam 3.1.1 (MONK, 2017).

3.2 MÉTODOS

Para lidar com perda de sinais em redes LoRa, e se chegar em um algoritmo que realize o gerenciamento de retransmissões, é proposta uma abordagem em nível de aplicação. Com base no primeiro passo da pesquisa sendo o estudo dos algoritmos e protocolos clássicos de retransmissão abordados em literaturas, é proposta então uma estratégia que teve como inspiração protocolos como *Go-Back-N*, repetição seletiva e TCP (FOROUZAN, 2009).

3.2.1 Algoritmo Proposto

O algoritmo para gerenciamento de retransmissão proposto consiste em uma comunicação bidirecional entre dispositivo final e servidor de aplicação, como mostra a Figura 6, tendo sua implementação em dispositivo e servidor de aplicação descrita nas sessões que seguem. Sendo utilizado como dispositivo final a placa de desenvolvimento TTGO TBeam 3.1.1, por apresentar um *Chip* ESP32 com modulação LoRa e ser voltada ao desempenho energético das aplicações que a utilizam. Esse dispositivo será configurado com a biblioteca LMIC 3.1.3, responsável por realizar a transmissão e recebimento de dados pelo transceptor SX1276 presente na placa, assim como codificar e decodificar os quadros de dados. Para comunicação com a aplicação, os dados serão recebidos por um ponto de acesso Radioenge RD43 3.1.2 e encaminhados para o servidor de redes *The Things Stack*. O servidor será integrado com o MQTT Server 3.1.4, possibilitando a comunicação com um MQTT *Client*, esse será programado com a linguagem Python utilizando a biblioteca *Eclipse Paho* 3.1.5.

3.2.1.1 Implementação do dispositivo final

Como no dispositivo final ocorre a transmissão de dados *uplink* e o recebimento e manipulação das mensagens *downlink* enviadas pelo servidor, o fluxo de dados referente a programação realizada no dispositivo é mostrado na Figura 7. Pelo fluxograma apresentado, podemos ver que uma vez recebida a mensagem de *downlink*, o dispositivo final decodifica a mesma e extrai a mensagem encaminhada pelo servidor de aplicação. Nesses dados, a partir do valor de FCNT, contém quais são as mensagens que precisam ser retransmitidos pelo dispositivo final, podendo a mensagem recebida conter um conjunto de valores FCNTs que precisam ser retransmitidos. Para identificação do quadro de dados que está sendo retransmitido, o dispositivo final encaminha na mesma mensagem o valor do FCNT que está sendo retransmitido, ou se a mensagem representa um envio original.

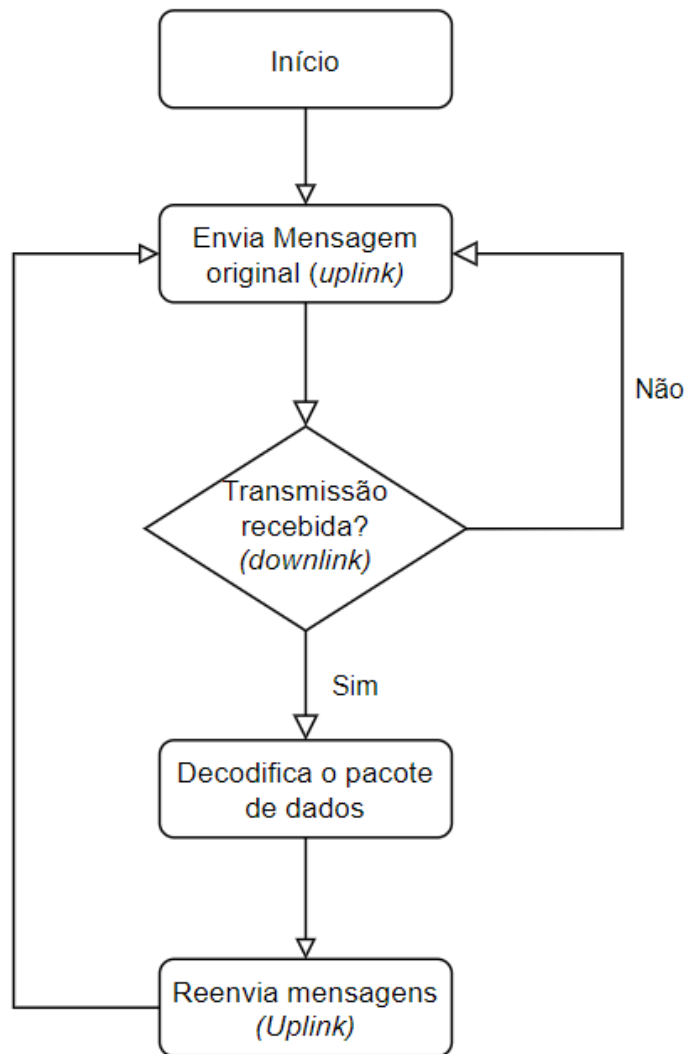
Para a execução durante a obtenção dos resultados foi adaptada uma aplicação referente ao rastreamento de dispositivos, nessa aplicação foram implementadas as transmissões e retransmissões de mensagens para atuarem conforme a Figura 7. Para contornar a perda de mensagens, se torna necessário a utilização de um *Buffer* interno para armazenar as mensagens enviadas, como o número de sequência e seus respectivos dados de latitude e longitude que haviam sido transmitidos. Dessa forma, sendo capaz de realizar a retransmissão das mesmas quando solicitado pelo servidor de aplicação.

Com a abordagem utilizada, o dispositivo final retransmite apenas os quadros de dados necessários, não sobrecarregando os pontos de acesso com mensagens de confirmação ou enviando quadros duplicados. Toda a implementação do dispositivo final foi realizada com a linguagem de programação C++ em conjunto com o Arduino IDE 3.1.6, utilizando também a biblioteca LMIC para implementação do protocolo LoRaWAN 3.1.3.

3.2.1.2 Implementação do servidor de aplicação

No servidor de aplicação, onde está sendo executado o MQTT *Client* com a utilização da biblioteca Paho para Python, foi desenvolvido o algoritmo como demonstra a Figura 8. A implementação seguiu o fato de que os dados são recebidos com um

Figura 7 – Fluxograma da troca de mensagens do dispositivo final

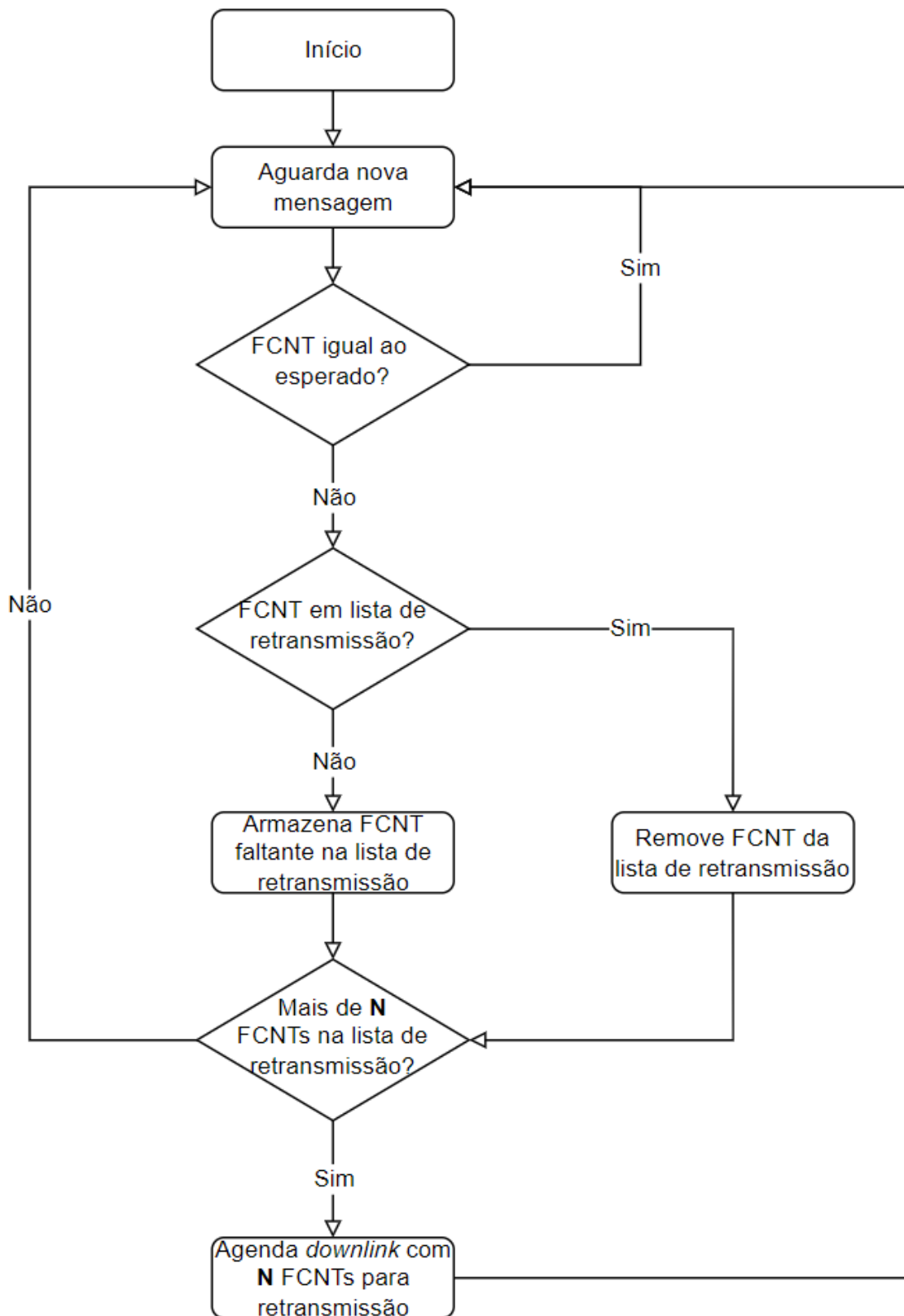


Fonte: autoria própria.

número de sequência FCNT, explicado na sessão 2.4.3.

