UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ANDRÉ FRANCISCO PASTÓRIO

TÉCNICAS DE GEOLOCALIZAÇÃO EM REDES LORAWAN COMO ABORDAGEM DE TOLERÂNCIA A FALHAS EM DISPOSITIVOS DE RASTREAMENTO BASEADOS EM GPS

TRABALHO DE CONCLUSÃO DE CURSO

TOLEDO
2021

ANDRÉ FRANCISCO PASTÓRIO

TÉCNICAS DE GEOLOCALIZAÇÃO EM REDES LORAWAN COMO ABORDAGEM DE TOLERÂNCIA A FALHAS EM DISPOSITIVOS DE RASTREAMENTO BASEADOS EM GPS

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná - UTFPR Campus Toledo, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Edson Tavares de Camargo
Universidade Tecnológica Federal do Paraná

TOLEDO
2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Toledo
Coordenação do Curso de Engenharia Eletrônica

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

Título do Trabalho de Conclusão de Curso Nº 7

Técnicas de Geolocalização em redes LoRaWAN como Abordagem de Tolerância a Falhas em Dispositivos de Rastreamento Baseados em GPS

por

André Francisco Pastório

Esse Trabalho de Conclusão de Curso foi apresentado às **10h00 do dia 09 de julho de 2021** como **requisito parcial** para a obtenção do título de **Bacharel em Engenharia de Computação**. Após deliberação da Banca Examinadora, composta pelos professores abaixo assinados, o trabalho foi considerado **APROVADO**.

Prof. Dr. Álvaro Ricieri Castro e Souza
UTFPR

Prof. Dr. Elder Elisandro Eschemberger
UTFPR

Prof. Dr. Edson Tavares de Camargo
UTFPR

Prof. Dr. Elder Elisandro Eschemberger
UTFPR

O termo de aprovação assinado encontra-se na coordenação do curso

Toledo, 09 de julho de 2021

AGRADECIMENTOS

Agradeço a todos os meus familiares que contribuíram diretamente ou indiretamente para a minha graduação. Obrigado pelo suporte e paciência durante toda a caminhada.

Aos meus amigos Matheus Pabis Esteves e Thiago Vinney Oliveira Almeida pela parceria e companheirismo durante toda a graduação. Vocês foram achados preciosos e foram muito importantes para a minha formação.

A todos os professores pelo conhecimento e conselhos passados. Em especial ao meu orientador Prof. Dr. Edson Tavares de Camargo. Obrigado pela oportunidade e contribuição imensurável na minha formação.

Na vida há tempo para se arriscar e tempo para se ser cauteloso, e um homem sensato sabe qual é a altura certa para cada uma destas coisas.
(Sociedade dos Poetas Mortos, 1989)

RESUMO

PASTÓRIO, André Francisco. **Técnicas de Geolocalização em redes LoRaWAN como Abordagem de Tolerância a Falhas em Dispositivos de Rastreamento Baseados em GPS**. 2021. 48 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Toledo, 2021.

A ampla adoção do conceito de Internet das Coisas (IoT) passa por aumentar a confiabilidade e a disponibilidade das suas aplicações perante falhas. A heterogeneidade dos componentes IoT, aliado a dispositivos limitados computacionalmente e integrados à Internet, impõe desafios adicionais para alcançar a tolerância a falhas. Cidades inteligentes compreendem um cenário IoT em que a cidade está conectada. Redes de longo alcance e baixo consumo de energia são utilizadas em tais aplicações para fornecer conectividade. Nesse contexto, o rastreamento de objetos é uma importante funcionalidade sendo aplicada, por exemplo, para rastrear veículos de uma frota em tempo real pela Internet. O uso de tecnologias de localização comuns, como o GPS, resulta em um elevado consumo de energia. Aliado a isso, o mal funcionamento de um dispositivo GPS pode comprometer a aplicação. Este trabalho investiga como a tolerância a falhas em IoT pode ser eficientemente desenvolvida através de uma revisão sistemática da literatura. Além disso, para mitigar a falha do GPS em dispositivos de rastreamento, este trabalho implementa uma estratégia de geolocalização híbrida em redes LoRaWAN. A estratégia considera que em caso de falha do GPS as coordenadas são estimadas através de algoritmos de geolocalização. A implementação é comparada com uma ferramenta comercial, chamada LoRa Cloud, que permite apenas mil requisições gratuitas diariamente. Um ambiente com três *gateways* LoRaWAN foi montado para realizar a avaliação. Os resultados demonstram erros inferiores a 151 e 126 metros no pior caso considerando a implementação realizada e a ferramenta LoRa Cloud, respectivamente.

Palavras-chave: IoT. Geolocalização. Tolerância a falhas. LoRaWAN. Cidades inteligentes.

ABSTRACT

The wide adaptation of the Internet of Things (IoT) concept involves increasing the reliability and availability of its applications in the event of failure. The heterogeneity of IoT components, combined with computationally limited devices integrated to the Internet, pose additional challenges to achieving fault tolerance. Smart cities comprise an IoT scenario where the city is connected. Long-range, low-power networks are used in such applications to provide connectivity. In this context, object tracking is an important functionality used, for example, to track fleet vehicles in real time over the Internet. The use of common location technologies such as GPS results in high energy consumption. Allied to that, or malfunction of a GPS device can compromise the application. This work investigates how fault tolerance in IoT can be efficiently developed through a systematic literature review. Furthermore, to mitigate a GPS failure in tracking devices, this work implements a hybrid location strategy in LoRaWAN networks. The strategy considers that in case of GPS failure, coordinates are estimated using location algorithms. The implementation is compared to a commercial tool, called LoRa Cloud, which only allows for a thousand daily requests. An environment with three gateways LoRaWAN was set up to perform an assessment. The results show errors below 151 and 126 meters in the worst case considering the implementation performed and the LoRa Cloud tool, respectively.

Keywords: IoT. Geolocation. Fault tolerance. LoRaWAN. Smart Cities.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

6LowPAN	*IPv6 over Low power Wireless Personal Area Networks*
AMQP	*Advanced Message Queuing Protocol*
AoA	*Angle of Arrival*
BLE	*Bluetooth Low Energy*
CEP	*Complex Event Processing*
CH	*cluster head*
CoAP	*Constrained Application Protocol*
CSS	*Chirp Spread Spectrum*
DLT	*Distributed Ledger Technology*
DNS	*Domain Name System*
GPS	*Global Positioning System*
HSR	*High-availability Seamless Redundancy*
IoT	*Internet of Things*
LoRa	*Long Range*
LPWAN	*Low Power Wide Area Network*
MAC	*Medium Access Control*
MQTT	*Message Queuing Telemetry Transport*
NTP	*Network Time Protocol*
ONS	*Object Name Service*
P2P	*Peer-to-Peer*
PoS	*Proof-of-Stake*
PoW	*Proof-of-Work*
QoS	*Quality of Service*
RF	Radiofrequência
RSSF	Redes de Sensores Sem Fio
RSSI	*Received Signal Strength Indication*
RTAP	*Reliable and Trustworthy Agreement Protocol*
SBESC	Simpósio Brasileiro de Engenharia de Sistemas de Computação
SDN	*Software Defined Network*
SENDI	*System for dEtecting and forecasting Natural Disasters based on IoT*
SF	*Spreading Factor*
TDoA	*Time Difference of Arrival*
ToA	*Time of Arrival*
ToF	*Time of Flight*
TTN	*The Things Network*
VANET	*Vehicular Ad-hoc NETwork*

WR	*White Rabbit*
XMPP	*Extensible Messaging and Presence Protocol*

SUMÁRIO

1 INTRODUÇÃO

O mercado global de soluções para usuários finais da Internet das Coisas (IoT) ultrapassou 100 bilhões de dólares em receita de mercado pela primeira vez em 2017 (BAIN, 2018). Há previsão que esse valor crescerá para cerca de 1,6 trilhão de dólares em 2025 (STATISTA, 2020), chegando a com 41 bilhões de dispositivos em 2027 (VXCHNGE, 2020). Este número deve crescer a medida que a conectividade com a Internet se torna uma característica padrão para uma grande variedade de dispositivos eletrônicos.

A IoT compreende domínios de aplicação diversos, como cidades inteligentes (CENTENARO et al., 2016), agricultura (TZOUNIS et al., 2017) e saúde (ISLAM et al., 2015). O objetivo é transformar dispositivos comuns do dia a dia em objetos inteligentes ao equipá-los com recursos de identificação, sensoriamento, processamento, comunicação e conexão com a Internet. Com isso, objetos podem se comunicar entre si e com outros dispositivos e serviços pela Internet para alcançar algum objetivo útil (WHITMORE; AGARWAL; XU, 2015; AL-FUQAHA et al., 2015). Sensores podem coletar informações do ambiente, como temperatura, umidade, posição geográfica, batimentos cardíacos e imagens de ambientes abertos ou fechados. As informações são enviadas pela rede de comunicação, geralmente sem-fio, para a Internet, onde são processadas e armazenadas em um serviço de nuvem. É possível solicitar remotamente aos objetos inteligentes que executem ações e até mesmo coordenem suas tarefas. A IoT pode ser vista, então, como uma extensão da Internet atual fazendo uso de seus protocolos padrões e de sua característica distribuída.

No contexto de cidades inteligentes, em que são compreendidas aplicações nos domínios de transporte público, logística, tráfego inteligente e rastreamento de frotas (KHELIFI et al., 2019). Nessas aplicações tem-se em comum o uso de um sistema de posicionamento global, *Global Positioning System* (GPS), para fornecer as coordenadas geográficas de veículos, pessoas ou objetos (PODEVIJN et al., 2018b). Dessa forma, é possível rastrear um veículo em movimento dentro de uma cidade, armazenar o histórico do seu percurso e, por exemplo, em um caso de caminhões de coleta seletiva de lixo, alertar a população quando o caminhão está próximo de sua residência.

O ponto central de qualquer aplicação de cidades inteligentes é a conectividade, capacidade de transmissão dos dados através de uma rede de comunicação. Em se tratando de cidades inteligentes, se destacam as redes de longo alcance e baixa potência, conhecidas como *Low Power Wide Area Network* (LPWAN) (CENTENARO et al., 2016). Uma LPWAN tem uma taxa de transferência baixa, mas pode cobrir áreas de até 50 quilômetros. Entre as tecnologias e protocolos LPWAN destaca-se o padrão aberto LoRaWAN que opera sobre a tecnologia sem fio *Long Range* (LoRa).

Embora promissora, concretizar a visão da IoT é uma tarefa árdua devido aos muitos desafios que precisam ser enfrentados, como disponibilidade, confiabilidade, mobilidade, de-

sempenho, escalabilidade, interoperabilidade, segurança e gerenciamento (AL-FUQAHA et al., 2015; RAZZAQUE et al., 2016). Entre todos esses desafios, destacam-se a disponibilidade e a confiabilidade, visto que uma falha pode colocar pessoas em perigo, resultar em perda financeira ou danos ambientais (MACEDO; GUEDES; SILVA, 2014), por exemplo. De fato, disponibilidade (*availability*) e confiabilidade (*reliability*) são atributos que se referem a confiança no funcionamento de sistemas, ou dependabilidade (*dependability*) (AVIZIENIS et al., 2004).

Um dos meios de se alcançar a confiança no funcionamento de um sistema é através de técnicas de tolerância a falhas. Estas técnicas visam garantir o funcionamento correto do sistema mesmo na ocorrência de falhas e são geralmente baseadas em redundância, exigindo componentes adicionais ou algoritmos especiais. Embora a tolerância a falhas em IoT venha sendo explorada em diversos trabalhos (NASSER et al., 2017; POWER; KOTONYA, 2018; GROVER; GARIMELLA, 2018), uma questão importante para desenvolver sistemas IoT tolerantes a falhas é conhecer as principais técnicas e soluções capazes de aumentar a confiabilidade e a disponibilidade. Abordagens como redundância e protocolos de consenso são empregadas a várias décadas e tradicionalmente utilizadas em sistemas distribuídos. No entanto, considerando a heterogeneidade e os recursos limitados dos dispositivos (processamento, memória, conectividade, bateria, etc.), é essencial conhecer as soluções que lidam com falhas no contexto de IoT.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é propor uma solução de tolerância a falhas para um sistema de rastreamento IoT baseado em GPS que faz uso de uma rede LoRaWAN no contexto de cidades inteligentes.

1.1.2 OBJETIVOS ESPECÍFICOS

São os objetivos específicos deste trabalho:

- estudar os conceitos de tolerância a falhas e IoT;
- pesquisar as técnicas de tolerância a falhas que são utilizadas em IoT;
- estudar as técnicas de geolocalização em uma rede LoRaWAN;
- implementar uma técnica de tolerância a falhas em dispositivos de rastreamento IoT em um estudo de caso;
- avaliar a eficiência da técnica.

1.2 JUSTIFICATIVA

Geolocalização é uma técnica essencial em sistemas de navegação que permite a identificação ou estimativa da localização geográfica de um objeto (PODEVIJN et al., 2018a).

Mesmo o GPS sendo a tecnologia mais comum, oferecendo localização precisa em tempo real, este consome muita energia, pois o processamento da informação ocorre no dispositivo (MANZONI et al., 2019). Abordagens de localização utilizando tecnologias sem-fio de radio frequência, WiFi, Bluetooth e ZigBee, podem ser empregadas em ambientes internos (LAM; CHEUNG; LEE, 2019). Da mesma forma LPWAN pode ser aplicada para ambientes externos.

Localização é um tópico importante de pesquisa, muito aplicado a navegação e rastreamento (LAM; CHEUNG; LEE, 2019). A localização geográfica de dispositivos IoT aprimora a informação aumentando o seu valor (MANSFIELD; GHITA; AMBROZE, 2017; LI et al., 2018). A tecnologia mais utilizada para este propósito é o GPS (PODEVIJN et al., 2018b). Diversas aplicações IoT se beneficiam desse sistema e podem fornecer serviços baseados em localização como, por exemplo, em áreas de saúde, agropecuária, cidades inteligentes e rastreamento de veículos (KHELIFI et al., 2019). Essas aplicações sofrem com a cobertura de rede e tolerância a falhas. Redes de longo alcance e baixo consumo de energia, LPWAN, podem ser utilizadas para cobrir áreas extensas e oferecem diferentes métodos de localização, com a utilização de GPS ou através de algoritmos de geolocalização.

LPWANs, como LoRa e Sigfox, permitem o crescimento de sistemas IoT em larga escala (MACKEY; SPACHOS, 2019). LoRa é uma tecnologia de radio frequência de baixa potência e longo alcance proprietária da Semtech. LoRa é a camada física da rede LoRaWAN, e utiliza a técnica de modulação CSS (*Chirp Spread Spectrum*) operando em bandas de radio frequência não licenciadas (GU; JIANG; TAN, 2018). Várias pesquisas abordam o tema de geolocalização utilizando LoRaWAN (PODEVIJN et al., 2018a; MANZONI et al., 2019; FARGAS; PETERSEN, 2017). São trabalhos que utilizam a geolocalização como meio principal de localização e conseguem adquirir precisão de até 75 metros. Em (LORA ALLIANCE, 2018) vários casos de uso são apresentados, demonstrando a capacidade e precisão ao utilizar uma rede LoRa para geolocalização.

Dispositivos IoT contam com recursos computacionais limitados, processamento e armazenamento, e são muitas vezes alimentados por baterias. O desenvolvimento de um sistema IoT deve levar em consideração tais limitações. Outro desafio é a heterogeneidade do sistema. Uma vez que IoT compreende várias áreas de sensoriamento. Estes são aspectos que dificultam o emprego de técnicas convencionais de tolerância a falhas, levando ao estudo e adaptação de diferentes técnicas.

Módulos GPS em sistemas de rastreamento além de consumirem muita energia, estão suscetíveis a falhas. A falha de um GPS nesse sistema resulta na inoperabilidade da aplicação. Utilizar LoRa para enviar as coordenadas de um módulo GPS para a nuvem, é uma alternativa de conectividade capaz de cobrir grandes áreas. Além de ser uma tecnologia de baixa potência LoRa oferece um sistema de geolocalização que não necessita de mensagens adicionais, que dependendo da aplicação, pode substituir o GPS.

Na cidade de Toledo, Paraná, a UTFPR/Toledo implantou a sua rede LoRaWAN para apoiar o desenvolvimento de aplicações para cidades inteligentes (ROSSATO; SPANHOL;

CAMARGO, 2020). Uma das aplicações consiste no rastreamento de caminhões de coleta seletiva de lixo. Os caminhões utilizam GPS para enviar as suas coordenadas geográficas para a rede através da rede LoRaWAN. Ocorre em alguns momentos a perda de sinal do GPS com o satélite, gerando a falha total do sistema, sendo impossível identificar a posição do caminhão até o GPS recuperar o sinal do satélite.

Dados esses fatores, este trabalho estuda a aplicação de geolocalização através de uma rede LoRa de forma a adquirir confiabilidade e tolerância a falhas em um sistema de rastreamento de veículos, com o objetivo de oferecer uma alternativa ao GPS em tais sistemas. A solução pode trabalhar em conjunto com o GPS, sendo acionada quando a falha no GPS é detectada. Ou ainda como uma solução independente, estimando a posição de um objeto desde que receba qualquer sinal do mesmo.

1.3 ORGANIZAÇÃO DO TRABALHO

A organização do trabalho segue da seguinte forma: O Capítulo 2 apresenta os conceitos de tolerância a falhas e a arquitetura IoT junto de uma revisão sistemática em tolerância a falhas em IoT. O Capítulo 3 descreve os conceitos de LoRa/LoRaWAN e geolocalização sendo apontados alguns trabalhos relacionados a área. No Capítulo 4 se encontra o processo de desenvolvimento da técnica de tolerância a falhas e dos experimentos realizados. Os resultados obtidos se encontram no Capítulo 5. O trabalho é concluído com considerações finais e trabalhos futuros no Capítulo 6.

