UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

MESTRADO ACADÊMICO EM SISTEMAS DE COMUNICAÇÃO

ÍTALO ROSSI ARAÚJO COSTA

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO PARA REDES FTTH UTILIZANDO O PROTOCOLO LORAWAN E A APLICAÇÃO ZABBIX

SOBRAL

2024

ÍTALO ROSSI ARAÚJO COSTA

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO PARA REDES FTTH UTILIZANDO O PROTOCOLO LORAWAN E A APLICAÇÃO ZABBIX

Dissertação apresentada ao Curso de Mestrado Acadêmico em Sistemas de Comunicação do Programa de Pós-Graduação em Engenharia Elétrica e de Computação do Centro de Engenharia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica e de Computação. Área de Concentração: Sistemas de Informação

Orientador: Prof. Dr. José Cláudio do Nascimento

SOBRAL

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C876i Costa, Ítalo Rossi Araújo.
IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO PARA REDES FTTH UTILIZANDO O PROTOCOLO LORAWAN E A APLICAÇÃO ZABBIX / Ítalo Rossi Araújo Costa. – 2024.
77 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Campus de Sobral, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Sobral, 2024.
Orientação: Prof. Dr. José Cláudio do Nascimento.

1. LoRaWAN. 2. IoT. 3. FTTH. 4. Sensoriamento. 5. Zabbix. I. Título.

CDD 621.3

ÍTALO ROSSI ARAÚJO COSTA

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO PARA REDES FTTH UTILIZANDO O PROTOCOLO LORAWAN E A APLICAÇÃO ZABBIX

Dissertação apresentada ao Curso de Mestrado Acadêmico em Sistemas de Comunicação do Programa de Pós-Graduação em Engenharia Elétrica e de Computação do Centro de Engenharia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica e de Computação. Área de Concentração: Sistemas de Informação

Aprovada em: 28 de Agosto de 2024

BANCA EXAMINADORA

Prof. Dr. José Cláudio do Nascimento (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Evilásio Costa Júnior
Universidade Federal do Ceará (UFC)

Prof. Dr. Francisco Leonardo Bezerra Martins
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

À Deus primeiramente, que me deu o dom da vida e me direcionou em várias etapas que foram cruciais para a realização desta pesquisa.

Ao meu pai, José Delano, e minha mãe, Lêda Maria, por sempre me apoiarem em minhas decisões e me incentivarem a crescer e me desenvolver.

À minha noiva, Benise Ferreira, que esteve ao meu lado e me deu forças para não desistir em muitos momentos. Agradeço também pelas discussões que geraram contribuições na escrita desta pesquisa.

Aos meus padrinhos, Mário Macedo e Celina Lemos, que me apoiaram muito na realização deste Mestrado. Agradeço também pelos diversos incentivos que viabilizaram o cumprimento dos requisitos do curso, assim como a realização desta pesquisa.

Ao meu orientador, professor Dr. José Cláudio do Nascimento, que me guiou durante toda pesquisa e iniciou discussões que foram muito importantes para o entendimento do projeto. Sua experiência e visão de projeto foram fundamentais para o desenvolvimento desta dissertação.

Aos professores do curso de Pós-Graduação em Engenharia Elétrica e de Computação da Universidade Federal do Ceará, pelos ensinamentos e contribuições com minha formação acadêmica.

Aos meus amigos e colegas de trabalho e da universidade, que me ajudaram em diversos aspectos e contribuíram para que a pesquisa fosse realizada.

À banca pela disponibilização de tempo e atenção para avaliação desta pesquisa e pelas suas contribuições.

Ao Instituto Cearense de Tecnologia, Empreendedorismo e Liderança (ICETEL) por disponibilizar os equipamentos necessários para a realização dos experimentos, assim como pelo laboratório onde foram realizados os testes.

À empresa Sobralnet Serviços e Telecomunicações LTDA, pela disponibilização de recursos informáticos para a instalação dos servidores necessários e realização dos experimentos.

Ao Dr. Ednardo Moreira Rodrigues e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

“O sucesso é treinável e você deve começar a caminhar em direção a ele agora!”