Como o FCNT é sequencial e incrementado em uma unidade para cada quadro enviado pelo dispositivo, é possível gerenciar quais quadros de dados não foram recebidos corretamente. Desse modo, como mostra o fluxograma da Figura 8, sempre que o servidor de aplicação recebe um quadro com FCNT fora de ordem é realizada uma verificação. Caso esse quadro ainda não tenha sido identificado pelo servidor o mesmo é salvo em um *Buffer* que contém somente os dados que foram perdidos durante a comunicação. Já se o número de sequência do dado recebido fora de ordem estiver presente nesse mesmo *Buffer*, o servidor de aplicação interpreta essa mensagem como sendo uma retransmissão e o remove da lista de solicitações de retransmissão. Os quadros não recebidos irão compor uma mensagem *downlink* a ser enviada ao

Figura 8 – Fluxograma do mecanismo de retransmissão



Fonte: autoria própria.

dispositivo solicitando a retransmissão dos dados faltantes. Essa mensagem *downlink* só será agendada pelo servidor de aplicação para transmissão quando houver **N** quadros de dados perdidos que irão compor uma única mensagem de solicitação de retransmissão. Após o dispositivo final reenviar os quadros faltantes, e os mesmos serem recebidos pelo servidor de aplicação, eles são removidos da lista de quadros perdidos.

O valor do fator de retransmissão **N** é ajustável com o propósito de verificar o seu impacto no desempenho da rede e nas solicitações de retransmissões. Busca-se, dessa forma, um algoritmo que consiga se adaptar para atender aplicações com diferentes requisitos. Abordando aplicações que necessitam de retransmissões instantâneas, logo após a identificação de uma mensagem faltante. Também podendo se adequar para casos que toleram um período maior para obtenção dos dados perdidos, dessa forma sendo possível solicitar mais de uma retransmissão por mensagem e reduzir o número de *downlinks* realizados pela rede. Serão testados diferentes valores referentes ao fator de retransmissão **N** durante a obtenção dos resultados.

3.2.2 Testes Práticos

Foram realizados testes de desempenho a fim de validar o aumento da confiabilidade na entrega de quadros com a utilização do algoritmo ALR-LoRaWAN. Será analisada dessa forma a troca de mensagens entre dispositivo final e servidor de aplicação em execuções utilizando diferentes configurações do servidor de aplicação com a alteração no valor do fator de retransmissão **N**. Será investigado o número de mensagens perdidas durante as execuções, a partir dessas mensagens perdidas quantas solicitações de retransmissões foram realizadas, e após a realização das retransmissões qual foi a porcentagem de quadros recebidos. Também são obtidas métricas quanto ao tempo decorrido entre recebimento dos dados, para verificar se apresenta um maior intervalo de tempo conforme se aumenta o valor de **N**, uma vez que esse interfere em quando a mensagem de solicitação de retransmissão será enviada.

Investiga-se assim se a aplicação apresenta uma redução no impacto das interferências presentes no meio de propagação que ocasionam perda das mensagens transmitidas pelo dispositivo. Além disso, será explorado o seu efeito na utilização dos pontos de acesso, a partir do número de mensagens geradas pelo algoritmo, buscando

examinar o seu congestionamento, levando em consideração as limitações regionais de acesso ao meio (MARAIS; ABU-MAHFOUZ; HANCKE, 2020).

3.2.3 Testes Comparativos

Além dos testes práticos de desempenho com o ALR-LoRaWAN, também se apresenta a comparação entre a abordagem adotada nesse trabalho e o algoritmo CR-LoRaWAN desenvolvido por Almeida *et al.* (2021). O objetivo é comparar ambos algoritmos em diferentes execuções, analisando as transmissões das mensagens, dessa forma verifica-se como cada abordagem se comporta com mensagens recebidas e não recebidas pelos pontos de acesso, podendo assim analisar as transmissões e retransmissões realizadas. Associado às retransmissões, será explorada tanto a capacidade quanto a porcentagem de quadros entregues pelos dispositivos com interferência na comunicação entre dispositivo final e ponto de acesso.

3.2.4 Testes em Campo

Serão realizados testes práticos em campo pela implementação do algoritmo em uma aplicação executada em ambiente real, reproduzindo como ocorreriam as interferências do meio durante o rastreamento de objetos em movimento. Pretende-se verificar dessa forma como o gerenciamento de retransmissões se comportou e auxiliou na garantia da entrega dos dados gerados pelo dispositivo final. Durante a transmissão das mensagens, foram coletados dados de localização geográfica, em tempo real, como latitude e longitude, também se armazenou o tempo de recebimento de cada uma das mensagens e os valores referentes ao RSSI e a Relação Sinal-Ruído, do Inglês *Signal-to-noise ratio* (SNR). O RSSI e SNR indicam a potência do sinal recebido e a razão entre essa potência e os nível de ruídos presentes no meio respectivamente (SPHINX, 2018).

4 RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados obtidos a partir da execução do algoritmo ALR-LoRaWAN proposto nesse trabalho. A seção está dividida em 3 cenários. O primeiro realizado apenas com o algoritmo de retransmissão proposto. Em seguida uma comparação prática com o algoritmo CR-LoRaWAN, concebido por Almeida *et al.* (2021). A seção finaliza com a utilização do ALR-LoRaWAN em um teste realizado em campo. Cada seção descreve o número de execuções realizadas e os diferentes parâmetros utilizados. Em seguida são expostos os resultados obtidos, como a relação entre o número de quadros perdidos e as solicitações de retransmissão realizadas.

4.1 RESULTADOS PRÁTICOS

Neste primeiro cenário, pretende-se avaliar o algoritmo proposto de forma detalhada, sendo mostrado quanto tempo se levou para que os quadros de fato fossem recebidos. Calcula-se a diferença de tempos em segundos entre o momento de recebimento do quadro atual e de seu antecessor. Foram adotados diferentes valores associados ao fator de retransmissão **N**, a fim de verificar o seu impacto quanto ao tempo que levaria para um quadro perdido ser retransmitido e recebido corretamente.

4.1.1 Ambiente de Realização das Execuções

Em todas as execuções se utilizou apenas um dispositivo final, esse estando localizado em uma residência a 5 Km de distância do ponto de acesso, que estava fixado na UTFPR Campus Toledo. Por conta do relevo local e, principalmente, das obstruções presentes no meio, ocorreram eventuais interrupções de sinal durante o período de transmissão de dados entre o dispositivo final e ponto de acesso. Este cenário é o mesmo encontrado em um ambiente real, onde sensores localizados em áreas remotas se comunicam com o ponto de acesso.

Durante os testes foram realizadas 6 execuções. Cada execução consistiu em enviar coordenadas geográficas pelo dispositivo final, simulando o rastreamento de objetos. Foram enviados 150 quadros de dados com FCNT distintos por execução. A cada execução os quadros foram enviados periodicamente a cada 30 segundos

em transmissões normais ou 10 segundos durante a realização de retransmissões. O intervalo de 30 segundos foi adotado para satisfazer as necessidades de transmissão em tempo real da aplicação de rastreamento executada. Já o período de 10 segundos para retransmissões foi adotado para que fosse possível reenviar até 3 quadros de dados em um período destinado a apenas um quadro quando em transmissão normal.

4.1.2 Resultados Obtidos

A Tabela 2 apresenta as 6 execuções realizadas, considerando diferentes valores para o fator de retransmissão N . Para um valor de N igual a 1 temos a retransmissão de quadros de dados sendo solicitadas logo após a identificação de sua ausência, que ocorre quando se recebe um FCNT fora de sequência. Quando o valor N aumenta para 2, torna-se necessário detectar a ausência de dois quadros de dados que irão compor uma única solicitação de retransmissão *Downlink*. O dispositivo final, por sua vez, após decodificar o quadro de dados, enviará as duas mensagens solicitadas, apresentando um espaçamento de 10 segundos entre cada uma das duas transmissões. A mesma lógica se estende para as execuções 5 e 6, com o valor N igual a 3.

Tabela 2 – Resultado das execuções.

	N	Enviados	Perdidos	Solicitações	Retransmissões
1° Execução	1	150	6	7	6
2° Execução	1	150	7	9	7
3° Execução	2	150	17	14	17
4° Execução	2	150	14	11	14
5° Execução	3	150	8	3	8
6° Execução	3	150	9	6	9

A Tabela 2 apresenta também quantas mensagens, no total, foram enviadas pelos dispositivos, lembrando que todas as execuções se limitaram a 150 mensagens. A coluna *Perdidos* descreve o número de mensagens que não chegaram a ser recebidas pelo ponto de acesso em sua primeira transmissão. A coluna *Solicitações* informa quantas mensagens foram enviadas pelo servidor de aplicação. Dessa forma, observa-se que nem todos os quadros solicitados são de fato retransmitidos, podendo haver interferência na mensagem *Downlink* enviada pelo ponto de acesso ou *Uplink* reenviada pelo dispositivo final. Por exemplo, se uma dada mensagem for perdida durante a sua

transmissão, o servidor de aplicação vai realizar uma solicitação para retransmissão. Porém, como essa mensagem utiliza o mesmo enlace para comunicação, ela pode ser perdida durante a sua transmissão, sendo necessário o envio de uma nova solicitação pelo servidor de aplicação. Já a coluna *Retransmissões* descreve quantos dos quadros de dados perdidos foram retransmitidos e recebidos com sucesso. Dessa forma, percebe-se que as colunas *Perdidos* e *Retransmissões* apresentam os mesmos valores, o que significa que todos os quadros de dados que foram perdidos foram recebidos através de retransmissões. Percebe-se que em alguns casos é necessário realizar mais de uma solicitação de retransmissão para o mesmo quadro de dados, como na primeira e segunda execução.