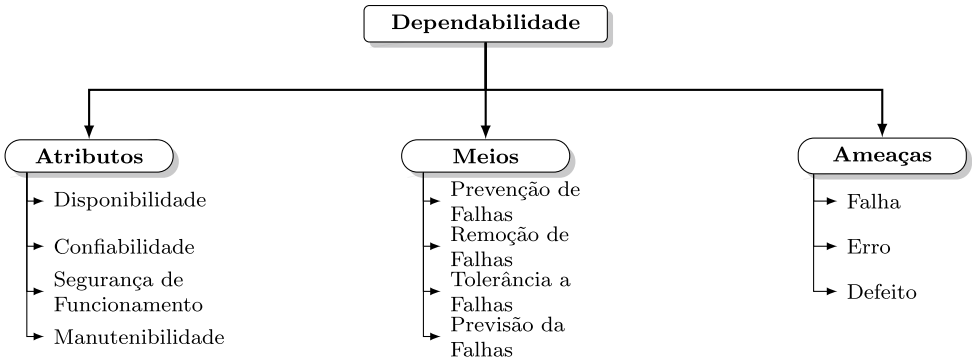
2 REVISÃO SISTEMÁTICA

Com o objetivo de identificar as principais abordagens e técnicas de tolerância a falhas que vêm sendo empregadas em IoT, foi executada uma revisão sistemática. A busca por artigos científicos contendo palavras como “tolerância a falhas” (*fault tolerance*) e “internet das coisas” (*internet of things*) em cinco importantes bases de dados retornou 2.004 trabalhos. Após a primeira e a segunda etapa, nas quais os critérios de exclusão foram aplicados, restaram 86 trabalhos para análise. Dentre esses, buscou-se quais artigos de fato tratam exclusivamente sobre tolerância a falhas em IoT. Um total de 44 foram selecionados para análise e classificadas conforme a solução de tolerância a falhas tem mais impacto.

2.1 TOLERÂNCIA A FALHAS

Com a adoção do conceito de IoT em segmentos como cidades inteligentes, agricultura, segurança e saúde, cresce também a demanda por soluções capazes de lidar com falhas. Em alguns sistemas IoT, uma única falha pode levar a consequências imprevisíveis. De acordo com (AVIZIENIS et al., 2004), dependabilidade (*dependability*) é a entrega de um serviço que justifique a sua confiança. É uma propriedade qualitativa que pode ser definida em termos de atributos, meios e ameaças, conforme representado na Figura 1.

Figura 1 – Árvore de dependabilidade.



Fonte: Adaptado de Trivedi, Bobbio e Muppala (2017).

O atributo de disponibilidade (*availability*) se preocupa com a prontidão do serviço e a capacidade de executar determinada função em um instante de tempo. Confiabilidade (*reliability*) é a capacidade de desenvolver uma determinada função em um determinado intervalo de tempo, dada a importância a continuidade do serviço. Segurança de funcionamento (*safety*) tem como objetivo proteger o ambiente e usuários quanto a perigos causados por falhas. Manutenibilidade (*maintainability*) é a capacidade de restaurar o sistema a um estado anterior de forma a executar uma função, possibilitando também manutenção e reparos.

Em relação as ameaças, uma falha (*fault*) pode ocasionar um erro interno (*error*), que pode se manifestar na forma de um defeito ou falha no serviço (*failure*) (TRIVEDI; BOBBIO; MUPPALA, 2017). Neste sentido, um defeito é uma manifestação da falha no universo do usuário.

Em sistemas baseados em trocas de mensagens, as falhas podem ser categorizadas em dois tipos: falhas de comunicação e falhas de processo (LAMPORT; LYNCH, 1990). As falhas de comunicação, quando implicam em perda de mensagens, caracterizam-se em omissão e temporização/desempenho (JALOTE, 1994). Processos podem ainda falhar por colapso/parada (*crash*), deixando de executar qualquer ação e não respondendo aos estímulos externos. A comunicação pode ainda considerar a possibilidade de falhas de particionamento de rede, que impossibilitam a comunicação entre determinados pares de nós. Em um modelo de falhas mais abrangente, as falhas arbitrárias, também conhecidas como bizantinas, incluem, além das falhas de comunicação e processo, aquelas geradas devido a comportamento malicioso.

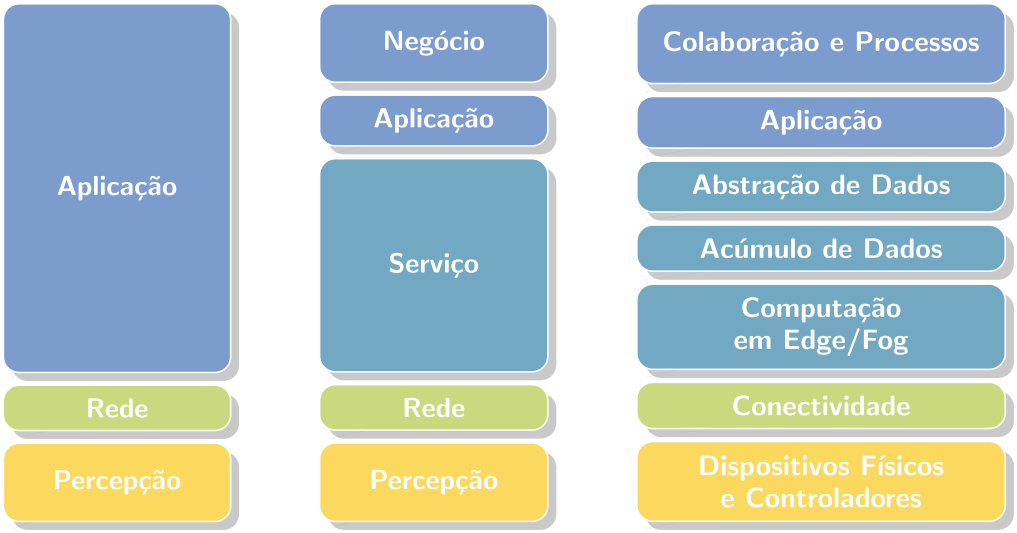
Quanto aos meios de se alcançar dependabilidade, a prevenção de falhas diz respeito ao desenvolvimento do sistema de forma a evitar que falhas internas sejam percebidas pela aplicação final. A remoção de falhas é um sistema capaz de detectar e corrigir erros antes que o defeito seja ocasionado. Assim, tolerância a falhas é o uso de técnicas para manter o funcionamento do sistema mesmo após a ocorrência de uma falha e a previsão de falhas é a capacidade de prever uma falha em termos das ameaças (TRIVEDI; BOBBIO; MUPPALA, 2017). Portanto, a definição do modelo de falhas adotado é crucial para o correto desenvolvimento de uma solução tolerante a falhas.

2.2 ARQUITETURA E PADRÕES IOT

Como afirma (AL-FUQAHA et al., 2015), existe uma necessidade crítica de uma arquitetura em camadas flexível para IoT capaz de interconectar bilhões ou trilhões de objetos heterogêneos pela Internet. Embora haja diversas arquiteturas de camadas propostas para IoT, nenhuma convergiu para um modelo de referência. Geralmente, a arquitetura de IoT é apresentada em um modelo com três ou cinco camadas (CRUZ et al., 2018; AL-FUQAHA et al., 2015), embora um modelo de sete camadas tenha sido definido pela Cisco (CISCO, 2014) (Figura 2).

No modelo de três camadas (CRUZ et al., 2018; AL-FUQAHA et al., 2015), a camada de percepção, também chamada de camada de sensoriamento, representa os objetos físicos e é responsável por coletar dados dos sensores e atuadores (informações de temperatura, umidade, peso, movimento, vibração, localização, etc.) e repassá-los para a camada de rede. A camada de Rede realiza a transmissão dos dados e define o protocolo empregado, a interface de comunicação e o roteamento. É comum o uso de tecnologias sem fio, geralmente voltadas ao baixo consumo de energia, como WiFi, LoRa, *Bluetooth Low Energy* (BLE), ZigBee e comunicação celular. A terceira camada faz a composição dos dados de forma a entregar um serviço para a aplicação, com foco em protocolos que consomem pouca largura de banda, como *Constrained Application Protocol* (CoAP), *Message Queuing Telemetry Transport* (MQTT), *Extensible Messaging and*

Figura 2 – Arquiteturas de camadas para IoT.



(a) 3 camadas. (b) 5 camadas. (c) 7 camadas.

Fonte: Autoria própria (2020).

Presence Protocol (XMPP) e *Advanced Message Queuing Protocol* (AMQP).

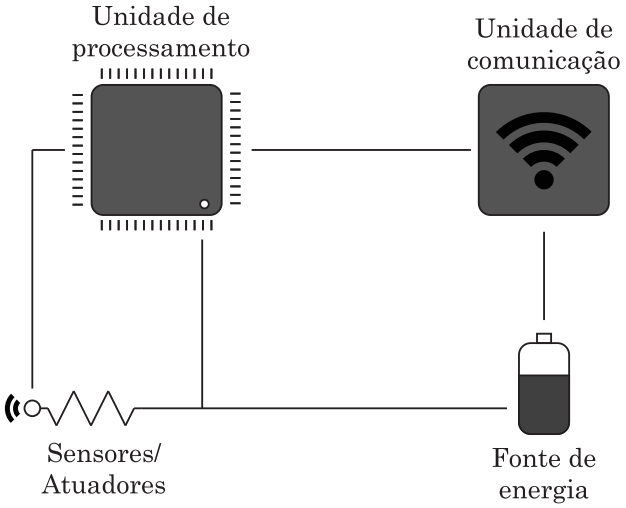
No modelo de cinco camadas (KHAN et al., 2012; WU et al., 2010), a camada de Negócio atua como a camada superior, responsável pelo controle e gerenciamento de serviços IoT fornecidos pela camada de Aplicação. A camada de Serviço atua como uma camada intermediária, realizando solicitações de serviços, agregando e processando dados e implementando funcionalidades generalizadas.

O modelo em sete camadas da Cisco (CISCO, 2014) mantém as primeiras camadas de percepção e rede, agora como Dispositivos físicos e controladores e Conectividade, respectivamente. A camada de serviço do modelo de cinco camadas é dividida em três. A Computação em Borda ou Névoa, *Edge*/*Fog*, realiza o processamento, análise e filtragem dos dados. Acúmulo de dados se refere ao armazenamento de dados. A Abstração de dados lida com o grande número de dados gerados nas camadas inferiores de forma a fornecer para a aplicação somente os seus dados de interesse. A sétima camada representa as pessoas e processos envolvidos para executar uma tarefa.

Os dispositivos IoT, presentes na primeira camada dos três modelos, também apresentam uma arquitetura básica composta por unidade de processamento, unidade de comunicação, sensores/atuadores e fonte de energia, de acordo com a Figura 3. A unidade de processamento é composta por um microcontrolador e unidade de memória e é responsável por coordenar todos os módulos do sistema e armazenar os dados. A unidade de comunicação apresenta um canal de comunicação, geralmente sem fio, utilizado para transmitir dados para a Internet ou para uma central de processamento. Os sensores e atuadores são módulos que estão em contato com o meio físico. Os sensores monitoram o ambiente transformando variáveis físicas em sinais analógicos ou digitais. Já os atuadores utilizam sinais elétricos para atuar no ambiente, como

ligar ou desligar um dispositivo, por exemplo. A fonte de energia fornece energia para o sistema, podendo dispor de bateria recarregável ou não.

Figura 3 – Arquitetura de dispositivos IoT.



Fonte: Adaptado de Santos et al. (2016).

2.3 METODOLOGIA E PROCESSO DE PESQUISA

Uma revisão sistemática é uma forma de sumarizar evidências de certa área do conhecimento e identificar questões ainda não pesquisadas, de forma a indicar um caminho a ser seguido (KITCHENHAM, 2004). A revisão sistemática executada neste trabalho se baseia na seleção de artigos que respondam a seguinte questão de pesquisa (QP):

QP: Quais são as principais técnicas e soluções de tolerância a falhas desenvolvidas para IoT?

A partir de então, foram definidas as bases de dados de seleção de artigos e as *strings* de busca. As bases de dados foram selecionadas de acordo com os seguintes critérios: (1) mecanismo de busca online; (2) mecanismo de busca avançada e; (3) base reconhecida. As fontes que cumprem os requisitos escolhidas foram IEEEXplore, ACM Digital Library, Springer Link, Science Direct e Portal de Periódicos da Capes. As *strings* de busca relacionam as palavras chave do assunto e foram agrupadas da seguinte forma:

S1: "*Fault tolerance*" AND "*Internet of Things*"

S2: ("*Fault-tolerant techniques*" OR "*fault tolerant technique*" OR "*fault tolerance techniques*") AND "*Internet of Things*"

Note-se que embora S1 e S2 possam formar uma única *string* de busca, S2 é mais restritiva, pois inclui técnicas de tolerância a falhas. A Tabela 1 apresenta o resultado da pesquisa nas bases de dados selecionadas.

Desde ponto em diante, a metodologia seguiu em três etapas: 1) aplicação de critérios de inclusão ou exclusão, 2) leitura dos títulos e resumos e 3) leitura completa das obras. Na

primeira etapa foram excluídos trabalhos anteriores a 2015, revisões (*reviews*, *surveys*), teses e artigos especulativos (*towards*, *new challenges*, etc.). Na segunda etapa, foram selecionados os artigos no qual o título e resumo estão relacionados a questão de pesquisa. Por fim, foram selecionados os trabalhos que tratam do assunto de forma mais incisiva a partir da sua leitura completa.

Quadro 1 – Quantidade de obras por base (Nov. 2019)

String	IEEE	ACM	Science Direct	Springer Link	Capes
S1	222	433	687	495	101
S2	4	14	28	20	0

Fonte: Autoria própria (2020).

A Tabela 2 apresenta o número de trabalhos aceitos em cada etapa. Inicialmente, 2.004 trabalhos foram encontrados. É possível observar a quantidade expressiva de trabalhos rejeitados de uma etapa para outra. Na primeira etapa, 1.370 trabalhos foram selecionados. Na segunda etapa, após a leitura do título e resumo, restaram 95 trabalhos.

Quadro 2 – Obras aceitas em cada etapa

Biblioteca	Quantidade	Etapa 1	Etapa 2	Etapa 3
IEEE	226	184	62	55
ACM	447	317	9	9
Science Direct	715	464	11	10
Springer Link	515	359	5	5
Capes	101	46	8	7
Total	2004	1370	95	86

Fonte: Autoria própria (2020).

A terceira etapa envolve a leitura completa dos artigos e selecionou 86 trabalhos[1], porém apenas 44 não apenas citam sua proposta como tolerante a falhas, mas também desenvolvem uma solução ou aplicam uma técnica de tolerância a falhas. Por exemplo, o trabalho de (SAHNI et al., 2017) consta entre os 86 selecionados na terceira etapa. No trabalho, uma rede sem-fio em malha é a base da proposta. Uma rede em malha inerentemente fornece caminhos redundantes. No entanto, o trabalho não desenvolve ou propõe uma solução de tolerância a falhas. Da mesma forma, o artigo em (TARAI; SHAILENDRA, 2019) não foi selecionado pois trata principalmente sobre o posicionamento ideal e seguro do controlador em uma rede de cidades inteligentes baseada em uma Rede Definida por Software (*Software Defined Network* (SDN)). A tolerância aparece de forma secundária quando os autores propõem um abordagem já prevista na literatura de SDN para lidar com falhas do controlador.

Os trabalhos foram então classificados a partir da camada da arquitetura de IoT na qual a solução de tolerância a falhas tem mais impacto, e analisados na próxima seção.

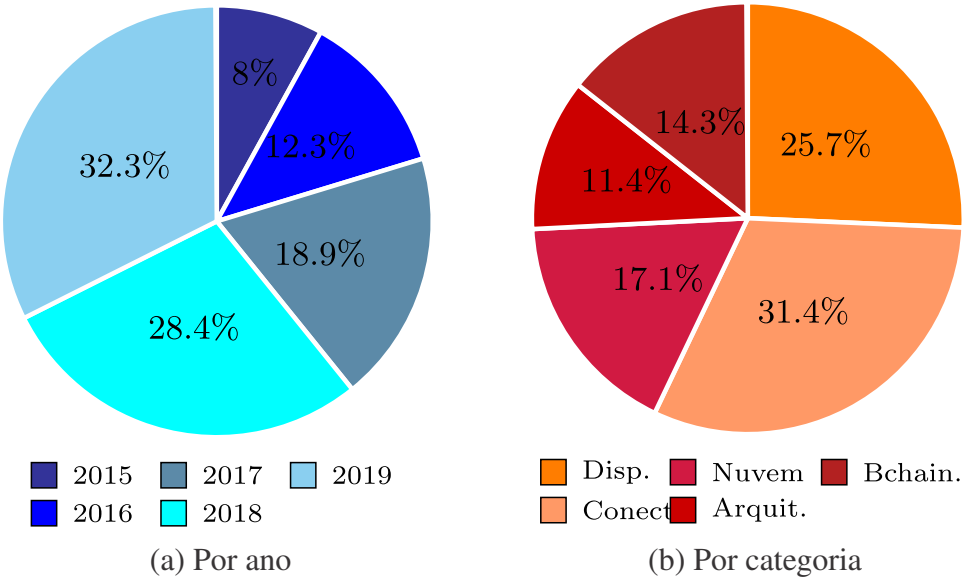
[1]Lista disponível em <https://github.com/EdanPotter/RevisaoIoT>

2.4 ANÁLISE DOS RESULTADOS

Através da revisão sistemática foi também possível identificar o crescimento de pesquisas relacionadas a tolerância a falhas em IoT ao longo dos anos, como mostra a Figura 4 (a), referente a etapa 1 da revisão.

Entre os artigos relevantes, apresenta-se os que desenvolvem, propõem ou aplicam uma solução ou técnica de tolerância a falhas. Os trabalhos foram classificados de acordo com a ênfase de atuação da solução de tolerância a falhas empregada dentro das seguintes categorias: dispositivo IoT, conectividade, borda, névoa e nuvem, arquitetura e *blockchain*. A Figura 4 (b) ilustra a porcentagem em cada categoria. Devido ao espaço limitado, 35 trabalhos são apresentados nas subseções seguintes de acordo com sua classificação.

Figura 4 – (a) Quantidade de trabalhos por ano e (b) por categoria.



Fonte: Autoria própria (2020).

2.4.1 DISPOSITIVO IOT

O trabalho em (QAIM; OZKASAP, 2018) emprega a técnica de replicação de dados em dispositivo de Redes de Sensores Sem Fio (RSSF) homogêneas para IoT. Um arcabouço, chamado DRAW, é apresentado para replicar dados de um dispositivo nos seus dispositivos vizinhos. A solução prevê a troca de mensagens periódica entre nós para disseminar atributos como o espaço disponível em memória de cada dispositivo. O modelo de falha é por colapso e o modelo de sistema adotado não prevê possibilidade de falhas enquanto a replicação acontece.