(Joel Jota)

RESUMO

A internet desempenha um papel fundamental na conexão de soluções inovadoras e pessoas, tornando essencial que os Provedores de Serviço de Internet (ISPs) ofereçam um serviço cada vez mais seguro, confiável e escalável. Para atender à essas demandas, faz-se necessário manter alta disponibilidade e qualidade das redes ópticas utilizadas na entrega destas conexões, tornando essencial o monitoramento dessas redes. A presente pesquisa aborda a implementação de uma rede de monitoramento baseada em tecnologia LoRaWAN, com foco em redes ópticas FTTH (Fiber to the Home), que permita a coleta de dados em tempo real para o gerenciamento eficiente dessas redes, visando reduzir o tempo de reparo e melhorar a disponibilidade do sinal para os usuários. O protocolo LoRaWAN é escolhida devido à sua capacidade de oferecer comunicações de longo alcance com baixo consumo de energia, o que é ideal para o monitoramento de redes ópticas extensas. A pesquisa detalha a configuração dos principais componentes da rede LoRaWAN: o nó de rede utilizando o módulo Heltec WiFi LoRa 32 V2, o *gateway* RD43HATGPS e os servidores de rede ChirpStack e de aplicação Zabbix. O Heltec WiFi LoRa 32 V2 foi escolhido por sua robustez e eficiência energética, enquanto o *gateway* RD43HATGPS foi selecionado pela sua alta potência e capacidade de integração com o Raspberry Pi 3. A pesquisa destaca a eficiência do sistema implementado na coleta e gerenciamento de dados das redes FTTH, assim como ressalta a importância do monitoramento contínuo das redes ópticas para garantir a qualidade do serviço e minimizar o tempo de inatividade. O uso da tecnologia LoRaWAN demonstrou ser uma solução eficaz para esse propósito, proporcionando uma comunicação confiável e de baixo custo.

Palavras-chave: LoRaWAN. IoT. FTTH. Sensoriamento. Zabbix

ABSTRACT

The internet plays a crucial role in connecting innovative solutions and people, making it essential for Internet Service Providers (ISPs) to offer increasingly secure, reliable, and scalable services. To meet these demands, it is necessary to maintain high availability and quality of the optical networks used to deliver these connections, making network monitoring essential. This research addresses the implementation of a monitoring network based on LoRaWAN technology, focusing on FTTH (Fiber to the Home) optical networks. The goal is to enable real-time data collection for the efficient management of these networks, aiming to reduce repair time and improve signal availability for users. The LoRaWAN protocol is chosen due to its capability to offer long-range communication with low energy consumption, which is ideal for monitoring extensive optical networks. The research details the configuration of the main components of the LoRaWAN network: the network node using the Heltec WiFi LoRa 32 V2 module, the RD43HATGPS gateway, and the ChirpStack network server and Zabbix application server. The Heltec WiFi LoRa 32 V2 was selected for its robustness and energy efficiency, while the RD43HATGPS gateway was chosen for its high power and ability to integrate with the Raspberry Pi 3. The research highlights the efficiency of the implemented system in collecting and managing data from FTTH networks and emphasizes the importance of continuous monitoring of optical networks to ensure service quality and minimize downtime. The use of LoRaWAN technology has proven to be an effective solution for this purpose, providing reliable and low-cost communication.

Keywords: LoRaWAN. IoT. FTTH. Sensing. Zabbix

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

10G-EPON	10 Gigabit Ethernet Passive Optical Network
ABP	Activation By Personalization
ADC	Analog to Digital Converter
ADR	Adaptative Data Rate
AES	Advanced Encryption Standard
API	Application Programming Interface
AppKey	Application Key
AppSKey	Application Session Key
BLE	Bluetooth Low Energy
CEO	Caixa de Emenda Óptica
CSS	Chirp Spread Spectrum
CTO	Caixa de Terminação Óptica
DAC	Digital to Analog Converter
DevAddr	Device Address
DHCP	Dynamic Host Configuration Protocol
DIO	Distribuidor Interno Óptico
eDRX	extended Discontinuous Reception
EPON	Ethernet Passive Optical Network
EUI	End Unique Identifier
FTTH	Fiber To The Home
GPON	Gigabit Passive Optical Network
GPS	Global Positioning System
HVAC	Heating, Ventilating and Air Conditioning
IEEE	Institute of Electrical and Electronic Engineers
IoT	Internet Of Things
IP	Internet Protocol
ISM	Industrial, Scientific, and Medical
ISO	Imagem de Sistema Operacional
ISP	Internet Service Provider
JSON	JavaScript Object Notation
LoRa	Long-Range