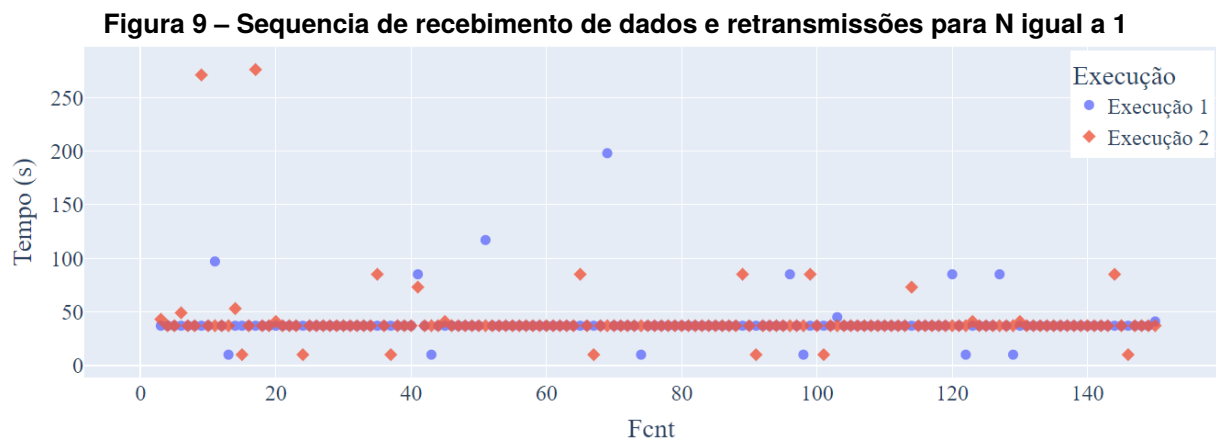
Ao se enviar uma mensagem de retransmissão, por se utilizar o mesmo enlace sem fio durante a comunicação, não se tem a garantia de que de fato a solicitação chegará no dispositivo ou a retransmissão será recebida pelo *Gateway*. Dessa forma a Tabela 3 apresenta em cada execução quantas solicitações de retransmissão foram necessárias para que cada quadro fosse retransmitido e recebido corretamente. Dessa forma, na execução de número 1 foram recebidos 5 quadros de dados pela primeira solicitação de retransmissão enviada pelo servidor e 1 durante a segunda solicitação, sendo esse, interferido em sua primeira tentativa de retransmissão. Já, na segunda execução, foram recebidos 5 quadros de dados na primeira tentativa e 2 só foram recebidos durante a segunda solicitação transmitida. A lógica se estende para todas as outras execuções.

	Execuções					
	Execução 1	Execução 2	Execução 3	Execução 4	Execução 5	Execução 6
1° Solicitação	5	5	13	9	7	5
2° Solicitação	1	2	2	5	1	1
3° Solicitação	-	-	-	-	-	3
4° Solicitação	-	-	-	-	-	-
5° Solicitação	-	-	2	-	-	-

Tabela 3 – Número de quadros recebidos por solicitações realizadas.

Com isso, é possível verificar que de fato a maioria das retransmissões estão ocorrendo logo após a sua solicitação de retransmissão. Porém, na execução 3 houve alguns quadros sendo recebidos durante a quinta solicitação e na 6 durante a terceira. Já durante a execução 4 vários quadros foram recebidos apenas durante a primeira e segunda solicitação.

A seguir são apresentadas as sequências de recebimento de dados para cada uma das execuções. Na Figura 9 temos a sequência de quadros referentes à primeira e à segunda execuções. Para um fator de retransmissão N igual a 1, a maioria dos quadros retransmitidos são recebidos em até 90 segundos de diferença. Mas, como nem todos os quadros de dados são recebidos na primeira tentativa de retransmissão, alguns apresentam um intervalo maior. O quadro com o maior intervalo entre transmissão e recepção atingiu 276 segundos, por se tratar de uma retransmissão que precisou de mais de uma solicitação de retransmissão para ser recebida corretamente. Também é possível notar alguns quadros sendo recebidos com um intervalo menor do que 30 segundos, sendo referente as retransmissões realizadas, com um período de 10 segundos como mencionado anteriormente.



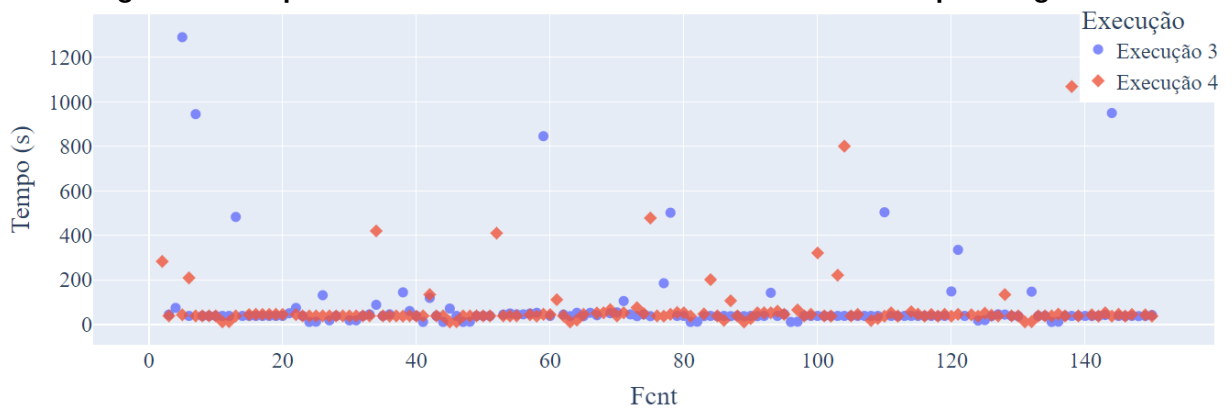
Fonte: autoria própria.

Já para um fator de retransmissão igual a 2 percebe-se que o atraso de recebimento se apresenta mais elevado, o que é esperado, uma vez que, para se realizar a solicitação de retransmissão é necessário que pelo menos dois quadros de dados tenham sido perdidos. Como demonstra a Figura 10, a maioria dos dados são recebidos em até 500 segundos de diferença, enquanto o pior caso para a execução 3 chega a 1290 segundos e para a execução 4 atinge 1050 segundos.

Na última execução mostrada na Figura 11, considerando N igual a 3, a maioria das retransmissões ocorrem em até 1500 segundos, porém podendo chegar, no pior dos casos, a 3697 segundos. O elevado tempo observado nessas execuções se deve aos mesmos motivos explicados quando se tinha um valor N igual a 2.

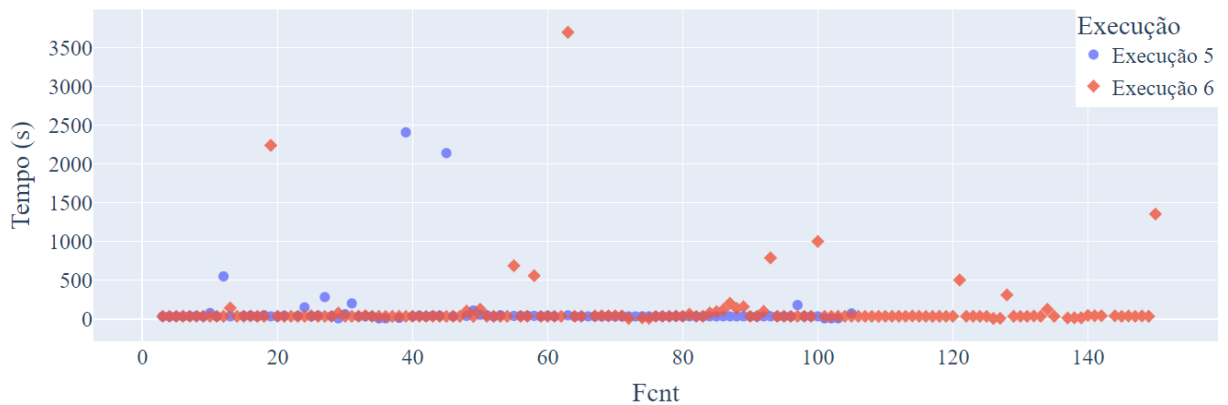
Com isso, verifica-se que em todas as execuções a maioria dos quadros são recebidos com um intervalo de 30 a 40 segundos. Após a solicitação de retransmissão

Figura 10 – Sequencia de recebimento de dados e retransmissões para N igual a 2



Fonte: autoria própria.

Figura 11 – Sequencia de recebimento de dados e retransmissões para N igual a 3



Fonte: autoria própria.

o próximo quadro de dados, que utiliza a janela de recebimento do dispositivo final, é recebido com uma diferença de 10 segundos, como mencionado anteriormente pela abordagem adotada.