Em (CELESTI et al., 2017), há microsserviços implementados em *containers* no próprio dispositivo IoT. Os microsserviços são responsáveis por interagir com um serviço hospedado na nuvem que, por sua vez, controla os dispositivos por meio dos microsserviços. Cada dispositivo contém um serviço em seu sistema operacional para monitorar as falhas dos microsserviços.

Caso a falha de um microsserviço não possa ser reparada, a nuvem desliga o dispositivo e ativa a sua cópia presente em outro dispositivo. O serviço de detecção conta com um servidor *Network Time Protocol* (NTP) para sincronizar os relógios e a solução prevê ainda a transferência de arquivos entre os dispositivos.

O trabalho em (CHUDZIKIEWICZ; FURTAK; ZIELINSKI, 2015) apresenta uma abordagem para a implementação do conceito de IoT no domínio militar. O diagnóstico baseado em comparação é usado para detectar as falhas nos processadores dos dispositivos. Nesse método, a mesma entrada é enviada para dois nós e a resposta é comparada. Com base nos resultados das comparações, é possível realizar o diagnóstico. Posteriormente, em um segunda etapa, com a rede particionada em *clusters*, o método é aplicado para diagnosticar em qual subestrutura da rede ocorreu a falha.

Uma abordagem de tolerância a falhas energeticamente eficiente para armazenar os dados do dispositivo em um memória não-volátil é apresentada em (XU; POTKONJAK, 2016). Os autores advogam o empregado de memória não-volátil para salvar os dados do dispositivo (*checkpointing*). Uma vez que falhas ocorram e o dispositivo seja reiniciado, os dados estarão prontamente disponíveis. No entanto, salvar dados em memória não-volátil aumenta a sobrecarga e o consumo energético. Para isso é proposto um algoritmo para selecionar o estado da aplicação com o menor número de dados.

Em (ZHOU et al., 2015), os autores criam serviços virtuais como a integração de um ou mais sensores físicos de diferentes modalidades para substituir um sensor físico em caso de falhas. Ou seja, os dados dos diferentes sensores são combinados para prover redundância. Por exemplo, tanto um sensor de presença quanto uma câmera e um microfone podem detectar a presença de pessoas no ambiente. Esses três sensores são então combinados em um sensor virtual. Para combinar diferentes sensores, o trabalho usa métodos de regressão para identificar e gerar os serviços virtuais. O trabalho ainda emprega um algoritmo genético para encontrar as melhores seleções de sensores físicos ou virtuais. Ideia semelhante apresenta o trabalho (CHILIPIREA et al., 2016): um método pressupõe uma lista de recursos do dispositivo e usa essa lista para identificar a sobreposição de recursos entre diferentes dispositivos. As sobreposições podem ser usadas para gerar regras que controlam o estado de energia dos dispositivos, a fim de atingir os objetivos de robustez e eficiência energética, assim como tolerância a falhas.

Uma abordagem de baixo custo para monitorar erros em dispositivos e classificá-los de acordo com a quantidade de erros que exibem para fins de manutenção é apresentada em (GUTIERREZ et al., 2017). Já em (GOTTSCHO et al., 2017) há uma abordagem dedicada para confiabilidade em memórias: ao identificar as falhas, uma mapa de memória é criado para evitar acesso a essas regiões. A identificação de sensores falhos é o tema principal do trabalho em (CHAKRABORTY et al., 2018). Um sensor quando desligado e retirado da fonte de energia, apresenta uma curva de queda de tensão característica, normalmente devido a capacitância parasita presente no circuito do sensor. Dessa forma é possível identificar a integridade de um sensor através da curva apresentada em leituras periódicas ou ao apresentar uma leitura anormal.

Os trabalhos desta subseção destacam a natureza não confiável dos dispositivos e as condições operacionais adversas. Os dispositivos geralmente são heterogêneos e limitados computacionalmente. Em um mesmo ambiente pode haver uma variedade de *hardware* e interfaces de comunicação, por exemplo. Geralmente, soluções adotadas em RSSF preveem dispositivos homogêneos e nem sempre há comunicação com a Internet. O trabalho (ZHOU et al., 2015) explora justamente a heterogeneidade dos sensores. Os trabalhos (QAIM; OZKASAP, 2018; CELESTI et al., 2017) usam a técnica de redundância no dispositivo. Em (XU; POTKONJAK, 2016) a redundância está nos dados que são salvos em memória não-volátil. O trabalho ainda busca diminuir o impacto no consumo de energia na solução proposta. Em (CELESTI et al., 2017), apesar de a solução facilitar a gerência dos dispositivos perante falhas, o uso de *containers* exige mais poder computacional do dispositivo e o leva a consumir mais energia. O trabalho (CHUDZIKIEWICZ; FURTAK; ZIELINSKI, 2015) emprega uma técnica onde os dispositivos executam testes entre si e comparam os resultados recebidos, o que pode ser uma alternativa para detectar falhas bizantinas. Já os trabalhos em (GUTIERREZ et al., 2017; GOTTSCHO et al., 2017; CHAKRABORTY et al., 2018) abordam falhas no dispositivo, memória e sensor.

2.4.2 CONECTIVIDADE

Aplicações em redes industriais IoT, incluindo *smart grids*, requerem recursos precisos de sincronização. Nesses sistemas, a redundância de tempo e dados se torna obrigatória. A solução em (GUTIÉRREZ-RIVAS et al., 2018) é a implementação de protocolo de redundância, *High-availability Seamless Redundancy* (HSR), para o *White Rabbit* (WR). O WR é uma tecnologia baseada em Ethernet e uma das suas principais características é a sincronização em sub-nanosegundos. O HSR é usado em topologias de anel e não requer a duplicação de nenhum nó ou enlace, apenas um enlace adicional para fechar o anel.

Em uma rede sem fio, a comunicação pode ser interrompida tanto pela falha do módulo quanto pelo enfraquecimento do sinal, por exemplo. O trabalho em (PARK, 2016) analisa a probabilidade de que todos os nós de uma rede possam se comunicar com todos os outros nós, conceito chamado de *all-terminal reliability*. O estudo é realizado para uma rede sem fio padrão 802.15.4 que assume diversas topologias, incluindo uma rede em malha aleatória. Os autores calculam o tempo médio até uma falha da rede. Para aumentar a confiabilidade é proposto o emprego de redundância de enlace através de uma outra interface de comunicação no dispositivo.

A virtualização de RSSF para IoT é abordada em (KAIWARTYA et al., 2018). O trabalho aborda a falha de comunicação em redes virtuais causadas pela falha de enlace em RSSF físicas. A falha de um enlace nas RSSFs afeta os serviços IoT executados pelas redes virtuais. No trabalho, formula-se um problema de otimização que visa maximizar a tolerância a falhas e minimizar o atraso na comunicação. A solução proposta emprega um algoritmo genético adaptado. Em (DAI et al., 2019) um algoritmo genético para posicionar deterministicamente nós de uma rede IoT industrial é empregado. O problema abordado se refere a conectividade de sensores em um ambiente com obstáculos. Deseja-se maximizar a vida útil da rede considerando atributos

como conectividade, tolerância a falhas, confiabilidade, eficiência energética e capacidade de sobrevivência da rede. Para obter tolerância a falhas considera-se que um conjunto de nós estará ativo enquanto outros ficam inativos. Nós inativos assumirão a computação de nós falhos.

O trabalho em (MOSTAFA et al., 2018) monitora aplicações críticas em RSSF IoT. Apesar de muitos trabalhos na literatura abordarem o monitoramento em RSSF, de acordo com os autores, há poucas pesquisas sobre o monitoramento eficiente de redes IoT em termos de consumo de energia e sobrecarga de comunicação. Os autores afirmam que a abordagem predominante para expandir o tempo de vida da rede é o agendamento do sono (*sleep scheduling*). Nessa abordagem os dispositivos ficam inativos para economizar energia e são ativados para verificar se há ação na rede. Para garantir que cada enlace seja monitorado, os autores propõem um modelo matemático que corresponde a uma otimização multiobjetivo do consumo de energia e a sobrecarga geral de comunicação do monitoramento da rede. A solução é decomposta em três fases onde técnicas da teoria dos grafos, como o problema da cobertura dos vértices e o problema do caixeiro viajante, são empregados na resolução.

Um mecanismo escalável para monitorar as falhas e conservar energia em RSSF é o agrupamento (*clustering*) (ZHOU; LIN; SHIH, 2015). Na abordagem, os nós dos sensores são organizados em vários grupos. Cada grupo tem um líder, chamado *cluster head* (CH). Os CHs são agregam os dados e os transmitem à estação base (ou *gateway*). O monitoramento é executado separadamente em cada grupo. Ao receber a informação sobre uma falha, o CH também é responsável por substituir o serviço falho por uma cópia. O trabalho em (ZHOU; LIN; SHIH, 2015) busca minimizar o custo da comunicação do monitoramento de falhas em *clusters*.

O trabalho em (LIN et al., 2019) lida com a falhas do CH. Para garantir que as mensagens sejam roteadas ao *gateway*, se um CH falhar, seus membros serão gerenciados por CHs sem falhas. Para escolher um novo CH, primeiro os recursos disponíveis de todos os CHs sem falhas são organizados logicamente como um CH virtual para ser o *backup* comum de todos os CHs com falhas. Em seguida, obtém-se a tolerância a falhas com o consumo total mínimo de energia entre todos os CHs sem falhas. Também no contexto de roteamento, o trabalho em (FAN et al., 2017) propõe um protocolo de roteamento para rede local IoT capaz de encontrar novas rotas caso a rota anterior tenha sido invalidada. Ou seja, o trabalho assume que há enlaces redundantes ou alternativos. Ao reparar as falhas, a rota original é retomada.

Embora haja uma extensa pesquisa sobre protocolos de roteamento em RSSF, a maioria não considera a arquitetura densa de IoT. Em (NASSER et al., 2017) a solução é agrupar nós usando múltiplas estações-base e prioridade de pacotes. A detecção de falhas conta com um pacote especial para notificar o CH caso o nó não tenha dados para enviar. Se o CH não receber nenhum dado ou pacote especial de um dispositivo IoT em um intervalo de tempo, o dispositivo é considerado falho.

Em (HASAN; AL-TURJMAN, 2017b; HASAN; AL-TURJMAN, 2017a), os autores afirmam que o roteamento tolerante a falhas é frequentemente formulado como um problema de otimização para estabelecer k-caminhos disjuntos que garantam a conectividade mesmo após a

falha de até $k-1$ caminhos. A solução gera grande esforço computacional, principalmente se resolvido em dispositivos individuais. No trabalho é desenvolvido um algoritmo de roteamento baseado em otimização de enxame de partículas. O algoritmo é computacionalmente eficiente e capaz de reconstruir os caminhos seguindo parâmetros de qualidade de serviço (*Quality of Service* (QoS)) em termos de consumo de energia, atraso e vazão.

Uma características das redes IoT é sua topologia dinâmica e instabilidade devido ao enfraquecimento de bateria, mobilidade, ruído, ou falha no módulo de comunicação. O desafio é ainda maior devido aos enlaces sem fio de curto e/ou longo alcance. Destaca-se o grande número de trabalhos sobre tolerância a falhas no roteamento em IoT. No entanto, a tolerância a falhas impacta em esforço computacional, sobrecarga gerada por troca de mensagens e problemas de escalabilidade conforme o problema aumenta. Trabalhos como (MOSTAFA et al., 2018; KAIWARTYA et al., 2018; HASAN; AL-TURJMAN, 2017b; DAI et al., 2019) lidam com a tolerância a falhas como um problema de otimização envolvendo o consumo de energia e os custos de comunicação. De fato, muitas soluções de roteamento em IoT tem suas bases em RSSF, como visto em (ZHOU; LIN; SHIH, 2015; LIN et al., 2019; NASSER et al., 2017).

2.4.3 BORDA, NÉVOA E NUVEM

Uma arquitetura confiável e tolerante a falhas para comunicação entre o dispositivo IoT e os vários níveis de computação entre borda e nuvem é definida em (GROVER; GARIMELLA, 2018). Na arquitetura proposta, a nuvem é distribuída em quatro níveis (nuvem, névoa, neblina e orvalho) com base na capacidade de processamento e distância do dispositivo IoT. Os dados coletados são replicados na borda da rede, ou seja, um dos três primeiros níveis. Na ocorrência de uma falha em um dos servidores da borda, o sistema redireciona a comunicação do dispositivo para um servidor alternativo. O servidor é escolhido no melhor nível possível na hierarquia, com base atraso máximo permitido.

O trabalho em (POWER; KOTONYA, 2018) propõe um arcabouço baseado em uma arquitetura de microsserviços. Há dois microsserviços complementares: um que usa processamento de eventos complexos (*Complex Event Processing* (CEP)) para detectar falhas em tempo real e outro que emprega algoritmos de aprendizado de máquina para detectar padrões e mitigar futuras falhas antes que elas ocorram. O primeiro é hospedado na névoa para oferecer rápida detecção e recuperação perante falhas e o segundo é hospedado na nuvem. Os microsserviços recebem informações dos dispositivos e suas ações são baseadas na análise dos dados recebidos. Os mesmos autores também empregam o CEP aliado a ciência do contexto para propor um arcabouço que descreve genericamente as falhas e seus efeitos no sistema (POWER; KOTONYA, 2019).

Em um contexto *smart home*, (ARDEKANI et al., 2017) propõe uma plataforma distribuída para aplicações heterogêneas. A ideia central é retirar o ponto único de falha representado na comunicação entre o *gateway* e a nuvem movendo a computação para os dispositivos na rede interna. Os dispositivos são capazes de criar cópias virtualizadas de cada elemento, que

são processadas pelos elementos reais e então transmitidas aos virtuais. Na plataforma, as aplicações podem escolher entre serviços de entrega de melhor esforço e entrega confiável. Já (KODESWARAN et al., 2016) trabalha com tolerância a falhas em monitoramento de atividades do cotidiano. Nesse ambiente muitas vezes são utilizados vários sensores para monitorar somente uma atividade. Com o objetivo de diminuir o período de manutenção e manter o funcionamento do sistema com a presença de sensores falhos é proposto um arcabouço para agregar os dados e extrair a redundância funcional presente nos sensores e agendar uma possível manutenção.

Falhas bizantinas na alocação de serviços na névoa é um dos objetivos de (XU et al., 2018). Quando um usuário requisita um serviço o nós principal envia os dados para as réplicas. O número de réplicas é $n \geq 3f + 1$, sendo f o número de nós falhos.

Os trabalhos desta categoria buscam distribuir o processamento centralizado da nuvem em equipamentos mais próximos dos dispositivos IoT. Além da redundância, a estratégia aumenta o desempenho, diminuindo a latência de comunicação e de detecção de falhas locais.

2.4.4 ARQUITETURA

Uma arquitetura escalável e tolerante a falhas no contexto da saúde é proposta em (GIA et al., 2015). Para contornar a baixa confiabilidade dos protocolos do padrão IEEE 802.15.4, os autores empregam o protocolo *IPv6 over Low power Wireless Personal Area Networks* (6LowPAN) e propõem uma arquitetura personalizada. Na arquitetura os *gateways* são compostos por nós de processamento, chamados de *sink nodes*, que monitoram o estado do sensores. Os sensores enviam dados para um dos nós de processamento do *gateway*. Ao perceber um sensor inativo, o *gateway* inicia um protocolo para descobrir o motivo da inatividade. Se após a conclusão do protocolo o nó ainda não responder, outro nó de processamento envia uma mensagem de alerta a todos os dispositivos. Se o dispositivo receber a mensagem, passa então a se comunicar com o outro nó de processamento.

A falha e atraso entre a comunicação entre dispositivos, composto por sensor e/ou atuador, e nuvem, é apresentado por (SHARMA et al., 2018). De fato, o trabalho apresenta uma arquitetura de sistema para garantir a continuidade de uma ação de controle, apesar da perda de de comunicação entre o atuador e a nuvem. Basicamente, a solução conta com redundância de nós e enlace. Entre as soluções propostas, caso a falha de um enlace seja detectada na comunicação com a nuvem, um controle local assume e começa a direcionar os dados para o atuador.

Uma arquitetura IoT orientada a serviço para prevenção e previsão de desastres usando aprendizado de máquina é apresentada em (PILLAI et al., 2019). A implementação inclui redundância modular tripla para garantir a disponibilidade e confiabilidade dos dados: três sensores, um principal e dois auxiliares. As leituras dos sensores são comparadas e um esquema de votação é executado na borda para assegurar que os dados encaminhados para a nuvem sejam confiáveis. Em um contexto semelhante, (FURQUIM et al., 2018) apresenta uma arquitetura e avaliação do mecanismo *System for dEtecting and forecasting Natural Disasters based on IoT* (SENDI). SENDI é um sistema tolerante a falhas baseado em IoT e aprendizado de máquina. A

arquitetura é composta por três níveis: Nuvem, névoa e RSSF. A comunicação entre os níveis é projetada para ser tolerante a falhas: quando um dispositivo de um nível falhar, o outro nível pode substituir as suas funções.

Nesta subseção, as soluções encontradas buscam prover tolerância a falhas por meio da organização dos dispositivos e a sua comunicação com a Nuvem, seja pelo monitoramento de atividade ou pelo uso de enlaces e sensores redundantes.

2.4.5 BLOCKCHAIN

Blockchain é uma solução de *Distributed Ledger Technology* (DLT) que define uma rede de transações distribuídas em que os nós transferem blocos de informação, contendo identidade e conteúdo, de forma *Peer-to-Peer* (P2P) (FERNÁNDEZ-CARAMÉS; FRAGA-LAMAS, 2018; LEKA; SELIMI; LAMANI, 2019). Devido a ser uma rede descentralizada, que provê privacidade, rastreabilidade e controle de acesso, vem sendo usada em várias áreas, especialmente criptomoedas e contratos inteligentes (*smart contracts*).

LVChain(YU et al., 2018) é uma solução de *blockchain* para controle de acesso em IoT através de um algoritmo de consenso por votação, mais leve que as soluções tradicionais que utilizam prova de trabalho (*Proof-of-Work* (PoW)) e prova de participação (*Proof-of-Stake* (PoS)). O objetivo é disponibilizar uma solução descentralizada, escalável, tolerante a falhas e que possa ser executada em modo *off-line*, levando também em consideração a limitação energética dos dispositivos.