LoRaWAN	Long Range Wide Area Network
LPP	Low Power Payload
LPSAN	Low-Power Short Area Network
LPWAN	Low-Power Wide Area Network
LTE	Long Term Evolution
LTE-M	Long Term Evolution for Machines
M2M	Machine To Machine
MAC	Media Access Control
MQTT	Message Queuing Telemetry Transport
NAP	Network Access Point
NB-IoT	Narrowband Internet of Things
NOC	Network Operations Center
NwkSKey	Network Session Key
ODN	Optical Distribution Network
ODP	Optical Distribution Point
OLT	Optical Line Termination
ONT	Optical Network Terminal
ONU	Optical Network Unit
OTAA	Over The Air Activation
OTDR	Optical Time Domain Reflectometer
PON	Passive Optical Network
PSM	Power Save Mode
RAM	Random Access Memory
SAC	Serviço de Atendimento ao Consumidor
SPI	Serial Peripheral Interface
SSH	Secure Shell
TTN	The Things Network
UDP	User Datagram Protocol
URl	Uniform Resource Locator
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network
XGPON	10 Gigabit Passive Optical Network

SUMÁRIO

1 INTRODUÇÃO

No atual cenário tecnológico, a internet vem desempenhando um papel crucial na conexão de soluções inovadoras e pessoas ao redor do mundo. O provedor de internet, ou Internet Service Provider (ISP), tem um papel fundamental para realizar essas conexões aos usuários, viabilizando o acesso às soluções existentes e ferramentas que são cada vez mais necessárias no cotidiano, como aplicativos bancários, redes sociais, sistemas de automações residenciais, entre outros. Nesse contexto, faz-se necessária a entrega de uma conexão de internet cada vez mais segura e confiável aos usuários (TANG; ZHOU, 2018), fazendo com que a fibra óptica seja o meio de transmissão mais indicado para cumprir com esses requisitos, além de conseguir suprir a crescente demanda por largura de banda (ENRIKO *et al.*, 2023).

Para a entrega dessa conexão de alta qualidade, os ISPs têm cada vez mais adotado a topologia Fiber To The Home (FTTH), que utiliza a fibra óptica e elementos passivos para constituir a rede, desde o servidor do ISP, até o interior das residências e empresas. De acordo com Hammadi (2022), essas redes FTTH são um dos sistemas de comunicação mais econômicos e fáceis de implementar na atualidade. Uma das características desse tipo de rede, é a utilização de elementos passivos, com exceção do servidor, denominado Optical Line Termination (OLT), e do equipamento no usuário, conhecido como Optical Network Unit (ONU). Essa topologia oferece benefícios importantes, como a imunidade à interferências eletromagnéticas nos cabos e elementos passivos, mas também apresenta desafios significativos para os provedores, particularmente no quesito de monitoramento da rede (HAMMADI, 2022). Tang e Zhou (2018) observaram que os métodos tradicionais de monitoramento, que usam Optical Time Domain Reflectometer (OTDR), apresentam limitações na precisão das medições em redes ópticas passivas, denominadas Passive Optical Network (PON), e são inadequados em certos cenários. Enriko *et al.* (2023) afirmam que o monitoramento dessas redes se tornou essencial para a garantia de disponibilidade e desempenho exigidos.

Nesse contexto, a presente pesquisa propõe um sistema de monitoramento para redes FTTH, utilizando para isso transmissores posicionados dentro das caixas de atendimento passivas das mesmas. Esses transmissores utilizam a tecnologia sem fio LoRa e o protocolo LoRaWAN para enviar os dados de sinal destas caixas para um servidor de rede ChirpStack. Este, por sua vez, disponibiliza os dados para um servidor de aplicação Zabbix, no qual é realizado o armazenamento das informações de sinal da rede. No Zabbix também é possível analisar o histórico de sinal da rede, gerar alarmes no caso de alteração deste sinal, envio de alertas para

aplicativos de mensagem como WhatsApp e Telegram, entre outras funcionalidades.

Com isso, o setor do ISP responsável pelo monitoramento da rede, conhecido como Network Operations Center (NOC), consegue se manter informado em tempo real sobre eventos da rede, como atenuações e rompimentos de cabos. Isso permite uma redução no tempo de reparo da rede, possibilitando inclusive o acionamento automático das equipes pelo Zabbix no caso de alguma alteração do sinal monitorado. Além disso, o armazenamento de histórico de sinal proporciona um valor de referência que pode ser comparado com medições futuras após reparos realizados na rede, possibilitando a validação dos procedimentos realizados pelas equipes.

A pesquisa de Nascimento (2024) propõe um sistema de monitoramento semelhante, com ênfase maior em um dispositivo de medição de sinal remoto. A presente pesquisa visa desenvolver um sistema de comunicação robusto, seguro, flexível e escalável, que poderá ser utilizado por dispositivos como o proposto pelo mesmo.