A partir das execuções realizadas, verifica-se como a utilização de diferentes fatores de retransmissão N podem influenciar no tempo para recebimento de um quadro de dados inicialmente perdido. Para um fator N igual a 1, o quadro com maior atraso para recebimento foi de 276 segundos; já para N igual a 2 o pior caso foi de 1290 segundos, e para o seu valor em 3 esse tempo aumentou para 3697 segundos. Assim, fica evidente que em aplicações que necessitam de uma troca de mensagens em tempo real um valor de N igual a 1 passa a ser mais adequado. Enquanto que, ao lidar com aplicações em que o momento de recebimento não é prioridade, mas sim o fato de que todos os dados gerados pelo dispositivo sejam recebidos, um valor de N igual a 2 ou 3 pode ser mais adequado.

Ao se utilizar um valor de N maior ocorre uma redução no envio das mensagens

de solicitações de retransmissão *downlink*, o que resulta em uma menor utilização dos pontos de acesso. A redução na utilização dos *Gateways* diminui a possibilidade de congestionamento dos mesmos, passando a ser um ponto positivo da proposta, uma vez que, a utilização excessiva dos pontos de acesso pode degradar o desempenho geral da rede, como destaca o trabalho de Marais, Abu-Mahfouz e Hancke (2020).

4.2 RESULTADOS COMPARATIVOS

Neste segundo cenário, busca-se comparar a eficiência dos algoritmos ALR-LoRaWAN desenvolvido e CR-LoRaWAN proposto por Almeida *et al.* (2021). Este cenário visa investigar o desempenho de cada um dos algoritmos mediante a sua aplicação em um contexto prático. Para tanto, 4 execuções distintas são realizadas durante os testes de cada algoritmo, com envio de 150 quadros de dados por execução. É levantado, em cada um dos testes, quantos quadros de dados foram recebidos corretamente durante o primeiro envio e quantos sofreram interferência do meio de propagação do sinal. Observando também quantos dos dados recebidos e perdidos durante o primeiro envio foram retransmitidos pelo dispositivo final. Para analisar o seu impacto quanto a utilização do *Gateway*, foi observado quantas mensagens de *downlink* o ponto de acesso precisou enviar em cada execução.

4.2.1 Ambiente de Realização das Execuções

Os testes práticos foram realizados utilizando dois dispositivos TTGO Tbeam 3.1.1, um executando o ALR-LoRaWAN e o outro executando o algoritmo CR-LoRaWAN. Ambos os dispositivos simularam uma aplicação de rastreamento de objetos pela coleta de coordenadas geográficas durante as execuções dos algoritmos. A comunicação foi realizada com um ponto de acesso localizado a 5 Km de distância da posição dos dispositivos finais. Como no teste prático realizado na sessão 4.1, o relevo local e os diversos níveis de sombreamento de sinal ocasionou interrupções na transmissão de alguns quadros de dados enviados.

Foram realizadas 4 execuções com cada um dos algoritmos, totalizando 8 execuções ao final da rodada de testes. Durante cada uma das execuções foram enviados 150 quadros de dados, utilizando um intervalo de 30 segundos entre transmissão para

ambos os dispositivos configurados, o critério para o período adotado segue o mesmo fundamento explicado na sessão 4.1.1.

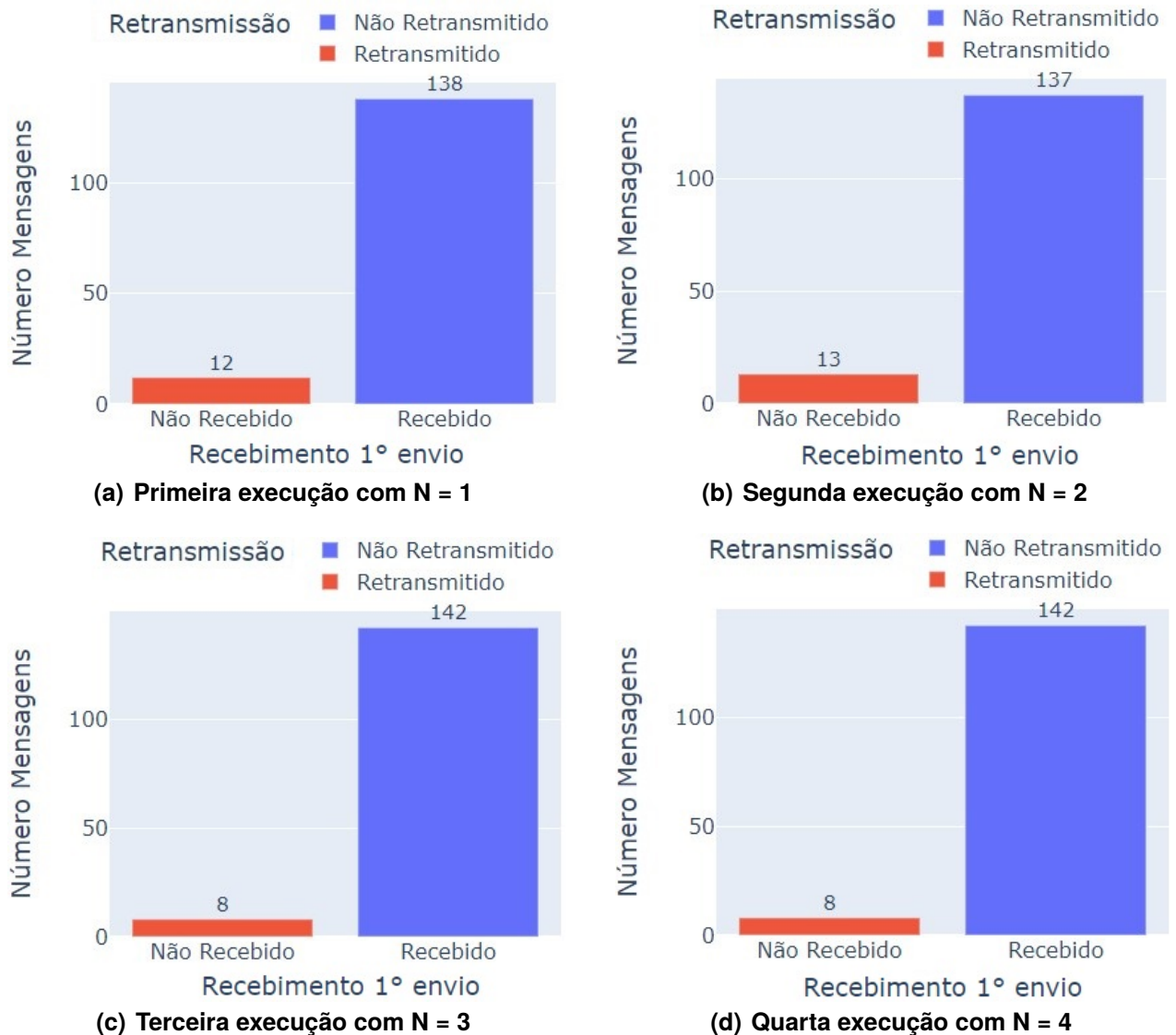
4.2.2 Resultados Obtidos

A Figura 12 apresenta os resultados obtidos durante as 4 execução realizadas com o algoritmo ALR-LoRaWAN proposto nesse trabalho. Os resultados exibem quantos quadros de dados foram recebidos ou não durante o primeiro envio, e desses quadros quantas retransmissões foram realizadas. As execuções foram realizadas utilizando diferentes fatores de retransmissão **N**, indo de 1 até 4, para cada execução respectivamente. Na primeira execução foi considerado um fator **N** igual a 1, nesse caso dos 150 quadros de dados enviados, 138 foram recebidos em sua primeira transmissão, enquanto 12 dos 150 foram perdidos e solicitados retransmissão pelo servidor de aplicação. Verifica-se que, após a retransmissão dos 12 quadros de dados que haviam sido interferidos, houve a garantia de que todos os dados gerados pelo dispositivo chegaram de fato ao servidor de aplicação.

Durante a segunda execução, em que se utilizou um fator **N** igual a 2, foram perdidas 13 mensagens enviadas pelo dispositivo, todas as 13 foram recebidas após as solicitações de retransmissão realizadas pelo servidor de aplicação. Já para a terceira e quarta execução, em que se utilizou um fator de retransmissão **N** igual a 3 e 4 respectivamente, houve apenas a interferência e 8 quadros de dados, esses que foram retransmitidos durante a execução da aplicação.