O trabalho de (BAI; XIA; FU, 2019) propõe uma arquitetura de consenso de duas camadas. A camada base é formada por *end* e *edge-devices*. Alguns dispositivos *edge* com maior capacidade de processamento são escolhidos para formar a camada superior. A camada base lida com blocos de transações, enquanto a camada superior trabalha os blocos de conjunto de transações. A tolerância a falhas é garantida por um algoritmo de consenso não-bizantino. De acordo com os resultados, mesmo com mais de 51% dos nós comprometidos, os membros confiáveis ainda podem convergir.

Já em (WANG et al., 2018) é definida uma arquitetura multicamadas com *Cloud* e *Fog* integradas. O objetivo é obter o consenso com um número mínimo de troca de mensagens, com garantias de tolerante a falhas. Para isso foi proposto um protocolo de consenso a ser executado na *Fog* (mais próximo do dispositivo final) e, posteriormente, na *Cloud*. Um protocolo de consenso baseado em consistência interativa (*interactive consistency*), que se baseia na maioria, é utilizado.

Em (YOON; CHOI; KIM, 2019) é proposta uma arquitetura de descobrimento de serviços baseada em *blockchain* denominada BlockONS. BlockONS tem o objetivo de cobrir as falhas de segurança existentes no protocolo *Object Name Service* (ONS), que apresenta vulnerabilidades parecidas com as existentes no protocolo *Domain Name System* (DNS). A solução para tolerância a falhas propõe o armazenamento de dados *off-chain* (transações registradas fora da *blockchain*), que podem ser sincronizados com *on-chain* mesmo quando o serviço de um nó falha.

Em sistemas de transporte inteligente, o problema de consenso bizantino é abordado por (WANG; LIN; YAN, 2019) em uma proposta de protocolo para *Vehicular Ad-hoc NETwork* (VANET). O consenso em uma rede VANET resulta em uma rede tolerante a falhas, uma vez que veículos são suscetíveis a falhas e o meio de transmissão pode sofrer interferências e ataques. Para isso são realizadas trocas de mensagens em *clusters* através do protocolo proposto, *Reliable and Trustworthy Agreement Protocol* (RTAP), finalizando em um acordo entre os dispositivos operantes, respeitando-se os requisitos do acordo bizantino.

Conforme observado nas trabalhos apresentados, a tolerância a falhas para IoT utilizando a tecnologia de *blockchain* é fornecida por meio de protocolos de consenso, nos quais os valores de entrada dos participantes devem resultar um acordo. Apesar das vantagens, os recursos computacionais limitados presentes em dispositivos IoT fazem com que seja um desafio implementar tal tecnologia na área(BAI; XIA; FU, 2019).

3 REFERENCIAL TEÓRICO

Embora a seção anterior tenha apresentado diversas soluções de tolerância a falhas para tecnologias de redes sem-fio, não se encontrou trabalho que aborde a tolerância a falhas em redes LPWAN. Também não se encontrou aplicação IoT que explore a falha de um módulo GPS em um dispositivo que visa informar a localização de algum objeto. Desta forma, este capítulo apresenta arquitetura e padrões LoRa e LoRaWAN. Os conceitos e técnicas de geolocalização são apresentados na sequência. Por fim a seção de trabalhos relacionados, onde são expostos alguns trabalhos que tem a geolocalização como tema principal.

3.1 LORA E LORAWAN

LoRa é uma tecnologia de radiofrequência (RF) proprietária da empresa Semtech. Permite enviar dados por longas distâncias, com baixo consumo de energia e baixas taxas de transmissão. LoRa utiliza a técnica de modulação *Chirp Spread Spectrum* (CSS) e opera em bandas de radiofrequência não licenciadas. No Brasil o padrão utilizado é o mesmo da Austrália, na faixa de 915 MHz, compreendendo a faixa de 902 MHz a 907,5 MHz e 915 MHz a 928 MHz.

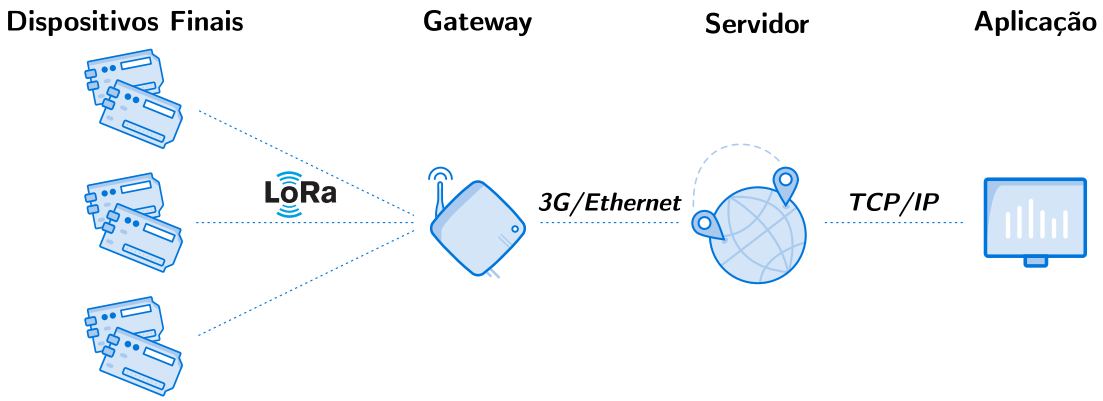
Para configuração de dispositivos finais que compõem uma rede LoRa, uma das principais variáveis a ser definida é o fator de espalhamento, *Spreading Factor* (SF), (OLIVEIRA; CONCEIÇÃO; NETO, 2018). O SF é um valor discreto, de 7 a 12, e influencia na duração da modulação interferindo nos valores de *bit rate* (quantidade de *bits* por segundo) e *air time* (tempo em que o pacote está no ar).

As mensagens em uma rede LoRa podem ocorrer em duas direções. *Uplink*, do dispositivo para a rede, ou um *downlink*, da rede para o dispositivo. Os dispositivos finais podem ser configurados em três classes. Classe A: o dispositivo final inicia a comunicação com o *gateway*, enviando um *uplink* e abrindo duas janelas de *downlink* do *gateway*. Classe B: o dispositivo final se comunica com o *gateway* através de mensagens periódicas, determinando o momento de transmissão do dispositivo e abrindo duas janelas de recepção. Classe C: o dispositivo inicia a transmissão, abre duas janelas de recepção mantendo uma aberta até a próxima transmissão (LORA ALLIANCE, 2015).

LoRaWAN é um protocolo de comunicação de rede que tem LoRa como camada física (LORA ALLIANCE, 2015). A Figura 5 mostra a topologia típica de uma rede LoRaWAN. Onde os dispositivos finais, são dispositivos que através de sensores e atuadores monitoram e atuam no ambiente. O *gateway*, responsável por conectar os dispositivos LoRa com a Internet repassando a informação obtida para o servidor. A aplicação utiliza de serviços fornecidos pelo servidor para exibir informações e, ou, fornecer funcionalidades.

Já a arquitetura de uma rede LoRaWAN, pode ser vista na Figura 6. Podemos mapear a LoRaWAN como a segunda e terceira camadas do modelo OSI (enlace e rede). Onde LoRa

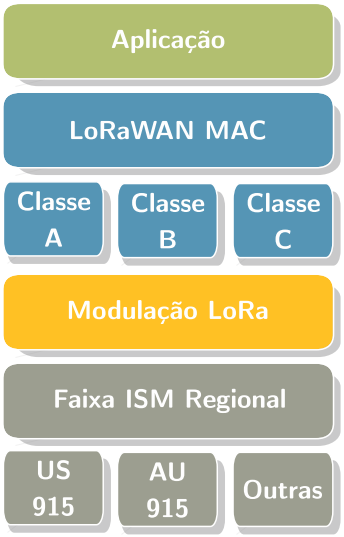
Figura 5 – Topologia de uma rede LoRaWAN.



Fonte: Autoria própria (2020).

forma a camada física, definindo a faixa de frequência a ser utilizada. A modulação LoRa se encaixa no enlace. E LoRaWAN MAC se equivale a camada de rede.

Figura 6 – Pilha do protocolo LoRaWAN.



Fonte: Adaptado de LORA ALLIANCE (2015).

3.2 GEOLOCALIZAÇÃO

Sistemas de RF realizam a troca de mensagens por ondas eletromagnéticas que viajam pelo ar com mudanças em amplitude e frequência. Conhecidas as características da onda é possível identificar o local da fonte de transmissão. Existem diversas abordagens de localização baseadas em RF: podendo ser baseada em medidas de distância como por tempo de chegada (*Time of Arrival* (ToA)), tempo de voo (*Time of Flight* (ToF)), indicador de potência do sinal recebido (*Received Signal Strength Indication* (RSSI)), e diferença de tempo de chegada (*Time Difference of Arrival* (TDoA)), ou até por medida de ângulo como ângulo de chegada (*Angle*

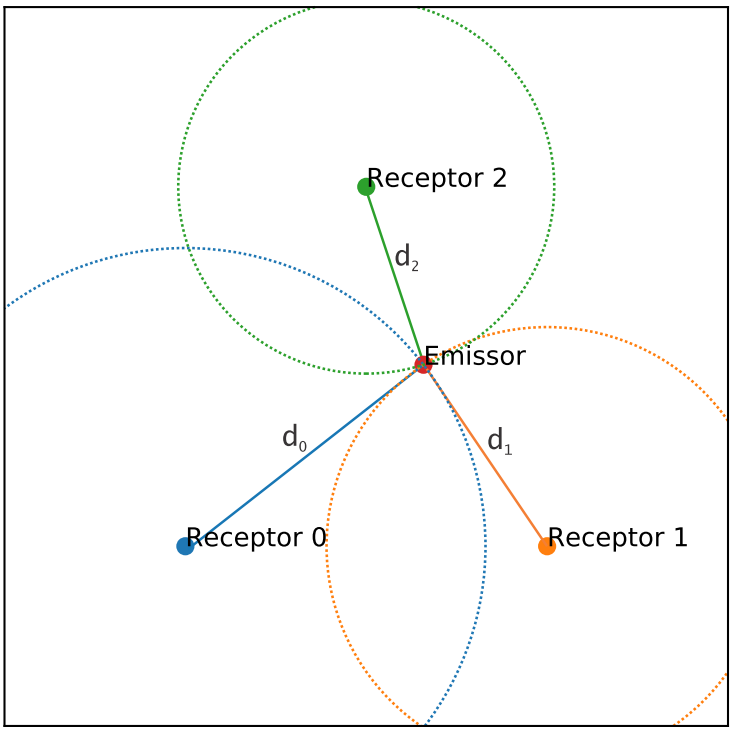
of Arrival (AoA)) (KHELIFI et al., 2019). Essas abordagens utilizam diferentes técnicas para estimar a posição de um objeto dependendo do número de dispositivos envolvidos, sendo as mais comuns triangulação, trilateração e multilateração. Essas técnicas utilizam duas bases receptoras, três e mais de três, respectivamente, para calcular a posição do emissor (FARGAS; PETERSEN, 2017).

As técnicas baseadas em medidas de distância RSSI e TDoA são descritas nesta seção. Estas técnicas foram escolhidas pois suas características estão presentes na rede LoRa. As deduções matemáticas são feitas com a multilateração como referência, dessa forma podendo ser generalizada para uma estratégia de trilateração.

3.2.1 INDICADOR DE POTÊNCIA DO SINAL RECEBIDO - RSSI

O indicador de potência do sinal recebido, RSSI, fornece informação sobre a qualidade do sinal que está sendo recebido. Conhecendo este valor e o modelo de propagação do sinal é possível estimar distância entre o transmissor e o receptor (GU; JIANG; TAN, 2018). Com três ou mais receptores essas distâncias combinam três circunferências, em que a intersecção entre elas aponta a localização do transmissor. A Figura 7 demonstra as circunferências obtidas através do cálculo de distância por RSSI de três *gateways*. Percebe-se que no ponto de intersecção entre elas tem-se o emissor.

Figura 7 – Circunferências de distância calculadas com RSSI.



Fonte: Autoria própria (2020).

A distância entre o transmissor e o receptor pode ser calculada através da Equação 2, que é derivada de um modelo empírico (MANZONI et al., 2019). Onde d é a distância entre

receptor e emissor, $RSSI$ é o indicador de potência do sinal, e n é um valor experimental, normalmente entre 2 e 6, que representa o fator de perda de sinal.

$$RSSI = -10n \log_{10}(d) \tag{1}$$

$$d = 10^{-RSSI/10n} \tag{2}$$

Utilizando Pitágoras a distância entre o emissor e os receptores pode ser representada pela Equação 3. Onde d_i é a distância entre o emissor e o receptor i, (x, y) é a posição do emissor, e (x_i, y_i) é a posição do receptor i.

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} = 10^{-RSSI/10n} \tag{3}$$

Rearranjando os membros obtém-se a Equação 4.

$$\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i = 0 \tag{4}$$

A solução do sistema de equações dado pela Equação 4 resulta em uma posição (x, y) correspondente ao emissor. Em (MANZONI et al., 2019) uma solução analítica é apresentada. Mas soluções analíticas compreendem o domínio das soluções exatas. Em um ambiente real existe ruído, interferência e outros fatores que podem fazer com que uma solução exata não exista.

3.2.2 DIFERENÇA DO TEMPO DE CHEGADA

Diferença do tempo de chegada, TDoA, é uma técnica de estimativa de posição baseada em hipérboles. É uma técnica que utiliza do tempo de chegada da mensagem. Dessa forma, caracterizando versatilidade, pois não há necessidade de sincronização entre transmissores e receptores (O'KEEFE, 2017). Para estimar a localização de um emissor, faz-se necessário pelo menos três bases de recebimento (BISSETT, 2018), *gateways*, como visto na Figura 8. A Figura 8a mostra as distância de cada receptor para o emissor, e na Figura 8b as hipérboles formadas pelo Receptor 0, que recebeu o sinal primeiro, são mostradas.

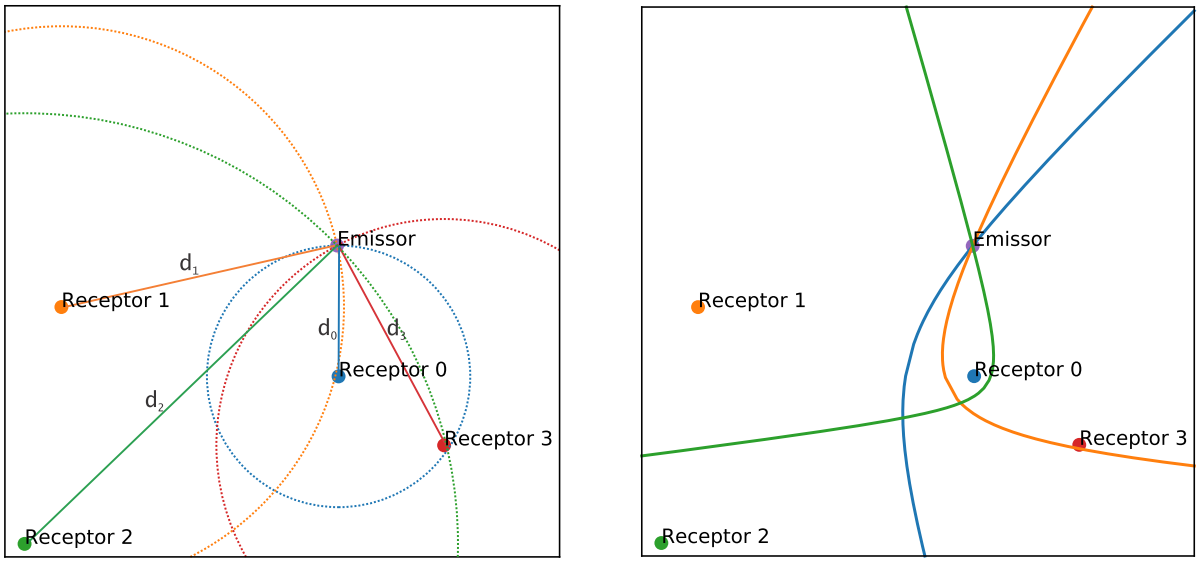
Dessa forma, para estimar a posição do emissor (x, y), tendo n bases receptoras de posição (X_i, Y_i), $i = 0, .., n - 1$. A Equação 5 representa a diferença de distância entre bases de recebimento d_i e d_j, com $j \neq i$, onde C é a velocidade da luz, t_i e t_j tempos de chegada nas respectivas bases de recebimento.

$$\Delta d_{i,j} = C(t_i - t_j) = d_i - d_j \tag{5}$$

Utilizando Pitágoras a distância entre o transmissor e os receptores pode ser representada pela Equação 6.

$$d_i = \sqrt{(x - X_i)^2 + (y - Y_i)^2} \tag{6}$$

Figura 8 – Multilatereção por TDoA.



(a) Distâncias dos receptores para o emissor.

(b) Hipérboles formadas pelo Receptor 0.

Fonte: Autoria própria (2020).

Fixando como âncora a base que recebeu o sinal primeiro e combinando as equações 5 e 6, têm-se um conjunto de equações de segundo grau, que podem ser vistas na Equação 7. Onde (x_a, y_a) é a posição da base que recebeu o sinal primeiro, $(x_j, y_j), j \neq i$ a posição das outras bases, t_a o tempo de recebimento na âncora e $t_j, j \neq i$ o tempo de recebimento das outras bases.

$$\sqrt{(x - x_a)^2 + (y - y_a)^2} - \sqrt{(x - x_i)^2 + (y - y_i)^2} + C(t_i - t_a) = 0 \tag{7}$$

A solução desse sistema não é algo trivial. A solução analítica apresenta diferentes modelos matemáticos (STEFAŃSKI, 2009). Outra resolução do problema se dá por métodos numéricos. Métodos numéricos oferecem uma complexidade menor ao custo de precisão e poder computacional. Métodos comuns são série de Taylor e mínimos quadrados empregados neste trabalho.

3.3 TRABALHOS RELACIONADOS

Esta seção apresenta trabalhos relacionados diretamente ao tema de geolocalização. Percebe-se diferentes resultados e métodos utilizados, mas boa parte se baseia no mesmo princípio de estimativa de posição.