1.1 Objetivo Geral

Implementar uma rede LoRa utilizando o protocolo LoRaWAN para integrar um sistema de monitoramento para redes ópticas FTTH, com a finalidade de armazenar o histórico de sinal da rede, detectar falhas em tempo real, gerar relatórios, enviar alarmes informativos ao NOC e contribuir com a redução no tempo de indisponibilidade da rede em caso de falhas.

1.2 Objetivos Específicos

a) Implementar uma rede LoRa em laboratório utilizando um Gateway da Radioengie e um ESP32 LoRa WiFi v2 como nó da rede;
b) Adotar e configurar um servidor de rede LoRa para comunicação com o Gateway e recepção dos dados coletados pelos sensores;
c) Adotar e configurar um servidor de aplicação LoRa, para armazenamento e tratamento dos dados;
d) Implementar no servidor de aplicação um dashboard para gerenciamento dos indicadores da rede, assim como alertas automáticos para o aplicativo Telegram no caso de eventos na rede FTTH;
e) Realizar testes gerando sinais de entrada no ESP32 para simulações de eventos em uma rede FTTH.

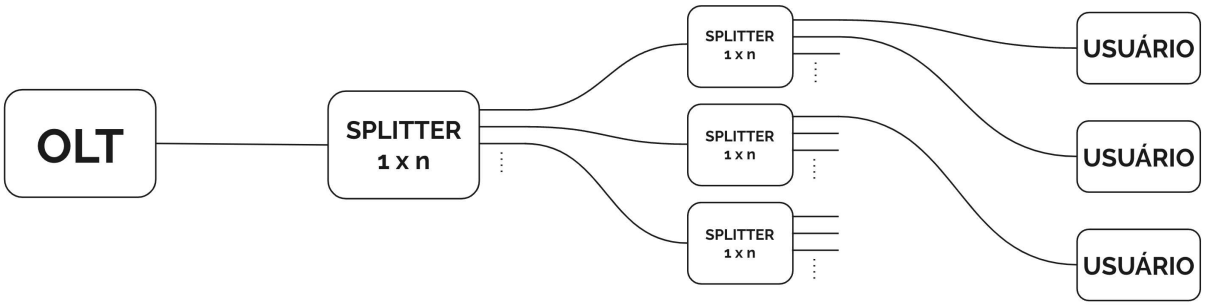
2 REDES FTTH (FIBER TO THE HOME)

As redes FTTH representam uma revolução nas telecomunicações modernas, oferecendo conectividade de fibra óptica diretamente até as residências dos usuários finais de forma confiável (SUGUMARAN *et al.*, 2021). Essa tecnologia tem sido amplamente adotada devido à sua capacidade de suportar altas velocidades de transmissão de dados e uma variedade de serviços avançados de banda larga (FILGUEIRAS; PESSOA, 2015), (ABDELLAOUI *et al.*, 2021). Esse tipo de rede utiliza fibra óptica para transmitir dados em forma de luz pulsada, garantindo maior largura de banda e menor degradação do sinal em comparação com tecnologias tradicionais baseadas em cobre. Essa abordagem não apenas aumenta a capacidade da rede, mas também permite alcançar distâncias maiores sem perda significativa de qualidade de sinal (ABDELLAOUI *et al.*, 2021).

A implantação de redes FTTH não apenas melhora a experiência dos usuários finais em termos de velocidade e confiabilidade da conexão, mas também impulsiona o desenvolvimento econômico e social das comunidades. A disponibilidade de conexões de alta velocidade facilita o acesso a serviços educacionais, oportunidades de emprego remoto e suporte a iniciativas governamentais digitais (RIDHO *et al.*, 2020).

Com relação à topologia, as redes FTTH são projetadas para maximizar a eficiência e a confiabilidade da conexão de fibra óptica até o usuário final, utilizando uma comunicação ponto a multiponto, onde um único ponto central no provedor de serviços (OLT) conecta-se a várias residências dos usuários finais através de divisores chamados *splitters* (Figura 1). Esta arquitetura permite uma distribuição eficiente de largura de banda e serviços personalizados para cada cliente (RIDHO *et al.*, 2020), (SALAMI; ADEWOLE, 2022).