Já nas execuções obtidas pelo dispositivo com implementação do algoritmo CR-LoRaWAN, verifica-se, na primeira execução, dos 150 quadros enviados 125 sendo recebido corretamente pelo ponto de acesso durante o primeiro envio, enquanto 25 foram perdidos. Apesar dos 125 quadros serem recebidos corretamente 16 desses quadros foram retransmitidos, isso acontece por conta da abordagem adotada pelo CR-LoRaWAN durante a tratativa de suas retransmissões, em que considera sempre o quinto quadro de dado como sendo um quadro de confirmação que engloba as últimas quatro mensagens enviadas, como descrito por Almeida *et al.* (2021) e mencionado na sessão 2.6. Enquanto os 25 quadros de dados que foram perdidos durante o primeiro envio, apenas 9 deles são retransmitidos pelo dispositivo final, perdendo com isso 16 quadros de dados que não chegaram a ser recebidos pelo ponto de acesso e servidor

Figura 12 – Resultados obtidos pelas execuções realizadas com o ALR-LoRaWAN



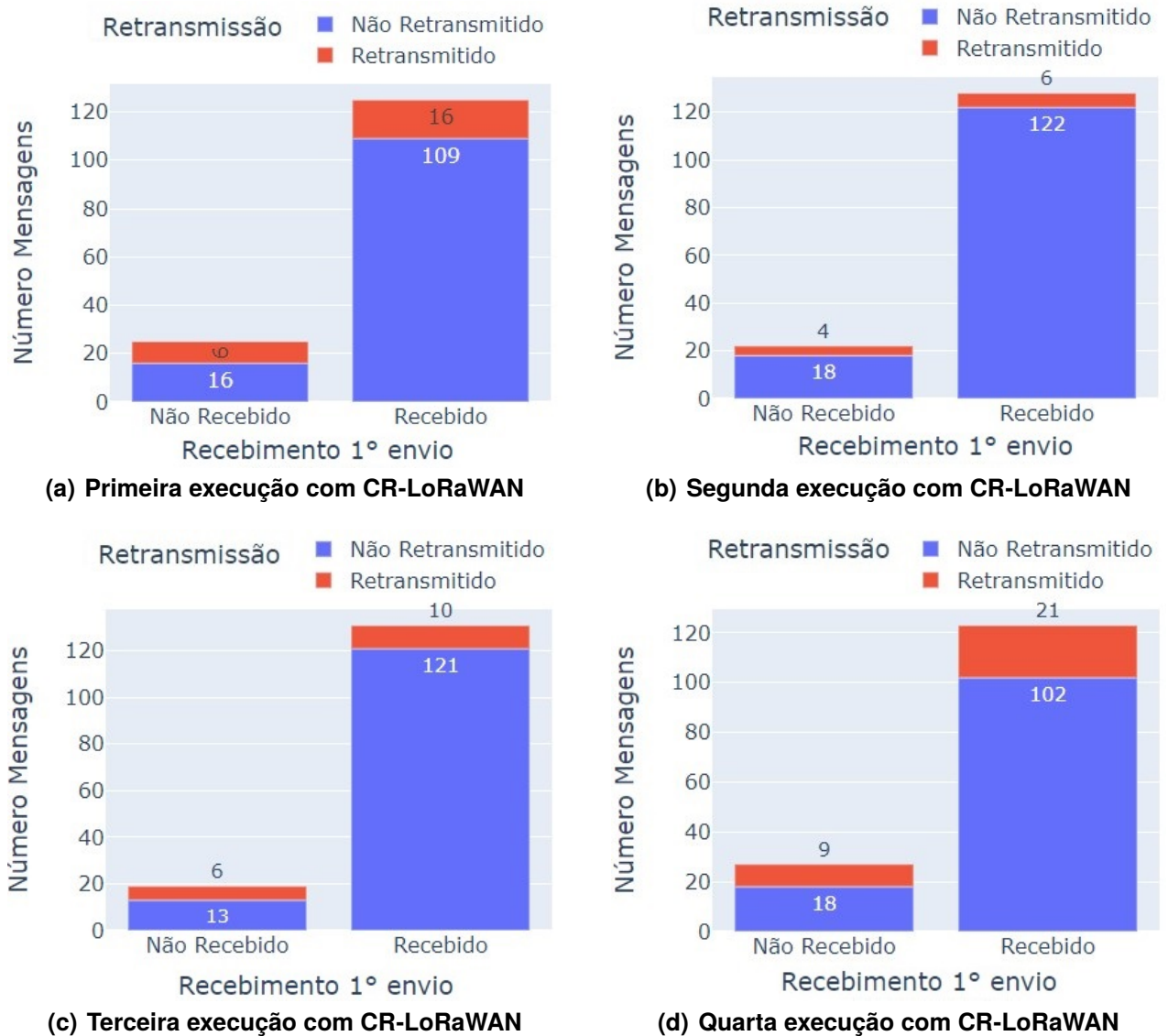
Fonte: autoria própria (2022).

de rede. Dessa forma apenas 36% dos quadros perdidos foram retransmitidos com sucesso.

Durante a segunda execução, dos 150 quadros de dados enviados, 128 foram recebidos corretamente pelo ponto de acesso no primeiro envio, desses, 6 foram retransmitidos. Já dos 22 que não foram recebidos, apenas 4 foram retransmitidos durante a execução, ocasionando apenas a retransmissão de 18,2% dos quadros perdidos. Na terceira execução essa porcentagem aumentou para 31,6%, sendo retransmitidos 6 dos 19 quadros perdidos, porém aumentou também as retransmissões de quadros de dados que já haviam sido recebidos, retransmitindo 10 dos 131 quadros recebidos corretamente. A última execução apresentou o maior número de transmissão de quadros

duplicados, das 123 mensagens recebidas corretamente, 21 delas foram retransmitidas, enquanto das 27 mensagens não recebidas 9 foram retransmitidas, resultando no recebimento de 33,3% das mensagens perdidas.

Figura 13 – Resultados obtidos pelas execuções realizadas com o CR-LoRaWAN



Fonte: autoria própria (2022).

A partir das execuções realizadas foi possível obter quantas mensagens *down-link* foram enviadas pelo ponto de acesso, seja por solicitação do servidor de aplicação, como ocorre no algoritmo ALR-LoRaWAN, ou por ACK, utilizando o algoritmo CR-LoRaWAN. Essa métrica é obtida para entender o impacto de cada uma das abordagens quanto a utilização dos pontos de acesso e potenciais congestionamentos que podem estar causando, fator crucial para o desempenho de redes LoRaWAN, como mencionado por Marais, Abu-Mahfouz e Hancke (2020).

A Tabela 4 apresenta o número de mensagens *downlink* enviadas durante as diferentes execuções pelos dois algoritmos utilizados. Durante a primeira execução o algoritmo ALR-LoRaWAN enviou 31 mensagens de *downlink*, enquanto o CR-LoRaWAN enviou 30. Porém na segunda execução, apesar do algoritmo CR-LoRaWAN enviar o mesmo número de mensagens *downlink*, o algoritmo ALR-LoRaWAN reduziu o seu número de envios para 10. Isso está atrelado ao fato da segunda execução ter utilizado um fator de retransmissão **N** igual a 2, dessa forma todas as solicitações de retransmissão encaminhadas pelo servidor de aplicação sempre compõem dois valores de FCNT, solicitando 20 retransmissões pelas 10 mensagens *downlink* enviadas. Apesar do ALR-LoRaWAN ter um menor número de quadros perdidos durante a terceira e quarta execução, ambas com 8 mensagens, é possível verificar o impacto do valor de **N** ao apresentar diferentes números de mensagens *downlink* entre ambas as execuções, reduzindo de 6 transmissões na execução 3 para 3 na execução 4, onde se utiliza um valor **N** igual a 4. Durante as execuções 2 e 3 o algoritmo CR-LoRaWAN manteve o envio de 30 mensagens *downlink*, valor que aumentou para 32 durante a quarta execução.

		Execuções			
		Execução 1	Execução 2	Execução 3	Execução 4
ALR-LoRaWAN	Perdidos	12	13	8	8
	<i>Downlinks</i>	31	10	6	3
CR-LoRaWAN	Perdidos	25	22	20	19
	<i>Downlinks</i>	30	30	30	32

Tabela 4 – Comparação entre mensagens Downlink e dados retransmitidos.

Dessa forma se evidencia que com a utilização do algoritmo ALR-LoRaWAN houve a garantia na entrega de todas as mensagens perdidas durante a comunicação, enquanto com o CR-LoRaWAN a porcentagem de quadros perdidos que foram retransmitidos esteve entre 18% a 36%. Também, quanto a utilização do ponto de acesso, apesar dos dois algoritmos desempenharem de forma semelhante durante a primeira execução, apresentando 31 e 30 transmissões *downlink*, as demais execuções do algoritmo ALR-LoRaWAN reduziram a utilização do *Gateway* significativamente, principalmente pela abordagem adotada com a variação do fator de retransmissão **N**, onde o CR-LoRaWAN se manteve em torno de 30 mensagens *downlink* por execução.

4.3 RESULTADOS EM CAMPO

O terceiro cenário foi realizado a partir da obtenção de resultados mediante a implementação do algoritmo ALR-LoRaWAN em um dispositivo de rastreamento. Reproduzindo um ambiente de execução real de uma aplicação em redes LoRaWAN, como mencionado por Camargo, E. T., Spanhol, F. A. e Souza com o rastreamento de caminhos da coleta seletiva de lixo. A partir do levantamento desses resultados busca-se verificar como se comporta a utilização do algoritmo, analisando os quadros de dados perdidos e quando os mesmos foram retransmitidos. Armazenando também durante todo o percurso a intensidade do sinal recebido pelo ponto de acesso, afim de verificar zonas com maior interferência de sinal.

4.3.1 Ambiente de Realização do Teste

Para a coleta de coordenadas geográficas foi utilizado um dispositivo TTGO TBeam, pois o mesmo é integrado com uma antena GPS e um chip ESP32 para comunicação com redes LoRaWAN, como mencionado na Seção 3.1.1. Os testes foram realizados a partir da locomoção nos perímetros da Universidade Tecnológica Federal do Paraná, campus Toledo, onde está presente o ponto de acesso utilizado para comunicação, sendo esse um *Gateway* Radioenge, Seção 3.1.2. Durante a trajetória o dispositivo final transmitiu os dados de latitude e longitude coletados, utilizando um intervalo de aproximadamente 30 segundos entre cada envio de dados. Após a realização do teste foram enviados 95 quadros de dados. A execução foi realizada considerando um fator de retransmissão **N** igual a 1, sendo então solicitada a retransmissão dos dados pelo servidor de aplicação logo após a identificação de sua obstrução.