O trabalho de (PODEVIJN et al., 2018b) utiliza uma rede LoRa pública para realizar o rastreamento de dispositivos. Através de cálculos TDoA foi possível estimar a posição do dispositivo em diferentes cenários, durante uma caminhada ou dirigindo, obtendo uma precisão média de 200 metros. Este trabalho evoluiu para (PODEVIJN et al., 2018a) em que, o algoritmo

de geolocalização leva em consideração a infraestrutura das ruas e a velocidade do dispositivo, que aumentou a precisão, resultando em uma precisão média de 75 metros.

Já em (MANZONI et al., 2019) através de uma rede LoRa é realizado a geolocalização de veículos em um ambiente interno. Com uma arquitetura formada por 3 *gateways* a localização dos veículos é estimada pela potência de sinal recebido, RSSI. Foram realizados experimentos em um estacionamento obtendo precisão de até 20 metros.

A localização de um dispositivo estático é realizada utilizando TDoA em uma rede LoRa em (FARGAS; PETERSEN, 2017). Por o dispositivo não apresentar movimento o trabalho pôde realizar várias medidas para estimar o seu local. Dessa forma, foi possível retirar os pontos discrepantes que poderiam causar um eventual erro. O trabalho conseguiu estimar a posição do dispositivo com uma média de 100 metros de precisão.

Em (CUNHA; VITOR; NETO, 2020) a estimativa de posição é feita em uma rede Wi-Fi por RSSI. Com um *smartphone* foi feita a coleta de dados, as diferentes potências de cada roteador, foi então realizada a estimativa de distância por diferentes modelos de RSSI para realizar os cálculos de trilateração. Por fim, os cálculos passaram por um processo de otimização que resultou na estimativa de um dispositivo com erro de 0,48 metros.

Percebe-se uma precisão variável nos trabalhos relacionados a geolocalização. Sendo relativa ao cenário e contexto proposto, por exemplo, em um ambiente interno a precisão é maior se comparada a um ambiente externo, mas talvez a precisão relativa seja parecida. Observa-se também o uso de diferentes técnicas para alcançar uma maior precisão, como em (PODEVIJN et al., 2018a) e (CUNHA; VITOR; NETO, 2020). Por fim, a maioria dos trabalhos aponta que é possível realizar a localização de objetos através da geolocalização.

4 SOLUÇÃO TOLERANTE A FALHAS PARA GEOLOCALIZAÇÃO

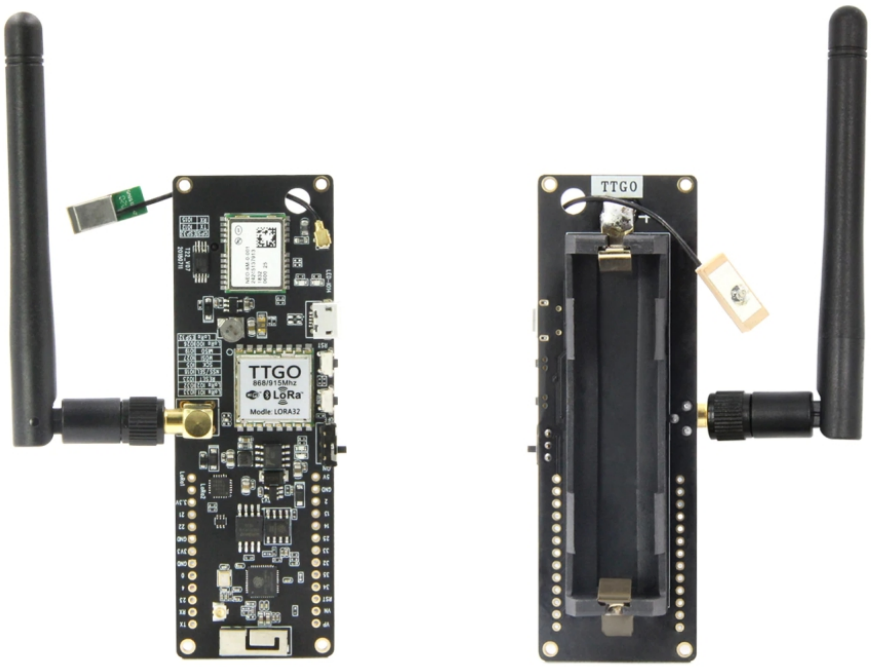
Este capítulo apresenta a metodologia utilizada durante o desenvolvimento do trabalho e descreve a solução proposta. Como na revisão sistemática, redundância é uma técnica tolerante a falhas muito utilizada. Sendo bastante aplicada em termos de conectividade, *hardware* e processamento. Pensando nisso, a solução proposta para localização em redes LoRaWAN como abordagem para mitigar a falha do GPS se encontra em um âmbito de redundância podendo ser classificada como de *hardware* e processamento em borda.

4.1 DISPOSITIVOS

4.1.1 TTGO T-BEAM

O dispositivo responsável por realizar o rastreamento dos caminhões da coleta seletiva de lixo é o TTGO T-BEAM. É dispositivo da LILYGO, construído em torno do microprocessador ESP32. O T-BEAM conta com todas as funcionalidades disponíveis do ESP32, WiFi, Bluetooth, ADC, DAC e etc, mais as funcionalidades dos módulos integrados, GPS e LoRa.

Figura 9 – TTGO T-Beam V0.7 ESP32.



Fonte: LILYGO (2020).

As especificações do T-BEAM são comparadas com outros dispositivos semelhantes e podem ser vistas na Tabela 1. Percebe-se que o T-BEAM devido a conter um ESP32 possui um

poder de processamento superior ao Arduino, mas ainda é limitado com relação ao Raspberry Pi. Além dos fatores apontados, o T-BEAM foi escolhido pois já teve o seu desempenho avaliado na rede LoRaWAN da UTFPR (ALMEIDA et al., 2020).

Tabela 1 – Comparação entre dispositivos.

	Arduino Uno R3	ESP32	Raspberry Pi 3 B+
CPU	ATmega328	Xtensa LX6 DualCore	BCM2837B0 QuadCore
Clock	20MHz	240Mhz	1.4Ghz
Arquitetura	8 bit	32 bit	64 bit
RAM	2 KB SRAM	520 KB SRAM	1 GB SDRAM

Fonte: Autoria própria (2020).

A programação do T-BEAM pode ser feita em C, C++ e Python. Podendo utilizar os *frameworks* do Arduino ou da Espressif, ESP-IDF. O protocolo LoRa disponível para o dispositivo é fornecido por *firmware* e distribuído em forma de biblioteca, LMIC. LMIC foi inicialmente criada pela IBM, que descontinuou o projeto. Sendo assumido então por Matthijs Kooijman, que hoje disponibiliza o código de forma aberta em seu GitHub. A arquitetura de um *firmware* que utiliza LMIC pode ser vista na Figura 10. Sendo a camada do microcontrolador a camada de *hardware*, contendo o módulo de radiofrequência LoRa, sensores e atuadores. A camada de *software* utiliza o *framework* do Arduino, sendo composta pela lógica da aplicação, os *drivers* dos componentes e da biblioteca LMIC. A biblioteca LMIC é responsável por seguir os protocolos LoRaWAN e *Medium Access Control* (MAC), manter o ambiente de execução e fazer a abstração de *hardware* com relação aos transmissores.

4.1.2 GATEWAYS LORA

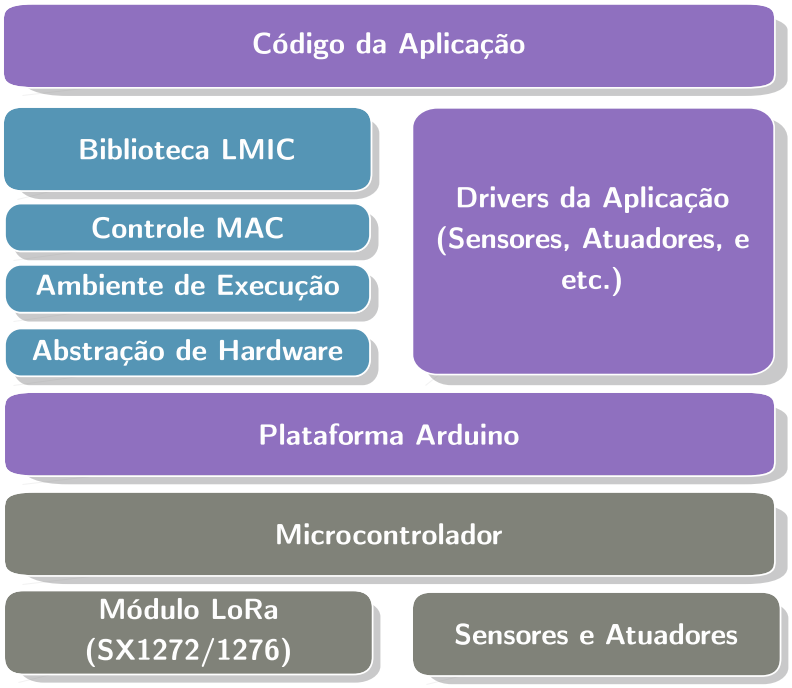
Foram utilizados três *gateways* compostos por um Raspberry Pi 3 B+ e um módulo, HAT, *gateway* LoRaWAN da Radioenge, que pode ser visto na Figura 11. Os módulos possuem GPS integrado, o que possibilita a sincronia de tempo entre os dispositivos com uma precisão de microsegundos.

Os *gateways* se conectam a rede através da *The Things Network* (TTN). TTN é uma rede colaborativa que conecta *gateways* LoRa (THE THINGS NETWORK, 2020). É uma plataforma gratuita e aberta que conta com várias funcionalidades e integrações a serem utilizadas por aplicações desenvolvidas.

4.2 GEOLOCALIZAÇÃO

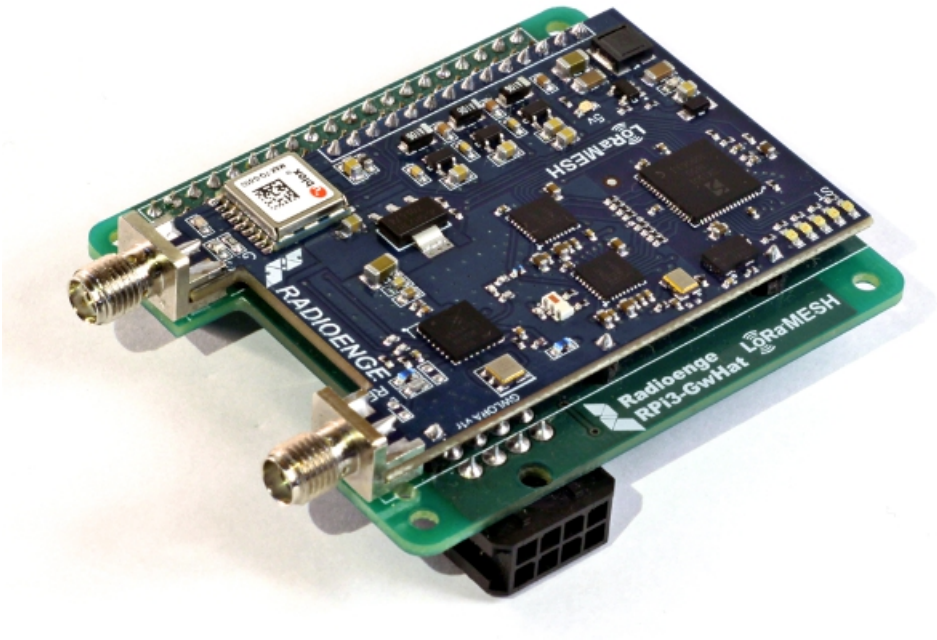
Para a geolocalização foram utilizados algoritmos de geolocalização, RSSI e TDoA, implementados em Python e a plataforma LoRa Cloud que realizam os cálculos utilizando RSSI e TDoA. Python foi escolhido devido a ser uma linguagem de fácil utilização e poderosa com

Figura 10 – Arquitetura em camadas de uma aplicação LMIC.



Fonte: Adaptado de MCCI (2019).

Figura 11 – *Gateway* LoRaWAN Radioenge RD43HAT.



Fonte: Radioenge (2020).

relação ao processamento de dados. LoRa Cloud é uma plataforma da Semtech que oferece o serviço de geolocalização através de uma API. O serviço grátis da plataforma permite mil

requisições diárias. Para mais requisições se faz necessário aderir a planos pagos. Os resultados das duas soluções de geolocalização serão comparados de forma a escolher a melhor opção para uma aplicação final.

A visualização dos dados ocorre por uma aplicação *Web* desenvolvida em JavaScript com o auxílio da biblioteca Leaflet. Leaflet é um biblioteca de visualização de mapas interativos que utiliza dados do OpenStreetMap. OpenStreetMap é um projeto de código aberto que tem o objetivo de criar um mapa livre e editável do mundo. É mantido totalmente pela comunidade que mantém atualizado os dados de estradas, trilhos, cafés, estações ferroviárias e entre outras informações geográficas.

4.3 SOLUÇÃO TOLERANTE A FALHAS

A solução consiste em envios periódicos de localização dos caminhões para a rede através da rede LoRa. Em casos em que o GPS não recebe sinal do satélite um valor nulo é enviado, deixando explícito que o cálculo de geolocalização é necessário. A posição do caminhão é estimada utilizando algoritmos de geolocalização e então é enviada para visualização no mapa.

O fluxograma da proposta pode ser visualizado na Figura 12. Onde os caminhões equipados com dispositivos LoRa enviam um pacote para o *gateway*, que repassa para o servidor de aplicação onde ocorre o processamento. Verificando a se existe coordenada ou se mais de três *gateways* receberam a mensagem, de forma a calcular a localização e ajusta-la. Finalizando com a visualização da posição na aplicação.

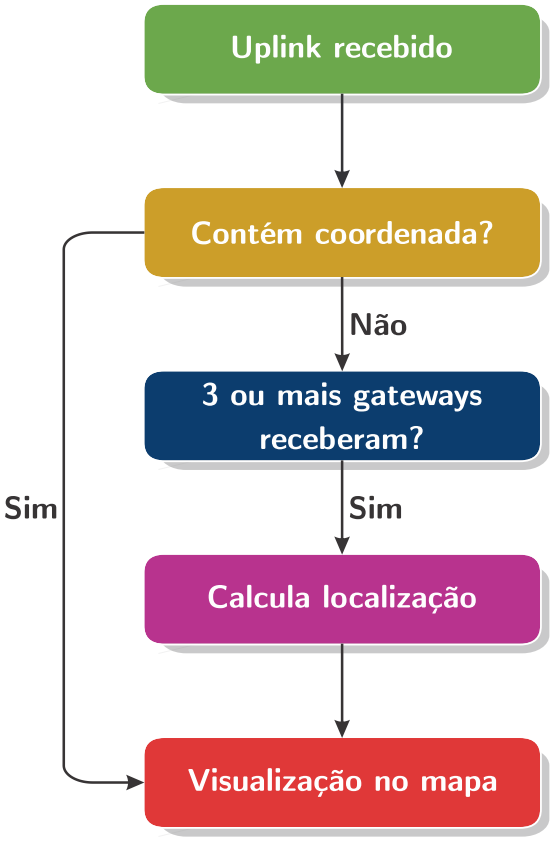
A Figura 13 apresenta a topologia da rede, onde estão presentes os caminhões equipados com dispositivos rastreadores, que enviam a sua posição geográfica através da LoRaWAN para os *gateways*. Os *gateways* repassam o pacote para o servidor de rede, TTN, que fornece os dados para o servidor de aplicação, onde ocorre o processamento. Durante o processamento os dados passam por algoritmos de geolocalização implementados e a solução comercial LoRa Cloud. A visualização dos dados ocorre pelo *framework* Leaflet com OpenStreetMap.

O processamento dos dados ocorre em algumas etapas. Primeiramente, identificando se o pacote contém a coordenada proveniente do GPS. Então a próxima etapa verifica se 3 ou mais *gateways* receberam o mesmo pacote. Se sim, é necessário a conversão das localizações angulares dos *gateways*, latitude e longitude, para o sistema cartesiano. Isto é feito utilizando o caso especial *plate carrée* da projeção equirretangular (KUMLER, 1994), cuja equação pode ser vista na Equação 8, onde R é o raio do planeta Terra, lng e lat longitude e latitude a serem convertidas, x e y resultado da conversão, e lng_0 e lat_0 a latitude e longitude do ponto central do mapa.

$$\begin{aligned} x &= R(lng - lng_0) \\ y &= R(lat - lat_0) \end{aligned} \tag{8}$$

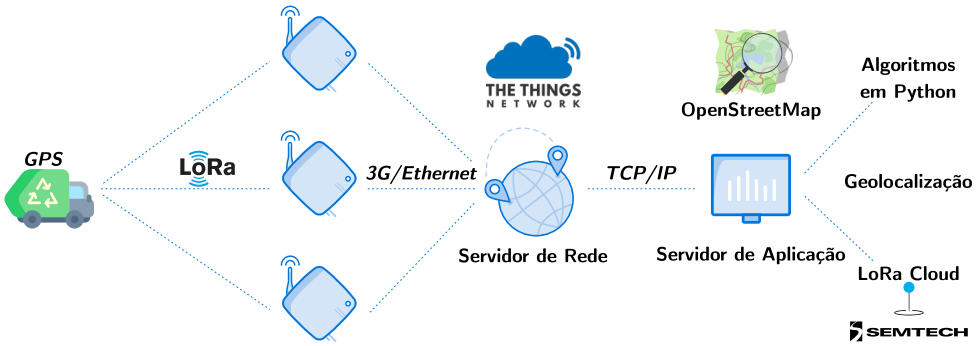
O algoritmo que recebe as três coordenadas pode realizar o calculo de geolocalização com as equações de TDoA (de acordo com Equação 7, Seção 3.2.2) e RSSI (de acordo com

Figura 12 – Fluxo de execução da proposta.



Fonte: Autoria própria (2021).

Figura 13 – Topologia da rede LoRaWAN.



Fonte: Autoria própria (2021).

Equação 4, Seção 3.2.1). O algoritmo faz uso da biblioteca SciPy para realizar a aproximação numérica dessas equações por mínimos quadrados e uma implementação de série de Taylor pelo método de Foy (FOY, 1976). Esta implementação faz parte do trabalho de (CROWELL, 2018), o qual foi utilizado e alterado durante este trabalho. A esta implementação foram adicionados os métodos de mínimos quadrados e RSSI ao algoritmo.

As coordenadas cartesianas estimadas podem ser revertidas para latitude e longitude e exibidas no mapa. Por fim, para os pacotes que possuem as coordenadas do GPS do dispositivo

no seu *payload* é possível calcular o erro entre GPS e calculado utilizando a fórmula inversa de Haversine, Equação 9 de (NICHAT, 2013), onde R é o raio do planeta Terra, lat e lng são as respectivas latitudes e longitudes dos pontos a serem calculadas as distâncias em radianos, e d a distância resultante.

$$d = 2R \cdot arcsen\sqrt{sen^2(\frac{lat_2 - lat_1}{2}) + cos(lat_1)cos(lat_2)sen^2(\frac{lng_2 - lng_1}{2})} \quad (9)$$

5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Este Capítulo apresenta o ambiente de teste com a metodologia indicada no Capítulo 4. Os resultados incluem uma comparação com o algoritmo de geolocalização implementado em Python com a ferramenta comercial LoRa Cloud. A seguir são descritos o ambiente de teste, os resultados com diferentes fatores de espalhamento e as considerações sobre os resultados.

5.1 AMBIENTE DE TESTES

Os *gateways* foram posicionados em áreas externas de duas residências localizadas em um bairro da cidade de Toledo, de maneira a tentar cobrir a maior área entre as residências. As posições das antenas de cada *gateway* podem ser vista na Figura 14. Percebe-se pelas imagens que as antenas não puderam ser posicionadas perfeitamente, altura baixa e obstáculos ao redor. Na Figura 15 estão presentes as localizações dos *gateways* em azul e o caminho a ser percorrido durante os testes em vermelho. Nesta imagem pode se perceber que os *gateways* 1 e 2 estão na mesma residência. Inicialmente, os experimentos estavam previstos para contar com 5 *gateways* instalados pela prefeitura para cobrir toda a área urbana do município de Toledo, como parte do acordo de cooperação firmado entre a universidade e o município. No entanto, devido a pandemia COVID-19 os *gateways* foram adquiridos, mas ainda não instalados. Os experimentos foram realizados com *gateways* adquiridos pela prefeitura. A instalação do *firmware* e configuração dos *gateways* foi realizada como parte do experimento. Os *gateways* contam ainda com um GPS cada, item essencial para sincronizar o tempo entre os equipamentos.

Figura 14 – Posições das antenas dos *gateways* durante os testes.

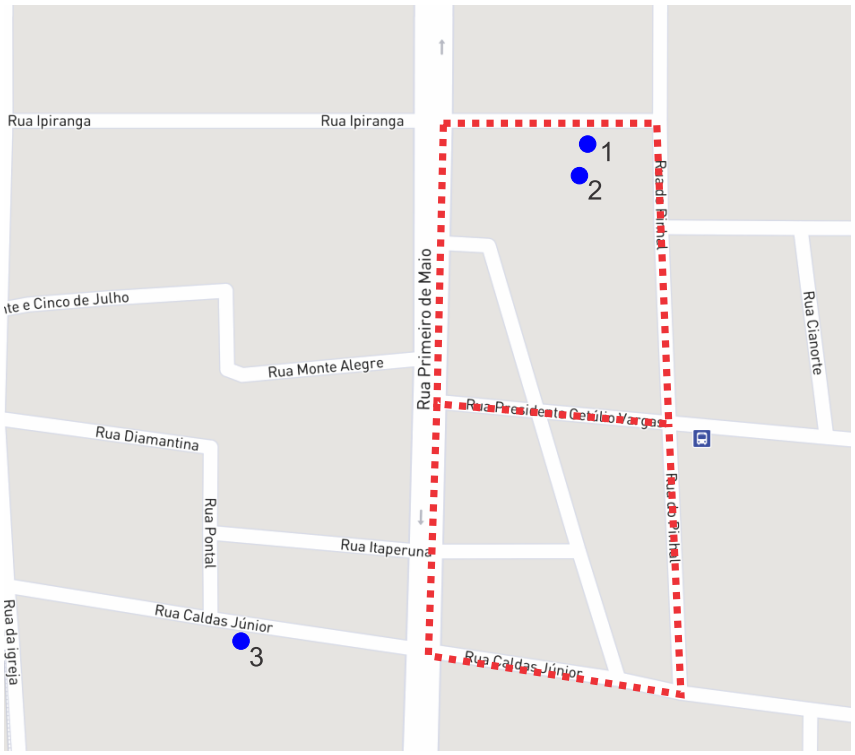
(a) Antena do *gateway* 1.

(b) Antena do *gateway* 2.

(c) Antena do *gateway* 3.

Fonte: Autoria própria (2021).

Figura 15 – Caminho realizado para os testes (vermelho) e localização dos *gateways* (azul).



Fonte: Autoria própria (2021).

O dispositivo T-Beam com o *firmware* de rastreamento, escrito em C++ com LMIC, foi levado a esta área para serem realizados os testes. O T-Beam foi configurado para enviar pacotes a cada 30 segundos. A avaliação aconteceu com 2 SFs diferentes, SF7 e SF10, em momentos distintos. Foram escolhidas duas configurações de SF diferentes para identificar o seu impacto nos resultados. O dispositivo foi levado em caminhadas realizando o percurso da Figura 15 de forma a simular o funcionamento do caminhão, obtendo os dados necessários para estimar a sua posição.

Os dados puderam ser visualizados através da aplicação Web desenvolvida com o auxílio da ferramenta Leaflet e OpenStreetMap. Sendo possível observar os pontos obtidos do GPS do dispositivo, calculados pelos algoritmos de geolocalização ou recebidos da LoRa Cloud. Na Figura 16 tem-se uma visão geral do *layout* da aplicação, com os botões, que alternam a visualização dos pontos, e o mapa, onde são representados os pontos.

Figura 16 – Aplicação Web.

Fonte: Autoria própria (2021).

5.2 RESULTADOS EM SF7

Esta seção apresenta os resultados obtidos com o parâmetro de transmissão em SF7. SF7 é uma configuração indicada para pequenas distâncias possuindo a maior taxa de transmissão de dados dentre as configurações possíveis.

Foram 27 pacotes recebidos, destes somente 9 foram recebidos pelos 3 *gateways*. A partir destes 9 pacotes foi possível estimar a posição do dispositivo, mas o erro só pôde ser calculado em 7, pois continham a posição do dispositivo GPS no *payload*. A Figura 17 mostra todos os pontos obtidos do GPS do dispositivo, mesmo os que foram recebidos por apenas um *gateway*.

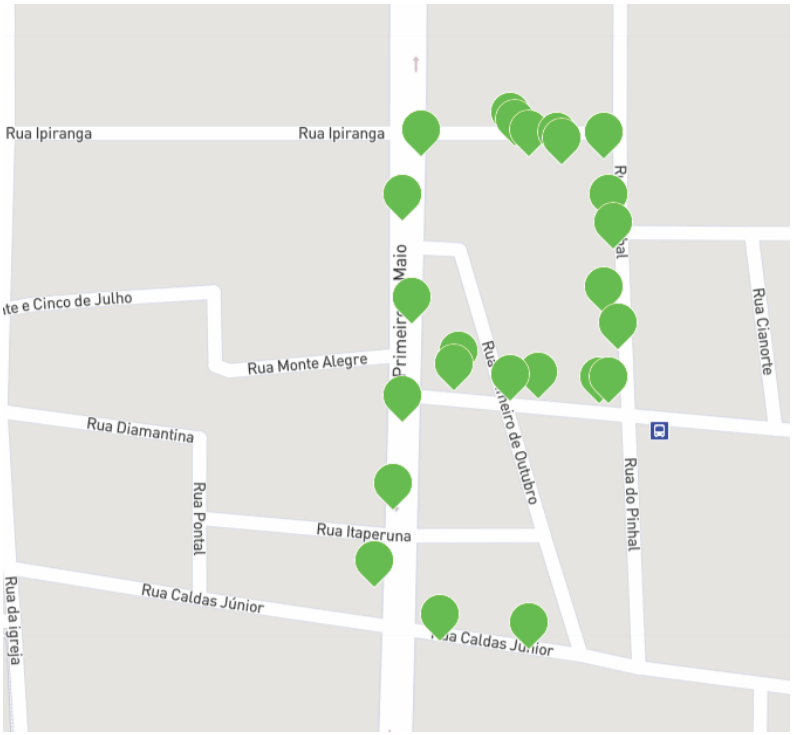
Nas Figuras 18 e 19 é possível ver os pontos calculados por TDoA e RSSI, respectivamente, junto dos pontos do GPS do dispositivo que foram recebidos pelos 3 *gateways*. As figuras apresentam os dois métodos de aproximação utilizados, mínimos quadrados em (a) e série de Taylor em (b). Em vermelho as posições calculadas por TDoA, em roxo por RSSI e em verde as posições recebidas do GPS.

As localizações obtidas da LoRa Cloud podem ser visualizadas na Figura 20 na cor laranja. LoRa Cloud utiliza seus próprios métodos de decisão de algoritmos. É possível especificar qual método a ser utilizado, TDoA, RSSI ou uma combinação dos dois. Mas a plataforma escolhe o método que considera ter maior precisão independente do especificado.

Buscando os resultados numéricos das posições calculadas foi realizada a média do erro encontrado em cada amostra, conforme Equação 9 Seção 4.3. As Tabelas 2 e 3 apresentam os resultados obtidos através de TDoA e RSSI, respectivamente, com a transmissão de dados em

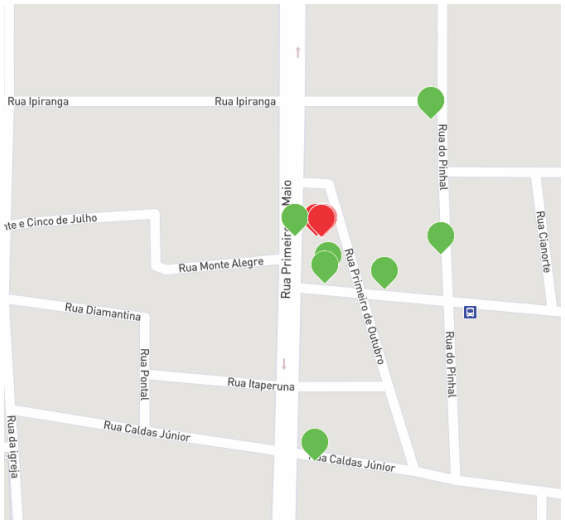
SF7. As tabelas comparam a precisão média dos resultados, junto da melhor e pior precisão em cada modelo de aproximação numérica, mínimos quadrados e série de Taylor. Percebendo-se que em TDoA a precisão foi maior utilizando mínimos quadrados, já em RSSI a aproximação por série de Taylor apresentou uma média e pior caso melhores.

Figura 17 – Posições do dispositivo recebidas do GPS em SF7.

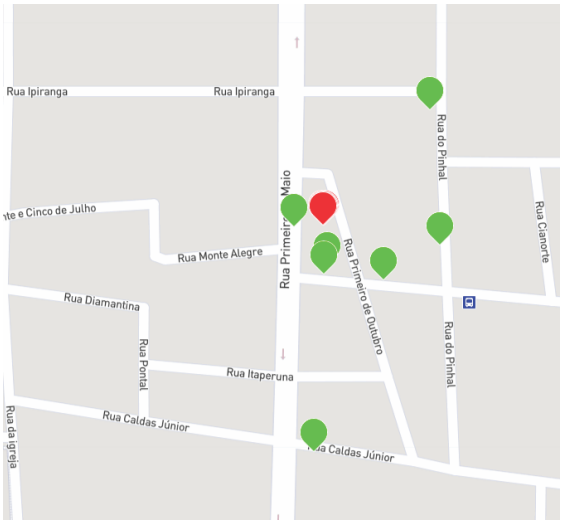


Fonte: Autoria própria (2021).

Figura 18 – Posições por TDoA (vermelho) e GPS (verde) em SF7.



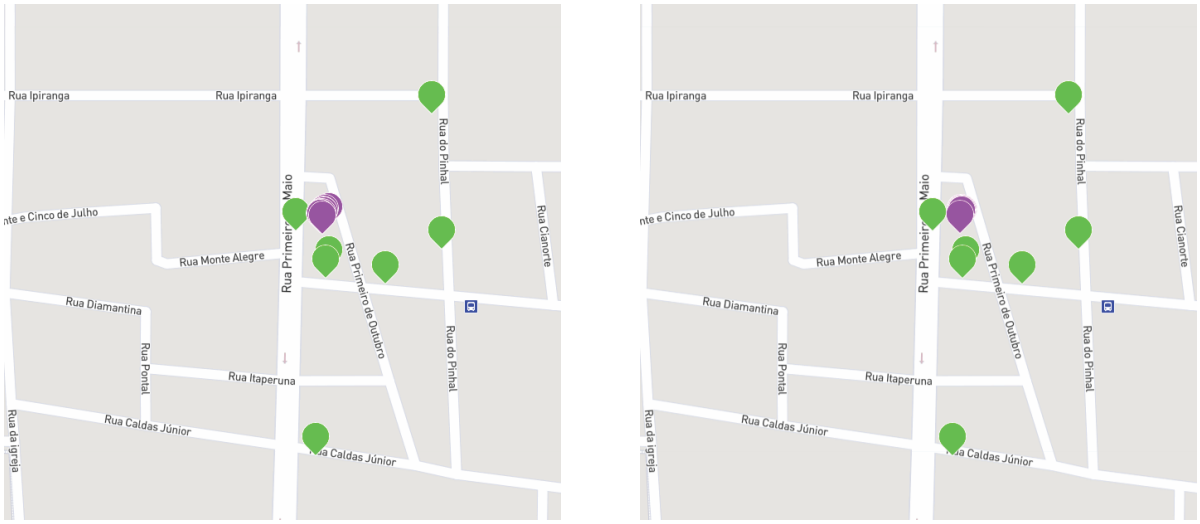
(a) Posições TDoA por mínimos quadrados.



(b) Posições TDoA por série de Taylor.

Fonte: Autoria própria (2021).

Figura 19 – Posições por RSSI (roxo) e GPS (verde) em SF7.

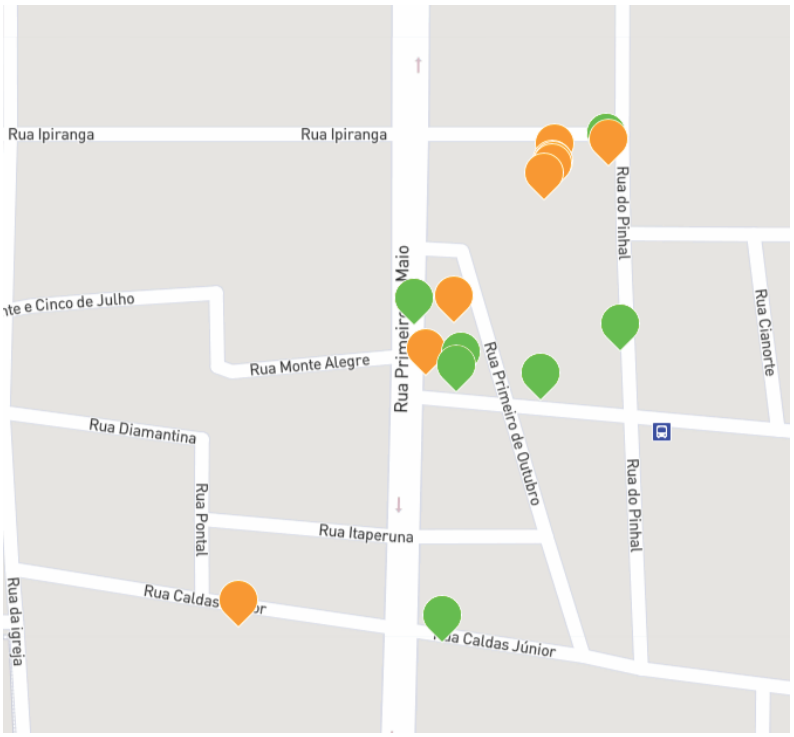


(a) Posições RSSI por mínimos quadrados.

(b) Posições RSSI por série de Taylor.

Fonte: Autoria própria (2021).

Figura 20 – Posições por LoRa Cloud (laranja) e GPS (verde) em SF7.



Fonte: Autoria própria (2021).

A Tabela 4 compara os resultados obtidos em TDoA e RSSI com os recebidos da plataforma LoRa Cloud. É apontada a melhor média, a melhor e pior precisão dos resultados calculados comparando da mesma forma os calculados pela LoRa Cloud. Observa-se que RSSI teve a melhor média, mas o melhor caso ocorre em TDoA com uma precisão de 18,22 metros.

Tabela 2 – Precisão do algoritmo de TDoA em SF7 em metros.

	Mínimos quadrados	Série de Taylor
Média	65,47	65,83
Melhor	18,22	19,89
Pior	147,08	148,89

Fonte: Autoria própria (2021).

Tabela 3 – Precisão do algoritmo de RSSI em SF7 em metros.

	Mínimos quadrados	Série de Taylor
Média	66,66	65,40
Melhor	18,38	19,89
Pior	150,41	145,40

Fonte: Autoria própria (2021).

Tabela 4 – Comparação dos resultados em SF7 em metros.

	TDoA	RSSI	LoRa Cloud
Média	65,47	65,40	79,60
Melhor	18,22	18,38	28,55
Pior	148,89	150,41	123,83

Fonte: Autoria própria (2021).