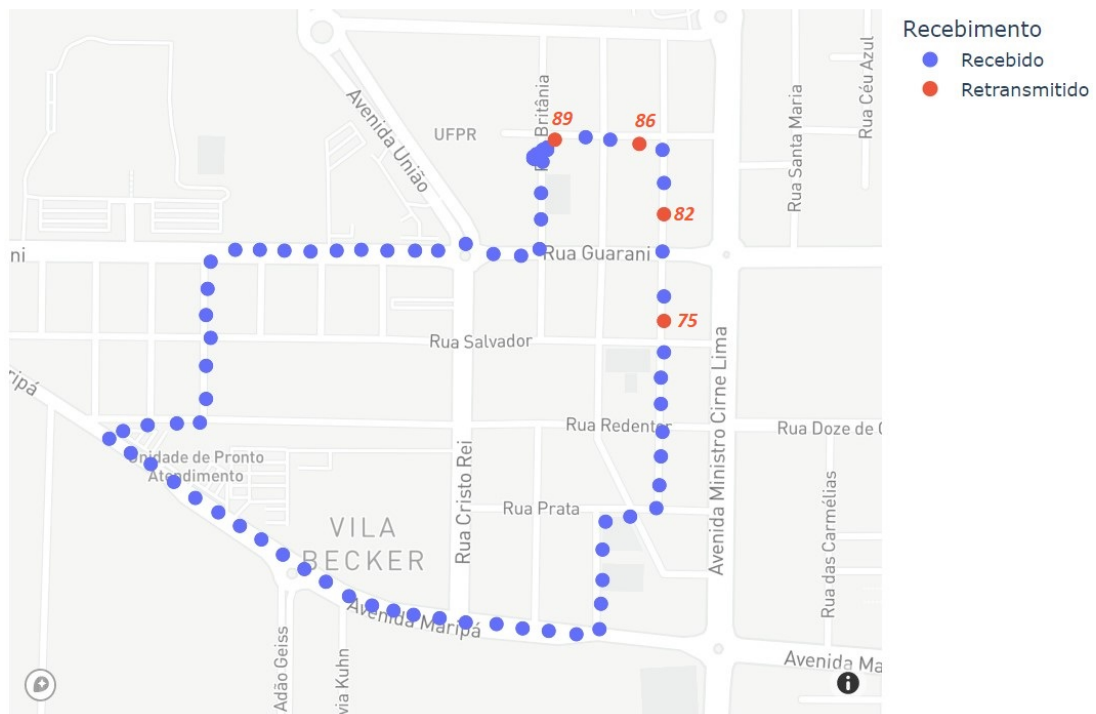
4.3.2 Resultados Obtidos

Na Figura 14 é apresentada a situação dos dados transmitidos durante a coleta de resultados em campo. Podemos ver que apenas 4 dos 95 quadros de dados foram perdidos durante a comunicação. Verifica-se que os dados perdidos foram retransmitidos após as solicitações enviadas pelo servidor de aplicação, durante o gerenciamento de recebimento das mensagens enviadas. Dessa forma percebe-se que os quadros com

FCNT igual a 75, 82, 86 e 89 foram interferidos durante a transmissão e retransmitidos com sucesso.

É possível notar durante as mensagens enviadas, por exemplo entre os quadros 75 e 82, o recebimento de apenas dois quadros de rastreamento, enquanto eles se espaçam entre si por 5 valores de FCNT. A falta de mensagens com coordenadas se deve ao fato do mecanismo de Taxa de Dados Adaptável, do Inglês *Adaptive Data Rate* (ADR) utilizado em redes LoRaWAN. O ADR é utilizado pelo ponto de acesso para indicar ao dispositivo final a potência e taxa de dados para envio das mensagens, afim de otimizar o consumo energético dos dispositivos. Dessa forma algumas mensagens são encaminhadas apenas para funcionamento do mecanismo, sem a presença de um *payload* (ALLIANCE, 2017).

Figura 14 – Mapa com a situação de recebimento dos dados



Fonte: autoria própria.

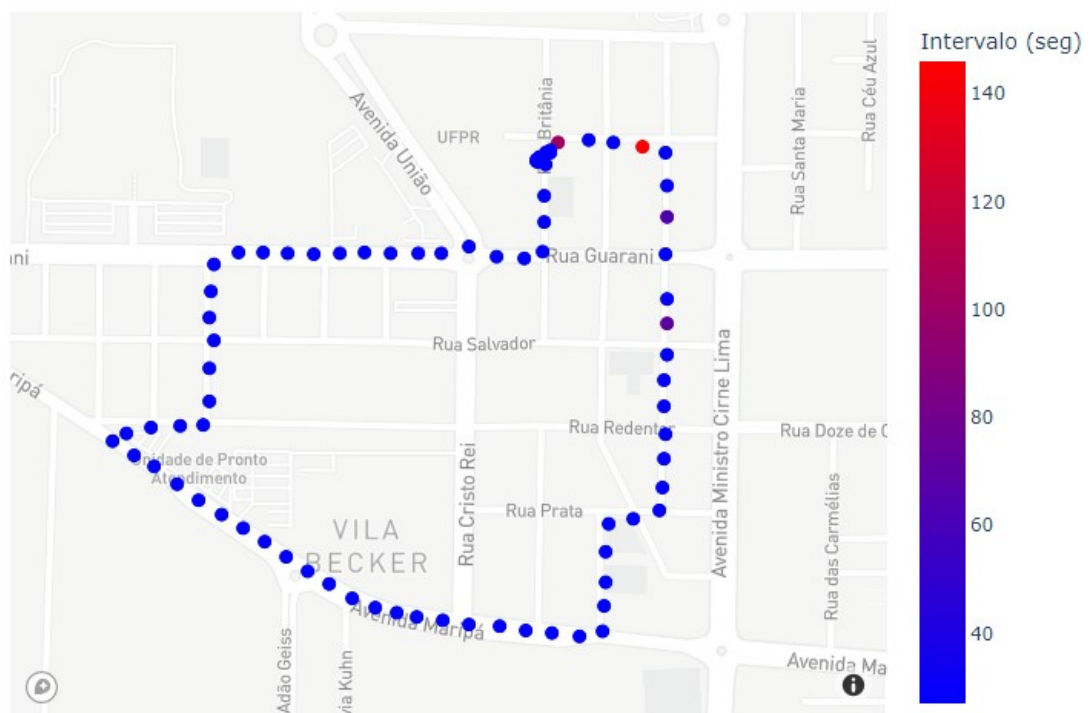
A Tabela 5 mostra em detalhe os valores recebidos para os quadros de dados que sofreram interferência e foram retransmitidos. Enquanto a Figura 15 apresenta o intervalo de tempo, em segundos, entre o recebimento de todos os dados durante a comunicação.

Como configurado no dispositivo final, o envio dos dados se manteve com um intervalo de aproximadamente 30 segundos. Já nos pontos que se tornou necessário o reenvio dos dados verifica-se um aumento nesse intervalo, uma vez que os mesmos

Tabela 5 – Resultado das execuções.

FCNT	Latitude	Longitude	Tempo (Seg.)	SNR (dB)	RSSI (dBm)
75	-24.73063914362426	-53.76066647533575	71 s	9	-90
82	-24.729287310200178	-53.76066647533575	64 s	8	-102
86	-24.728396816754156	-53.76100979811012	146 s	4.2	-108
89	-24.72834317257066	-53.76218997014702	102 s	6.5	-108

foram perdidos durante a transmissão padrão e foi necessário aguardar o recebimento de uma mensagem correta para o servidor solicitar retransmissão.

Figura 15 – Mapa com o intervalo de recebimento dos dados

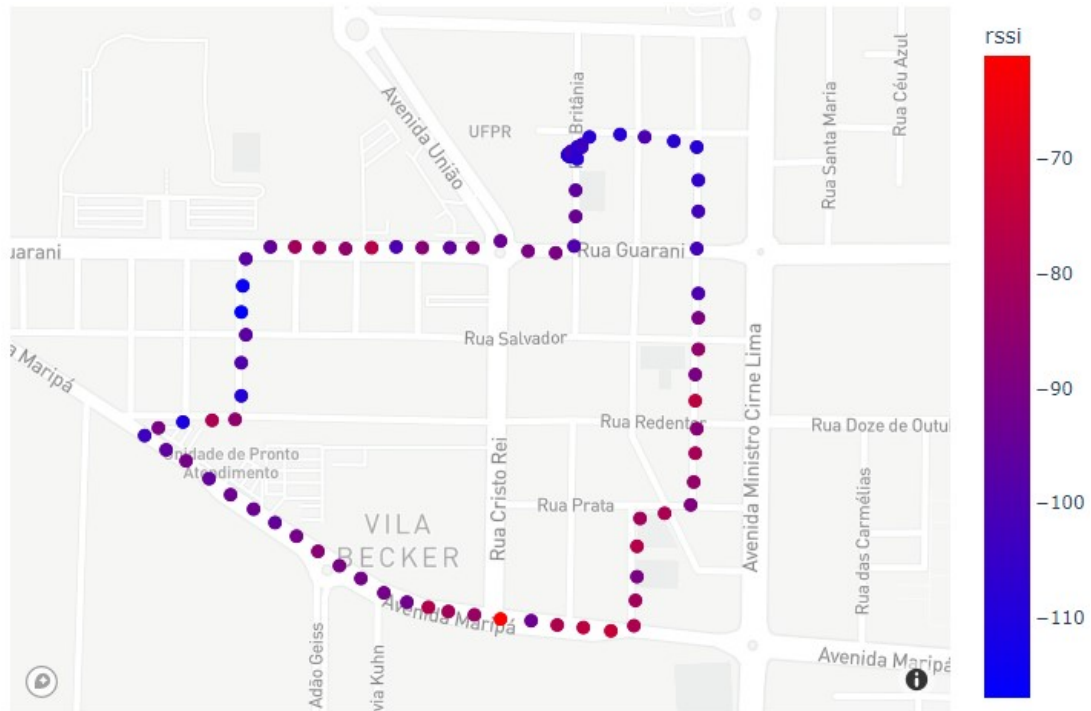
Fonte: autoria própria.