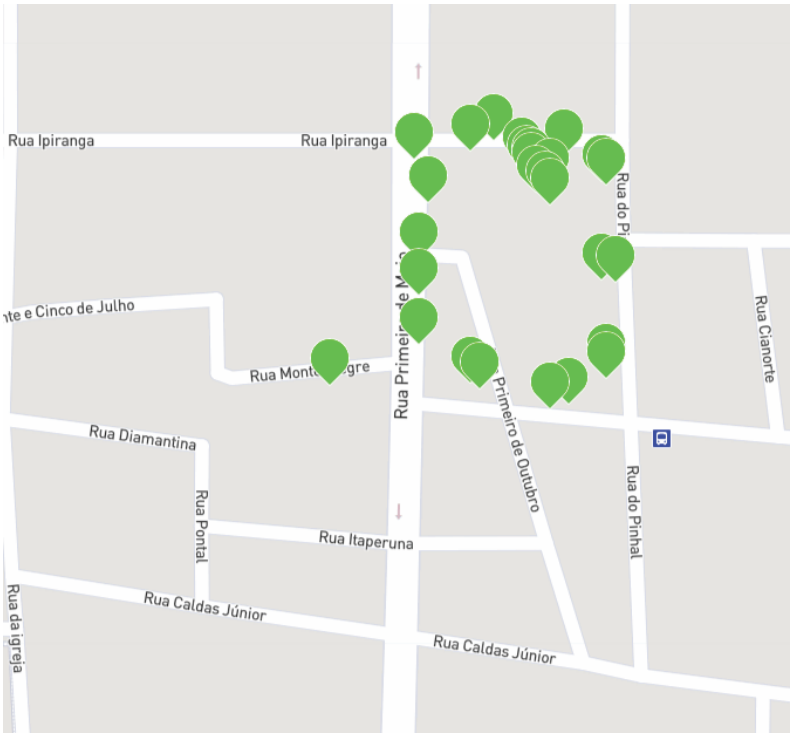
Os resultados apresentaram uma boa precisão no melhor caso, chegando a aproximadamente 19 metros em RSSI e TDoA. No pior caso os resultados apresentam precisão aproximada de 149 e 151 metros para TDoA e RSSI, respectivamente. Realizando uma comparação visual através dos mapas, observa-se que os pontos calculados pelo algoritmo de aproximação estão sobrepostos. Ao contrário, os pontos calculados pela LoRa Cloud estão mais dispersos. Tal fato indica um possível enviesamento dos cálculos devido a posição dos *gateways*.

5.3 RESULTADOS EM SF10

Os resultados obtidos em SF10 são apresentados nesta seção. SF10 é uma configuração de transmissão intermediária, indicada para médias distâncias com uma taxa de transmissão menor do que em SF7. Tendo em consideração esta característica, foram 78 pacotes destes, 20 foram recebidos pelos 3 *gateways*. Os pontos obtidos do GPS do dispositivo podem ser vistos na Figura 21, apresentando a localização do dispositivo obtida através de seu GPS.

As localizações obtidas dos cálculos de geolocalização podem ser vistas nas Figuras 22, TDoA em vermelho, e 23, RSSI em roxo, junto das posições recebidas pelo GPS, em verde. Da mesma forma apresentada nos resultados de SF7, em (a) aproximação por mínimos quadrados e

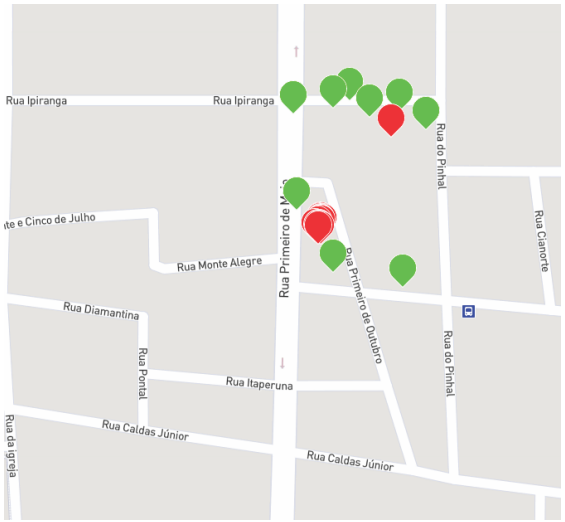
Figura 21 – Posições do dispositivo recebidas do GPS em SF10.



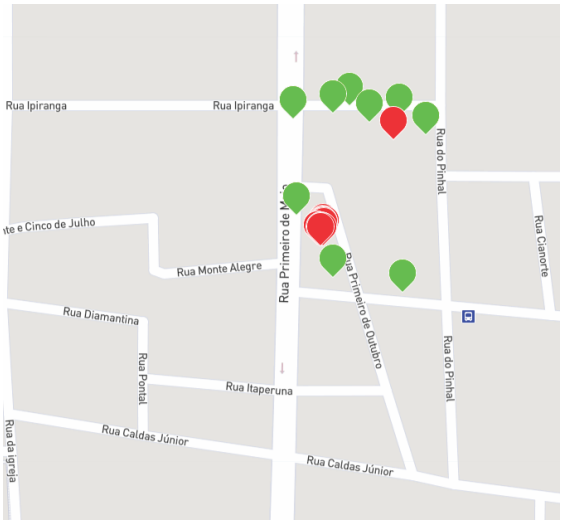
Fonte: Autoria própria (2021).

em (b) por série de Taylor. Já na Figura 24 podem ser visualizados os pontos recebidos da LoRa Cloud, em alaranjado, e do GPS, em verde.

Figura 22 – Posições por TDoA (vermelho) e GPS (verde) em SF10.



(a) Posições TDoA por mínimos quadrados.

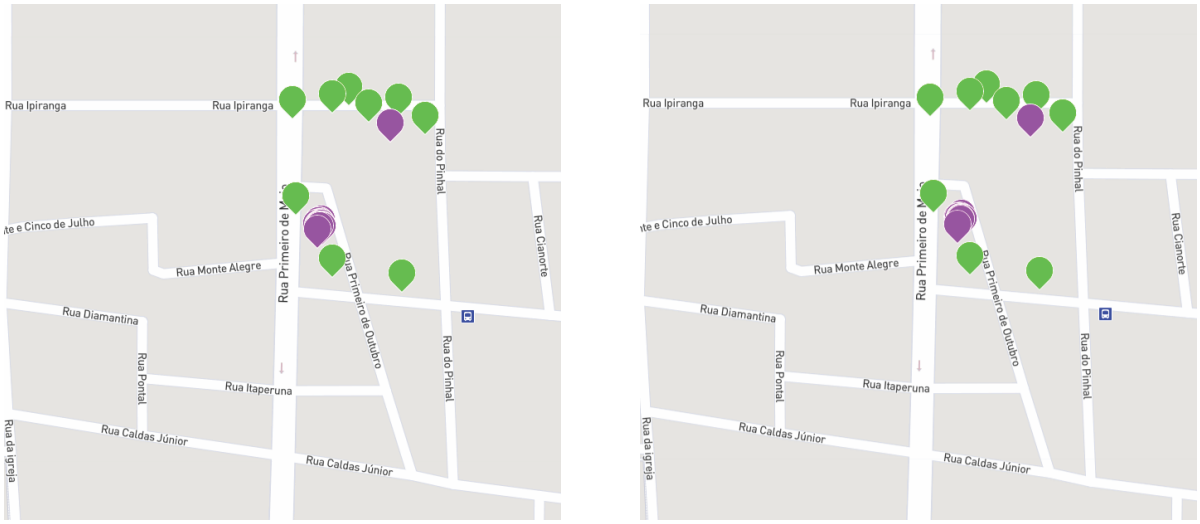


(b) Posições TDoA por série de Taylor.

Fonte: Autoria própria (2021).

A média dos erros, melhor e pior caso, são apresentados na Tabela 5 para TDoA e na Tabela 6 para RSSI. Nesta configuração, a série de Taylor superou o método de mínimos

Figura 23 – Posições por RSSI (roxo) e GPS (verde) em SF10.

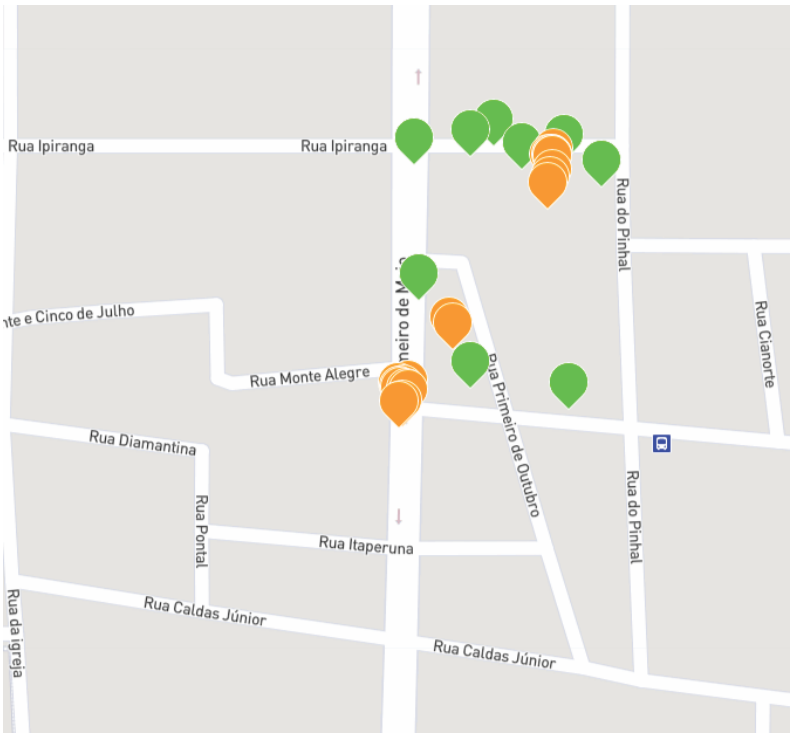


(a) Posições TDoA por mínimos quadrados.

(b) Posições TDoA por série de Taylor.

Fonte: Autoria própria (2021).

Figura 24 – Posições por LoRa Cloud (laranja) e GPS (verde) em SF10.



Fonte: Autoria própria (2021).

quadrados para caso médio e o pior caso tanto para TDoA quanto para RSSI. Já para o melhor caso, a diferença foi pequena em relação a aproximação por mínimos quadrados.

Tabela 5 – Precisão do algoritmo de TDoA em SF10 em metros.

	Mínimos quadrados	Série de Taylor
Média	73,22	71,71
Melhor	23,64	24,85
Pior	98,26	95,90

Fonte: Autoria própria (2021).

Tabela 6 – Precisão do algoritmo de RSSI em SF10 em metros.

	Mínimos quadrados	Série de Taylor
Média	72,79	70,50
Melhor	21,61	22,23
Pior	99,77	96,07

Fonte: Autoria própria (2021).

Os resultados obtidos com os algoritmos de TDoA e RSSI são comparados com os recebidos da LoRa Cloud na Tabela 7. Observa-se que LoRa Cloud obteve os melhores resultados na média e melhor caso. Chegando a uma precisão em cerca de 11 metros. Mas no pior caso o resultado foi inferior aos outros métodos.

Tabela 7 – Comparação dos resultados em SF10 em metros.

	TDoA	RSSI	LoRa Cloud
Média	71,71	70,50	66,14
Melhor	23,64	21,61	10,94
Pior	98,26	99,77	125,81

Fonte: Autoria própria (2021).

5.4 CONSIDERAÇÕES SOBRE OS RESULTADOS

Os resultados dos algoritmos de geolocalização com TDoA e RSSI foram muito semelhantes, tanto em SF7 quanto em SF10. Mesmo ao comparar os resultados de TDoA com RSSI a diferença é pequena, nas médias aproximadamente 1 metro. Um dos fatores que influenciaram nos resultados de geolocalização com TDoA foi a precisão dos relógios dos *gateways*. Como a velocidade de propagação do sinal RF LoRa se compara a velocidade da luz, 299792458 metros por segundo, e com a precisão de tempo dos *gateways* em 1 micro segundo, existe um erro embutido nos cálculos de no mínimo 300 metros. Pois a cada nano segundo 1 metro de erro é adicionado. Este foi um parâmetro descoberto durante experimentação, já que a documentação dos dispositivos não indicavam tal característica.

Mesmo com a baixa precisão de tempo dos *gateways* não foram identificados erros maiores que 151 metros. Acontece que como 2 dos 3 *gateways* se encontravam relativamente próximos, muitas vezes o tempo de chegada do pacote era o mesmo, fazendo com que o sistema de equações tivesse praticamente, uma equação a menos. Em outros casos, o tempo de chegada era muito parecido em todos os *gateways*. Dessa forma, as equações aproximavam-se muito a equação de uma circunferência, explicando o motivo dos cálculos resultarem sempre em um ponto central do mapa.

No caso dos cálculos por RSSI, identificou-se pouca influência do fator *n* presente no modelo apresentado. Tendo sido alterado durante experimentação e verificando os mesmos resultados. Mas no caso do RSSI, era esperado um comportamento parecido com o obtido, devido ao posicionamento dos *gateways*.

Em se tratando da LoRa Cloud, apesar de ter sido especificado nas requisições o cálculo de geolocalização por TDoA, a maior parte dos resultados retornaram como RSSI. Apesar disso, alguns pontos apresentaram uma boa precisão, chegando em até 10 metros de precisão durante a transmissão em SF10. Avaliando os demais resultados, em SF7 LoRa Cloud apresentou um erro médio maior, se comparado com os demais algoritmos, mas um intervalo de erro menor, entre o melhor e o pior. Comportamento contrário ao que ocorreu em SF10, em que a precisão média da LoRa Cloud é a melhor.

6 CONCLUSÃO

Foram estudados os conceitos de IoT e tolerância a falhas, e das principais técnicas de tolerância a falhas utilizadas em IoT. A revisão sistemática, presente neste trabalho, e que serviu de base para o desenvolvimento do TCC foi desenvolvida em um contexto de iniciação científica durante o período de um ano foi submetida e aceita no X Simpósio Brasileiro de Engenharia de Sistemas de Computação (SBESC) (PASTÓRIO; CAMARGO; RODRIGUES, 2020).

As técnicas de geolocalização foram estudadas sendo encontradas diversas técnicas e formas de se estimar a posição de um objeto. Podendo ser utilizadas em vários âmbitos diferentes, por exemplo LoRa e Wi-Fi. Dentro do contexto de LoRa e LoRaWAN duas se destacaram, TDoA e RSSI, e foram objetos de estudo deste trabalho.

Utilizando os conhecimentos adquiridos foi proposta uma abordagem tolerante a falhas em um sistema de rastreamento. Onde, através de envios periódicos de mensagens o dispositivo pode ser rastreado por GPS e, no caso de falha do GPS, a sua posição pode ser estimada através dos algoritmos de aproximação implementados e pela ferramenta comercial LoRa Cloud. Esta redundância foi implementada em uma aplicação Web, de forma a visualizar os dados.

O trabalho teve o objetivo de propor uma solução tolerante a falhas para um sistema de rastreamento IoT. Avaliando os resultados obtidos foi identificado que utilizar técnicas de geolocalização pode ser uma alternativa para o rastreamento de um objeto com precisão de até 151 metros. Dessa forma, aplicações em que a posição do objeto não precisa ser extremamente exata, podem utilizar desta proposta para adquirir tolerância a falhas em seu funcionamento.

6.1 TRABALHOS FUTUROS

Este trabalho obteve resultados promissores e que podem ser estudados com mais profundidade em trabalhos futuros. São diversos os caminhos a serem seguidos a partir daqui. Um deles é repetir os experimentos quando os *gateways* estiverem instalados na área urbana do município para comparar com os resultados deste trabalho. Dessa forma haverá a possibilidade de coletar muito mais pontos e avaliar melhor a precisão dos métodos.

Um trabalho como (PODEVIJN et al., 2018a) pode ser replicado buscando melhorias. O trabalho realiza o ajuste das localizações estimadas de acordo com o que se espera. Por exemplo, se está sendo rastreado um veículo espera-se que as posições estimadas, e recebidas, estejam localizadas em alguma rua ou rodovia.

Uma solução parecida com a proposta neste trabalho pode ser implementada utilizando a fórmula inversa de Haversine. As equações de TDoA e RSSI são basicamente cálculos de distâncias, substituir o termo em raiz quadrada pela fórmula inversa de Haversine permite a aproximação numérica diretamente pela latitude e longitude, sem necessidade de conversão para coordenadas cartesianas. Isto pode vir a aumentar a precisão do algoritmo, pois evita a

propagação do erro entre conversões.

Outra alternativa é o uso de aprendizado de máquina para realizar os cálculos de geolocalização. A rede neural pode ser treinada utilizando os dados de ToA, RSSI e até mesmo as posições recebidas e estimadas anteriormente para calcular a posição do emissor. Realizado o treinamento em um ambiente de testes pode ser avaliada sua eficiência em outros cenários.

Referências

AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys Tutorials**, v. 17, n. 4, p. 2347–2376, 2015. Citado 3 vezes nas páginas 1, 2 e 6.

ALMEIDA, T. V. O. et al. Avaliação de dispositivos de rastreamento em uma rede lorawan no contexto de cidades inteligentes. In: **Anais do IV Workshop de Computação Urbana**. Porto Alegre, RS, Brasil: SBC, 2020. p. 1–14. ISSN 2595-2706. Disponível em: <https://sol.sbc.org.br/index.php/courb/article/view/12349>. Citado na página 25.

ARDEKANI, M. S. et al. Rivulet: A fault-tolerant platform for smart-home applications. In: **Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference**. New York, NY, USA: Association for Computing Machinery, 2017. (Middleware '17), p. 41–54. ISBN 9781450347204. Citado na página 14.

AVIZIENIS, A. et al. Basic concepts and taxonomy of dependable and secure computing. **IEEE Trans. Dep. and Secure Computing**, v. 1, n. 1, p. 11–33, 2004. Citado 2 vezes nas páginas 2 e 5.

BAI, H.; XIA, G.; FU, S. A two-layer-consensus based blockchain architecture for iot. In: **IEEE 9th ICEIEC**. [S.l.: s.n.], 2019. p. 1–6. Citado 2 vezes nas páginas 16 e 17.

BAIN. **Unlocking Opportunities in the Internet of Things**. 2018. Online. Acesso em 23/06/2019. Disponível em: <https://www.bain.com/insights/unlocking-opportunities-in-the-internet-of-things/>. Citado na página 1.

BISSETT, D. **Analysing TDoA Localisation in LoRa Networks**. Dissertação (Mestrado) — Delft University of Technology, 10 2018. Citado na página 21.

CELESTI, A. et al. A watchdog service making container-based micro-services reliable in iot clouds. In: **IEEE 5th FiCloud**. [S.l.: s.n.], 2017. p. 372–378. Citado 2 vezes nas páginas 10 e 12.

CENTENARO, M. et al. Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios. **IEEE Wirel. Comm.**, v. 23, n. 5, p. 60–67, Oct. 2016. ISSN 1558-0687. Citado na página 1.

CHAKRABORTY, T. et al. Fall-curve: A novel primitive for iot fault detection and isolation. In: **16th ACM SenSys**. New York, NY, USA: ACM, 2018. (SenSys '18), p. 95–107. ISBN 9781450359528. Citado 2 vezes nas páginas 11 e 12.

CHILIPIREA, C. et al. Energy efficiency and robustness for iot: Building a smart home security system. In: **IEEE ICCP**. [S.l.: s.n.], 2016. p. 43–48. Citado na página 11.

CHUDZIKIEWICZ, J.; FURTAK, J.; ZIELINSKI, Z. Fault-tolerant techniques for the internet of military things. In: **2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)**. [S.l.: s.n.], 2015. p. 496–501. Citado 2 vezes nas páginas 11 e 12.

CISCO. The internet of things reference model. In: **WhitePaper**. [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 6 e 7.

CROWELL, C. **LoRaWAN Geolocalization using TDOA**. Dissertação (Mestrado) — University of New Hampshire, 10 2018. Disponível em: <https://github.com/ccrowell44/LoRa-Geolocation>. Citado na página 28.

CRUZ, M. A. A. da et al. A reference model for internet of things middleware. **IEEE Internet of Things Journal**, v. 5, n. 2, p. 871–883, 2018. Citado na página 6.

CUNHA, R. de L.; VITOR, G. B.; NETO, A. de M. Estimation of mobile point distance based on the signal strength of wireless network routers. **IEEE Latin America Transactions**, v. 18, n. 01, p. 120–129, 2020. Citado na página 23.

DAI, L. et al. A nature-inspired node deployment strategy for connected confident information coverage in industrial internet of things. **IEEE Internet of Things Journal**, v. 6, n. 6, p. 9217–9225, 2019. Citado 2 vezes nas páginas 12 e 14.

FAN, K. et al. Failure resilient routing via iot networks. In: **IEEE iThings/GreenCom/CPSCom/SmartData**. [S.l.: s.n.], 2017. p. 845–850. Citado na página 13.

FARGAS, B. C.; PETERSEN, M. N. Gps-free geolocation using lora in low-power wans. In: **2017 Global Internet of Things Summit (GIoTS)**. [S.l.: s.n.], 2017. p. 1–6. Citado 3 vezes nas páginas 3, 20 e 23.

FERNÁNDEZ-CARAMÉS, T. M.; FRAGA-LAMAS, P. A review on the use of blockchain for the internet of things. **IEEE Access**, v. 6, p. 32979–33001, 2018. Citado na página 16.

FOY, W. H. Position-location solutions by taylor-series estimation. **IEEE Transactions on Aerospace and Electronic Systems**, AES-12, n. 2, p. 187–194, 1976. Citado na página 28.

FURQUIM, G. et al. How to improve fault tolerance in disaster predictions: A case study about flash floods using iot, ml and real data. **Sensors (Basel, Switzerland)**, MDPI AG, v. 18, n. 3, p. 907, 2018. ISSN 1424-8220. Citado na página 15.

GIA, T. N. et al. Fault tolerant and scalable iot-based architecture for health monitoring. In: **2015 IEEE Sensors Applications Symposium (SAS)**. [S.l.: s.n.], 2015. p. 1–6. Citado na página 15.

GOTTSCHO, M. et al. Low-cost memory fault tolerance for iot devices. **ACM Trans. Embed. Comput. Syst.**, Association for Computing Machinery, New York, NY, USA, v. 16, n. 5s, set. 2017. ISSN 1539-9087. Citado 2 vezes nas páginas 11 e 12.

GROVER, J.; GARIMELLA, R. M. Reliable and fault-tolerant iot-edge architecture. In: **2018 IEEE SENSORS**. [S.l.: s.n.], 2018. p. 1–4. Citado 2 vezes nas páginas 2 e 14.

GU, C.; JIANG, L.; TAN, R. Lora-based localization: Opportunities and challenges. **CoRR**, abs/1812.11481, 2018. Disponível em: <http://arxiv.org/abs/1812.11481>. Citado 2 vezes nas páginas 3 e 20.

GUTIERREZ, M. D. et al. Low cost error monitoring for improved maintainability of iot applications. In: **IEEE DFT**. [S.l.: s.n.], 2017. p. 1–6. Citado 2 vezes nas páginas 11 e 12.

GUTIÉRREZ-RIVAS, J. L. et al. White rabbit hsr: A seamless subnanosecond redundant timing system with low-latency data capabilities for the smart grid. **IEEE Trans. Ind. Informatics**, v. 14, n. 8, p. 3486–3494, 2018. Citado na página 12.

HASAN, M. Z.; AL-TURJMAN, F. Optimizing multipath routing with guaranteed fault tolerance in internet of things. **IEEE Sensors Journal**, v. 17, n. 19, p. 6463–6473, 2017. Citado na página 13.

HASAN, M. Z.; AL-TURJMAN, F. Swarm-based data delivery framework in the ad hoc internet of things. In: **GLOBECOM**. [S.l.: s.n.], 2017. p. 1–6. Citado 2 vezes nas páginas 13 e 14.

ISLAM, S. M. R. et al. The internet of things for health care: A comprehensive survey. **IEEE Access**, v. 3, p. 678–708, 2015. Citado na página 1.

JALOTE, P. **Fault Tolerance in Distributed Systems**. USA: Prentice-Hall, Inc., 1994. ISBN 0133013677. Citado na página 6.

KAIWARTYA, O. et al. Virtualization in wireless sensor networks: Fault tolerant embedding for internet of things. **IEEE Internet of Things Journal**, v. 5, n. 2, p. 571–580, 2018. Citado 2 vezes nas páginas 12 e 14.

KHAN, R. et al. Future internet: The internet of things architecture, possible applications and key challenges. In: **10th Int'l Conf. on Frontiers of Inf. Technology**. [S.l.: s.n.], 2012. p. 257–260. Citado na página 7.

KHELIFI, F. et al. A survey of localization systems in internet of things. **Mobile Networks and Applications**, v. 24, n. 3, p. 761–785, Jun 2019. ISSN 1572-8153. Disponível em: <https://doi.org/10.1007/s11036-018-1090-3>. Citado 3 vezes nas páginas 1, 3 e 20.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele Univ.**, v. 33, 08 2004. Citado na página 8.

KODESWARAN, P. A. et al. Idea: A system for efficient failure management in smart iot environments. In: **Proceedings of the MobiSys**. New York, NY, USA: Association for Computing Machinery, 2016. p. 43–56. ISBN 9781450342698. Citado na página 15.

KUMLER, M. Flattening the earth: Two thousand years of map projections. **Cartographic Perspectives**, n. 18, p. 32–33, Jun. 1994. Disponível em: <https://cartographicperspectives.org/index.php/journal/article/view/cp18-kumler>. Citado na página 27.

LAM, K.; CHEUNG, C.; LEE, W. Rssi-based lora localization systems for large-scale indoor and outdoor environments. **IEEE Transactions on Vehicular Technology**, v. 68, n. 12, p. 11778–11791, 2019. Citado na página 3.

LAMPORT, L.; LYNCH, N. Distributed computing: Models and methods. In: VAN LEEUWEN, J. (Ed.). **Formal Models and Semantics**. Amsterdam: Elsevier, 1990, (Handbook of Theoretical Computer Science). p. 1157 – 1199. ISBN 978-0-444-88074-1. Citado na página 6.

LEKA, E.; SELIMI, B.; LAMANI, L. Systematic literature review of blockchain applications: Smart contracts. In: **2019 International Conference on Information Technologies (InfoTech)**. [S.l.: s.n.], 2019. p. 1–3. Citado na página 16.

LI, Y. et al. Towards location enhanced iot: Characterization of lora signal for wide area localization. In: **2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)**. [S.l.: s.n.], 2018. p. 1–7. Citado na página 3.

LILYGO. **LILYGO**. 2020. Acesso em 22/09/2020. Disponível em: <http://www.lilygo.cn/>. Citado na página 24.

LIN, J. et al. Efficient fault-tolerant routing in iot wireless sensor networks based on bipartite-flow graph modeling. **IEEE Access**, v. 7, p. 14022–14034, 2019. Citado 2 vezes nas páginas 13 e 14.

LORA ALLIANCE. **LoRaWAN: What is it? A technical overview of LoRa and LoRaWAN**. 2015. Acesso em 22/09/2020. Disponível em: <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>. Citado 2 vezes nas páginas 18 e 19.

LORA ALLIANCE, S. C. **GEOLOCATION WHITEPAPER**. 2018. Acesso em 22/09/2020. Disponível em: <https://lora-alliance.org/sites/default/files/2018-04/geolocation_whitepaper.pdf>. Citado na página 3.

MACEDO, D.; GUEDES, L. A.; SILVA, I. M. D. A dependability evaluation for internet of things incorporating redundancy aspects. **Proc. 11th IEEE Int'l Conf. Net., Sensing and Control**, p. 417–422, 2014. Citado na página 2.

MACKEY, A.; SPACHOS, P. Lora-based localization system for emergency services in gps-less environments. In: **IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. [S.l.: s.n.], 2019. p. 939–944. Citado na página 3.

MANSFIELD, T. O.; GHITA, B. V.; AMBROZE, M. A. Signals of opportunity geolocation methods for urban and indoor environments. **Annals of Telecommunications**, v. 72, n. 3, p. 145–155, Apr 2017. ISSN 1958-9395. Disponível em: <https://doi.org/10.1007/s12243-016-0559-y>. Citado na página 3.

MANZONI, P. et al. Indoor vehicles geolocalization using lorawan. **Future Internet**, MDPI AG, v. 11, n. 6, p. 124, May 2019. ISSN 1999-5903. Disponível em: <http://dx.doi.org/10.3390/fi11060124>. Citado 4 vezes nas páginas 3, 20, 21 e 23.

MCCI. **Arduino LoRaWAN MAC in C (LMIC)**. 2019. Acesso em 22/09/2020. Disponível em: <https://github.com/mcci-catena/arduino-lmic/blob/master/doc/LMIC-v3.0.99.pdf>. Citado na página 26.

MOSTAFA, B. et al. An energy-efficient multiobjective scheduling model for monitoring in internet of things. **IEEE Internet of Things Journal**, v. 5, n. 3, p. 1727–1738, 2018. Citado 2 vezes nas páginas 13 e 14.

NASSER, N. et al. Routing in the internet of things. In: **GLOBECOM 2017 - 2017 IEEE Global Communications Conference**. [S.l.: s.n.], 2017. p. 1–6. Citado 3 vezes nas páginas 2, 13 e 14.

NICHAT, M. Landmark based shortest path detection by using a* algorithm and haversine formula. 04 2013. Citado na página 29.

OLIVEIRA, L. de; CONCEIÇÃO, A. da; NETO, L. S. Revisão sistemática da literatura sobre aplicações das tecnologias lora e lorawan. In: **Anais Estendidos do VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais**. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/11002>. Citado na página 18.

O'KEEFE, B. Finding location with time of arrival and time difference of arrival techniques. **ECE Senior Capstone Project**, 2017. Citado na página 21.

PARK, J. All-terminal reliability analysis of wireless networks of redundant radio modules. **IEEE Internet of Things Journal**, v. 3, n. 2, p. 219–230, 2016. Citado na página 12.

PASTÓRIO, A. F.; CAMARGO, E. T.; RODRIGUES, L. A. 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**. Porto Alegre, RS, Brasil: SBC, 2020. p. 57–64. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13091>. Citado na página 40.

PILLAI, A. S. et al. A service oriented iot architecture for disaster preparedness and forecasting system. **Internet of Things**, p. 100076, 2019. ISSN 2542-6605. Citado na página 15.

PODEVIJN, N. et al. Tdoa-based outdoor positioning with tracking algorithm in a public lora network. **Wireless Communications and Mobile Computing**, Hindawi, v. 2018, p. 1864209, May 2018. ISSN 1530-8669. Disponível em: <https://doi.org/10.1155/2018/1864209>. Citado 5 vezes nas páginas 2, 3, 22, 23 e 40.

PODEVIJN, N. et al. Tdoa-based outdoor positioning in a public lora network. In: **12th European Conference on Antennas and Propagation (EuCAP 2018)**. [S.l.: s.n.], 2018. p. 1–4. Citado 3 vezes nas páginas 1, 3 e 22.

POWER, A.; KOTONYA, G. A microservices architecture for reactive and proactive fault tolerance in iot systems. In: **IEEE 19th WoWMoM**. [S.l.: s.n.], 2018. p. 588–599. Citado 2 vezes nas páginas 2 e 14.

POWER, A.; KOTONYA, G. Complex patterns of failure: Fault tolerance via complex event processing for iot systems. In: **IEEE iThings/GreenCom/CPSCom/SmartData**. [S.l.: s.n.], 2019. p. 986–993. Citado na página 14.

QAIM, W. B.; OZKASAP, O. Draw: Data replication for enhanced data availability in iot-based sensor systems. In: **Int'l Conf. DASC/PiCom/DataCom/CyberSciTech**. [S.l.: s.n.], 2018. p. 770–775. Citado 2 vezes nas páginas 10 e 12.

RADIOENGE. **Gateway LoRaWAN**. 2020. Acesso em 22/09/2020. Disponível em: <https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html>. Citado na página 26.

RAZZAQUE, M. A. et al. Middleware for internet of things: A survey. **IEEE Internet of Things Journal**, v. 3, n. 1, p. 70–95, 2016. Citado na página 2.

ROSSATO, J.; SPANHOL, F. A.; CAMARGO, E. T. Implantação e avaliação de uma rede sem-fio de longo alcance e baixa potência para cidades inteligentes. In: **Anais do IV CoUrb / IV Workshop de Computação Urbana**. Porto Alegre, RS, Brasil: SBC, 2020. p. 1–14. Citado na página 4.

SAHNI, Y. et al. Edge mesh: A new paradigm to enable distributed intelligence in internet of things. **IEEE Access**, v. 5, p. 16441–16458, 2017. Citado na página 9.

SANTOS, B. P. et al. Internet das coisas: da teoria à prática. In: SIQUEIRA, F. A. et al. (Ed.). **Livro de Minicursos - SBRC 2016**. Brasil: Sociedade Brasileira de Computação, 2016. cap. 1, p. 1–50. Citado na página 8.

SHARMA, S. et al. Iot based innovative dual level control system with fault tolerance fail safe capability. In: **ICICCS**. [S.l.: s.n.], 2018. p. 307–312. Citado na página 15.

STATISTA. **Forecast end-user spending on IoT solutions worldwide from 2017 to 2025**. 2020. Acesso em 30/03/2020. Disponível em: <https://www.statista.com/statistics/976313/global-iot-market-size/>. Citado na página 1.

STEFAŃSKI, J. Hyperbolic position location estimation in the multipath propagation environment. In: WOZNIAK, J. et al. (Ed.). **Wireless and Mobile Networking**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 232–239. Citado na página 22.

TARAI, S. K.; SHAILENDRA, S. Optimal and secure controller placement in sdn based smart city network. In: **2019 International Conference on Information Networking (ICOIN)**. [S.l.: s.n.], 2019. p. 254–261. Citado na página 9.

THE THINGS NETWORK. **The Things Network (TTN)**. 2020. Acesso em 22/09/2020. Disponível em: <https://www.thethingsnetwork.org/>. Citado na página 25.

TRIVEDI, K. S.; BOBBIO, A.; MUPPALA, J. **Reliability and Availability Engineering: Modeling, Analysis, and Applications**. Cambridge: Cambridge University Press, 2017. ISBN 978-1-316-16304-7. Citado 2 vezes nas páginas 5 e 6.

TZOUNIS, A. et al. Internet of things in agriculture, recent advances and future challenges. **Biosystems Engineering**, v. 164, p. 31 – 48, 2017. ISSN 1537-5110. Citado na página 1.

VXCHNGE. **Comprehensive Guide to IoT Statistics You Need to Know in 2020**. 2020. Acesso em 30/03/2020. Disponível em: <https://www.vxchnge.com/blog/iot-statistics>. Citado na página 1.

WANG, S. et al. Reaching agreement in an integrated fog cloud iot. **IEEE Access**, v. 6, p. 64515–64524, 2018. Citado na página 16.

WANG, S.-C.; LIN, Y.-J.; YAN, K.-Q. Reaching byzantine agreement underlying vanet. **KSII Transactions on Internet and Information Systems**, KSII, the Korean Society for Internet Information, v. 13, n. 7, p. 3351, 2019. ISSN 1976-7277. Citado na página 17.

WHITMORE, A.; AGARWAL, A.; XU, L. The internet of things–a survey of topics and trends. **Information Systems Frontiers**, Kluwer Academic Publishers, USA, v. 17, n. 2, p. 261–274, abr. 2015. ISSN 1387-3326. Citado na página 1.

WU, M. et al. Research on the architecture of internet of things. In: **3rd ICACTE**. [S.l.: s.n.], 2010. v. 5, p. V5–484–V5–487. Citado na página 7.

XU, J.-w. et al. SIoTFog: Byzantine-resilient IoT fog networking. **Frontiers of Information Technology & Electronic Engineering**, v. 19, n. 12, p. 1546–1557, dez. 2018. ISSN 2095-9230. Citado na página 15.

XU, T.; POTKONJAK, M. Energy-efficient fault tolerance approach for internet of things applications. In: **2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)**. [S.l.: s.n.], 2016. p. 1–8. Citado 2 vezes nas páginas 11 e 12.

YOON, W.; CHOI, I.; KIM, D. Blockons: Blockchain based object name service. In: **IEEE ICBC**. [S.l.: s.n.], 2019. p. 219–226. Citado na página 16.

YU, Y. et al. Lvchain: A lightweight and vote-based blockchain for access control in the iot. In: **IEEE 4th ICCC**. [S.l.: s.n.], 2018. p. 870–874. Citado na página 16.

ZHOU, S. et al. Supporting service adaptation in fault tolerant internet of things. In: **IEEE SOCA**. [S.l.: s.n.], 2015. p. 65–72. Citado 2 vezes nas páginas 11 e 12.

ZHOU, S.; LIN, K.; SHIH, C. Device clustering for fault monitoring in internet of things systems. In: **IEEE WF-IoT**. [S.l.: s.n.], 2015. p. 228–233. Citado 2 vezes nas páginas 13 e 14.