Como mostra a Tabela 5 o tempo para recebimento a partir das retransmissões foi de 71 segundos para o quadro 75, 64 segundos para o 82, 146 segundos para o 86 e 102 segundos para o 89. Dessa forma o pior caso de retransmissão, com 142 segundos, levou aproximadamente 5 vezes o intervalo padrão para o seu recebimento. Já o menor intervalo, de 64 segundos, levou aproximadamente o dobro do tempo padrão de 30 segundos, sendo dessa forma enviada a solicitação de retransmissão pelo servidor de aplicação logo após o próximo dado recebido corretamente após a interferência.

Na Figura 16 é possível visualizar o valor referente ao RSSI recebido por cada dado. Como explica a Seção 2.4.1 esse valor expressa a potência do sinal recebido, onde um sinal fraco apresenta um valor de RSSI próximo a -120dBm. Já valores próximos a

-30dBm caracterizam um sinal forte (SPHINX, 2018). Dessa forma percebe-se regiões com coloração azul representando sinais fracos, como é o caso das posições em que os dados foram interferidos, como os pontos referentes aos FCNTs 75, 82, 86 e 89 apresentados na Figura 14.

Figura 16 – Mapa com o RSSI de recebimento dos dados

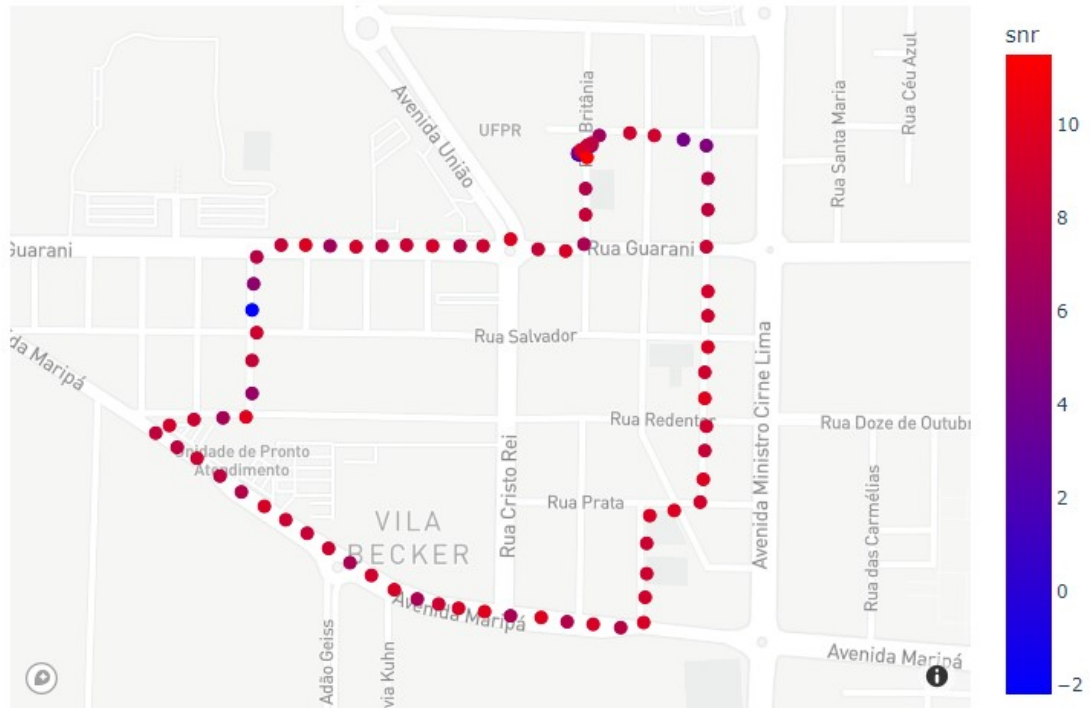


Fonte: autoria própria.

Já o SNR de cada dado recebido é apresentada na Figura 17. Conforme exposto na Seção 2.4.1 esse valor determina a qualidade do sinal recebido mediante aos diferentes ruídos presentes no meio de propagação. Como mencionado por Sphinx (2018) a tecnologia LoRa opera a partir de valores para SNR variando de 10dB a -20dB. Para valores positivos de SNR, em que a potência do sinal recebido é maior que a potência do ruído de fundo presente no meio, o sinal recebido pode ser demodulado sem acarretar corrompimento no dado transmitido. Como apresentado na Figura 17, durante a maior parte da transmissão dos dados essa relação esteve perto de 10dB, expressando uma baixa interferência do ruído presente no meio. Já na Tabela 5 podemos ver que o menor valor registrado para as retransmissões foi de 4.2dB durante o recebimento do dado referente ao FCNT 86.

A partir dos resultados obtidos é evidenciado como ocorre a comunicação de uma aplicação em redes LoRaWAN com a utilização de um algoritmo de retransmissão, analisando durante a comunicação locais com maior interferência do sinal, sendo

Figura 17 – Mapa com o SNR de recebimento dos dados



Fonte: autoria própria.

essas regiões que ocorreram a perda dos dados transmitidos. Dessa forma o algoritmo proposto se mostrou adequado para tolerância a falhas, contornando a perda de sinais durante a comunicação, sendo responsável, ao final da execução em campo, por solicitar a retransmissão dos 4 dados que haviam sido interferidos em um baixo intervalo de tempo, chegando no pior dos casos a 146 segundos.

5 CONCLUSÕES

Com a proposta de um algoritmo para lidar com retransmissão de dados em redes LoRaWAN, foi possível contornar os problemas de interferência de sinal, uma falha comum em redes sem fio. O algoritmo ALR-LoRaWAN foi implementado para gerenciamento de retransmissão em nível de aplicação. Sendo desenvolvido todo o gerenciamento das mensagens em um servidor de aplicação com integração ao protocolo de comunicação MQTT. A proposta apresentada foi aceita e publicada no XXIII Workshop de Testes e Tolerância a Falhas (WTF), realizado em 2022 pela Sociedade Brasileira de Computação (SBC) e pelo Laboratório de Redes de Computadores (LARC) (PASTÓRIO, J. P.; SOUZA; CAMARGO, E. T., 2022).

Os experimentos trouxeram diferentes visões para validação da proposta desenvolvida. Analisando a utilização de diferentes fatores de retransmissão N na tolerância a falhas e no intervalo de recebimento das retransmissões. Constatando que, para um valor de N menor são realizadas mais solicitações de retransmissão, assim como o atraso de recebimento das mensagens perdidas se torna menor. Sendo assim um valor de N menor mais adequado para aplicações que necessitam de troca de mensagens em tempo real, como é o caso de rastreamentos de objetos móveis.

Foi possível comparar o algoritmo proposto com o algoritmo CR-LoRaWAN, verificando em um cenário prático o impacto de cada algoritmo na tolerância a falhas e no número de solicitações de retransmissão enviadas. Pelos resultados obtidos o ALR-LoRaWAN conseguiu contornar todas as perdas de sinais ocorridas, enquanto o algoritmo CR-LoRaWAN contornou apenas 18% a 36% das falhas. Se mostrando assim o ALR-LoRaWAN mais adequado para aplicações que necessitam de garantia na entrega de suas mensagens.

Com o emprego do algoritmo em um exemplo prático em ambiente real, reproduzindo o rastreamento de objetos em movimento, se chegou em um aumento significativo na confiabilidade de entrega das mensagens, uma vez que, após as retransmissões, 100% dos dados gerados pelo dispositivo foram recebidos.

Pelos resultados obtidos conclui-se que os objetivos desse trabalho foram alcançados, uma vez que, pelos testes realizados com o algoritmo ALR-LoRaWAN foi possível contornar as perdas de sinais presentes em redes LoRaWAN, aumentando a confiabilidade na entrega das mensagens.

5.1 TRABALHOS FUTUROS

Em trabalhos futuros, se torna importante realizar a avaliação do desempenho energético da abordagem adotada, na medida que, ao serem solicitadas retransmissões em um maior número de quadros de dados, são necessárias decodificações em mais janelas de recebimentos de dados dos dispositivos, os mesmos precisando realizar o processamento de retransmissão desses pacotes.

Outra possibilidade está em desenvolver um estudo aprofundado referente ao impacto dos diferentes valores de retransmissão N no congestionamento dos pontos de acesso, além dos testados durante os resultados apresentados. A importância desse estudo está no fato de que, como mencionado por Marais, Abu-Mahfouz e Hancke (2020), a utilização excessiva de mensagens de *downlink* pode afetar negativamente o desempenho da rede.

Se torna uma alternativa considerável a avaliação do algoritmo ALR-LoRaWAN em diferentes aplicações, como também passando a considerar a presença de um maior número de pontos de acesso e dispositivos finais presentes na rede durante a troca de mensagens.

REFERÊNCIAS

- ABDELFADEEL, K. Q. *et al.* Fine-Grained Scheduling for Reliable and Energy-Efficient Data Collection in LoRaWAN. **IEEE Internet of Things Journal**, v. 7, n. 1, p. 669–683, 2020. DOI: [10.1109/JIOT.2019.2949918](https://doi.org/10.1109/JIOT.2019.2949918).
- ADAMS, D. **The hitchhiker's guide to the galaxy**. 1. American ed. New York: Harmony Books, 1980. ISBN 9780517542095.
- ALLIANCE, L. **LoRaWAN 1.1 specification**. Beaverton, Oregon, Estados Unidos, 2017. Disponível em: https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_specification_v1.1.pdf. Acesso em: 13 mai. 2022.
- ALMEIDA, T. V. O. *et al.* Em Busca de um Algoritmo de Retransmissão em RedesLoRaWAN. **V Workshop de Computação Urbana**, 2021.
- CAMARGO, E. T.; SPANHOL, F. A.; SOUZA, Á. R. C. Deployment of a LoRaWAN network and evaluation of tracking devices in the context of smart cities. **J. Internet Serv. Appl.**, v. 12, n. 1, p. 8, 2021. DOI: [10.1186/s13174-021-00138-7](https://doi.org/10.1186/s13174-021-00138-7). Disponível em: <https://doi.org/10.1186/s13174-021-00138-7>.
- CAPUZZO, M.; MAGRIN, D.; ZANELLA, A. Confirmed Traffic in LoRaWAN: Pitfalls and Countermeasures. **17th Annual Mediterranean Ad Hoc Networking Workshop**, Capri Island, Italy, p. 87–93, 2018.
- CHAUDHARI, B. S.; ZENNARO, M. **LPWAN Technologies for IoT and M2M Applications**. , Academic Press, 2020.
- FERREIRA, A. E. *et al.* A study of the LoRa signal propagation in forest, urban, and suburban environments. **Annals of Telecommunications**, 2020. DOI: <https://doi.org/10.1007/s12243-020-00789-w>.
- FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. São Paulo SP: AMGH Editora, 2009.
- FRAGA-LAMAS, P. *et al.* Design and Empirical Validation of a LoRaWAN IoT Smart Irrigation System. **Proceedings**, 2020. DOI: <https://doi.org/10.3390/s20236865>.
- FRAMINGHAM, M. **The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025**). 2019. Disponível em: <https://www.businesswire.com/news/home/20190618005012/en/The-Growth-in-Connected-IoT-Devices-is-Expected-to-Generate-79.4ZB-of-Data-in-2025-According-to-a-New-IDC-Forecast>. Acesso em: 1 jun. 2022.
- GOMEZ, C.; CROWCROFT, J. Multimodal Retransmission Timer for LPWAN. **IEEE Internet of Things Journal**, v. 7, n. 6, p. 4827–4838, 2020. DOI: [10.1109/JIOT.2020.2971609](https://doi.org/10.1109/JIOT.2020.2971609).

HILLAR, G. C. **Hands-On MQTT Programming with Python: Work with the lightweight IoT protocol in Python.** , Packt Publishing Ltd, 2018.

INDUSTRIES, T. T. **What is a LoRaWAN Network Server?** 2020. Disponível em: <https://www.thethingsindustries.com/news/what-lorawan-network-server/>. Acesso em: 1 jun. 2022.

JOHNSTON, S. J. *et al.* City Scale Particulate Matter Monitoring Using LoRaWAN Based Air Quality IoT Devices. **Sensors**, 2019. DOI: <https://doi.org/10.3390/s19010209>.

KHANNA, A.; KAUR, S. Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. **Computers and Electronics in Agriculture**, p. 218–321, 2019.

KRANENBURG, R.; BASSI, A. IoT Challenges. **Communications in Mobile Computing**, v. 1, dez. 2012. DOI: 10.1186/2192-1121-1-9.

KUROSE, J.; ROSS, K. **Redes de Computadores e a Internet.** São Paulo SP: Pearson, 2013.

LAVRIC, A.; POPA, V. Internet of Things and LoRa™ Low-Power Wide-Area Networks: A survey. In: 2017 International Symposium on Signals, Circuits and Systems (ISSCS). Suceava, ROM: [s. n.], 2017. P. 1–5. DOI: 10.1109/ISSCS.2017.8034915.

LILYGO. **LILYGO® TTGO T-Beam V1.1 ESP32 433/868/915/923Mhz WiFi Wireless Bluetooth Module GPS NEO-6M SMA LORA 32 18650 Battery Holder.** 2021. Disponível em: http://www.lilygo.cn/claprod_view.aspx?TypeId=62&Id=1281&FId=t28:62:28. Acesso em: 15 mai. 2022.

LORA ALLIANCE. **LoRaWAN™ 1.1 specification.** Beaverton, Oregon, Estados Unidos, 2017. P. 101. Disponível em: https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf. Acesso em: 12 mai. 2022.

MARAIS, J. M.; ABU-MAHFOUZ, A. M.; HANCKE, G. P. A Survey on the Viability of Confirmed Traffic in a LoRaWAN. **IEEE Access**, v. 8, p. 9296–9311, 2020. DOI: 10.1109/ACCESS.2020.2964909.

MARTÍNEZ, C.; ERAS, L.; DOMÍNGUEZ, F. The Smart Doorbell: A proof-of-concept Implementation of a Bluetooth Mesh Network. In: 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM). Guayaquil, EC: [s. n.], 2018. P. 1–5. DOI: 10.1109/ETCM.2018.8580325.

MCCI, C. **Arduino LoRaWAN MAC in C (LMIC).** MCCI Corporation. 2018. Disponível em: <https://redmine.laas.fr/attachments/download/1505/LMIC-v2.3.pdf>. Acesso em: 5 nov. 2021.

MEKKI, K. *et al.* A comparative study of LPWAN technologies for large-scale IoT deployment. **ICT Express**, 2017.

MONK, S. **Electronics cookbook: practical electronic recipes with Arduino and Raspberry Pi**. First edition. Sebastopol, CA: O'Reilly, 2017. OCLC: ocn953432200. ISBN 9781491953402.

MOORE, T.; MCCI, C. **Arduino-LMIC library ("MCCI LoRaWAN LMIC Library")**. 2018. Disponível em: <https://github.com/mcci-catena/arduino-lmic>. Acesso em: 1 jun. 2022.

PASTÓRIO, A.; RODRIGUES, L.; CAMARGO, E. de. Uma Revisão Sistemática da Literatura Sobre Tolerância a Falhas em Internet das Coisas. In: ANAIS Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais. Evento Online: SBC, 2020. P. 57–64. DOI: 10.5753/sbesc_estendido.2020.13091. Disponível em: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13091.

PASTÓRIO, A.; SPANHOL, F.; CAMARGO, E. Avaliação do Impacto da Temperatura e Umidade em Sensores de Baixo Custo de Material Particulado. In: ANAIS do VI Workshop de Computação Urbana. Fortaleza - CE: SBC, 2022. p.1–14.

PASTÓRIO, J. P.; SOUZA, Á. R. C.; CAMARGO, E. T. Simulação do monitoramento de veículos em uma rede LoRaWAN usando NS-3. **XIX Escola Regional de Redes de Computadores**, 2021.

PASTÓRIO, J. P.; SOUZA, Á. R. C.; CAMARGO, E. T. Um Algoritmo de Gerenciamento de Retransmissão em Nível de Aplicação para redes LoRaWAN. **XXIII Workshop de Testes e Tolerância a Falhas**, 2022.

RADIOENGE. **LoRa Gateway Radioenge**. Radioenge. 2018. Disponível em: <https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html>. Acesso em: 5 nov. 2021.

RAJ, P.; RAMAN, A. C. **The Internet of Things: Enabling technologies, platforms, and use cases**. Boca Raton, FL: CRC Press, 2017.

ROSSATO, J.; SPANHOL, F.; CAMARGO, E. Implantação e Avaliação de uma Rede Sem-Fio de Longo Alcance e Baixa Potência para Cidades Inteligentes. In: ANAIS do IV Workshop de Computação Urbana. Rio de Janeiro: SBC, 2020. P. 192–205. DOI: 10.5753/courb.2020.12363. Disponível em: <https://sol.sbc.org.br/index.php/courb/article/view/12363>.

SANTOS, B. P. *et al.* Internet das coisas: da teoria a prática. **Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, 2016.

SEMTECH. **LoRa and LoRaWAN: A Technical Overview**. Edição: Semtech Corporation. 2019. Disponível em: <https://lora-developers.semtech.com>

/documentation/tech-papers-and-guides/lora-and-lorawan/. Acesso em: 12 mai. 2022.

SENEVIRATNE, P. **Beginning LoRa Radio Networks with Arduino: Build Long Range, Low Power Wireless IoT Networks**. Canada, CA: Apress, 2019.

SHANMUGA SUNDARAM, J. P.; DU, W.; ZHAO, Z. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. **IEEE Communications Surveys Tutorials**, v. 22, n. 1, p. 371–388, 2020. DOI: 10.1109/COMST.2019.2949598.

SINGH, R. K. *et al.* Leveraging LoRaWAN Technology for Precision Agriculture in Greenhouses. **Sensors (Basel, Switzerland)**, 2020. DOI: <https://doi.org/10.3390/s20071827>.

SPHINX. **LoRa documentation master file**. Sphinx Alabaster. 2018. Disponível em: <https://lora.readthedocs.io/en/latest/>. Acesso em: 13 mai. 2022.

ÍNDICE REMISSIVO

ACK, 14, 24, 27

ADR, 49

ALR-LoRaWAN, 15, 36–38, 43, 44, 46–
48, 53, 54

ARQ, 17

CR-LoRaWAN, 14–16, 43, 44, 47, 53

CSS, 20

FCNT, 15, 24

FSK, 20

IDE, 31

IoT, 13, 17

LARC, 53

LoRa, 19–21

LoRaWAN, 21

LPWAN, 13, 17

M2M, 30

MAC, 29

MQTT, 30

PAN, 19

RSSI, 22

RTO, 26

RTT, 14, 26

SBC, 53

SF, 20

SMA, 28

SNR, 37

TTN, 17

UTFPR, i

WTF, 53