Figura 1 – Esquema de atendimento da rede FTTH

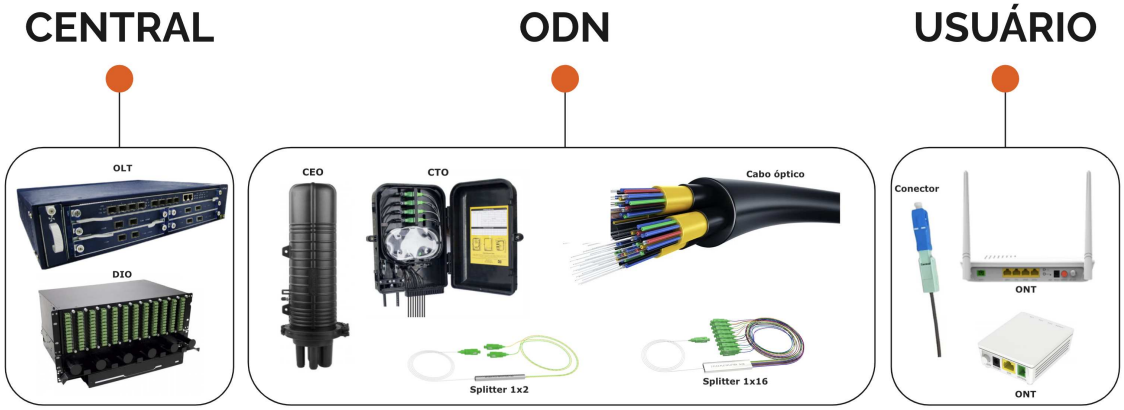


Fonte: Autoria própria (2024).

2.1 Componentes de uma rede FTTH

Como principais componentes de uma rede FTTH, podemos citar: Opical Line Terminal (OLT), Optical Distribution Network (ODN), Optical Distribution Point (ODP) e Optical Network Terminal (ONT) (RIDHO *et al.*, 2020), (SUGUMARAN *et al.*, 2021). A Figura 2 mostra um esquema que divide a rede FTTH em três principais partes com seus respectivos componentes.

Figura 2 – Componentes da rede FTTH



Fonte: Autoria própria (2024).

2.1.1 OLT (Optical Line Terminal)

O terminal de linha óptica é um dispositivo central dessa rede, ficando posicionada no ambiente interno da estrutura do ISP, juntamente com o Distribuidor Interno Óptico (DIO), que é utilizado para interconexão entre os cabos ópticos e a OLT. A OLT é responsável por converter a comunicação elétrica em comunicação óptica para os usuários finais, assim como gerenciar o envio e recebimento de dados das ONUs.

2.1.2 ODN (Optical Distribution Network)

A rede de distribuição óptica é a parte passiva da rede FTTH, responsável por interligar a OLT à porção final da rede, próximo à residência dos assinantes. É na ODN onde estão contidos os cabos ópticos, Caixa de Emenda Óptica (CEO), Caixa de Terminação Óptica (CTO)) e *splitters*, que são os divisores de sinal passivos utilizados para esse tipo de rede de comunicação Ponto-a-Multi-Ponto.

2.1.3 ODP (Optical Distribution Point)

O ponto de distribuição óptica, também chamado de Network Access Point (NAP) ou CTO, é uma caixa que fica na porção final da rede FTTH. É a partir desta, que os usuários são conectados na rede, através de um cabo óptico entre a residência do mesmo e esse equipamento

2.1.4 ONT (Optical Network Unit)

O terminal de rede óptica, também conhecido com ONU, é um dispositivo que fica posicionado dentro da residência do usuário da rede FTTH. A ONT é responsável por converter o sinal óptico proveniente da rede, para sinal elétrico que se comunicará com a rede do usuário através de interfaces como *Wireless Fidelity (Wi-Fi)* ou *Ethernet*.

2.2 Tecnologias em redes FTTH

As redes FTTH possibilitam a aplicação de diversas tecnologias que sejam compatíveis com redes ópticas Ponto-Multi-Ponto, sendo as principais: Ethernet Passive Optical Network (EPON), Gigabit Passive Optical Network (GPON), 10 Gigabit Ethernet Passive Optical Network (10G-EPON) e 10 Gigabit Passive Optical Network (XGPON) (SUGUMARAN *et al.*, 2021)

2.2.1 EPON (Ethernet Passive Optical Network)

Normatizado pelo padrão Institute of Electrical and Electronic Engineers (IEEE) 802.3ah, essa tecnologia utiliza protocolos Ethernet para a transmissão de dados, sendo uma solução eficiente para cenários com altas demandas de largura de banda, possuindo comunicação síncrona com 1,25Gbps de taxa de dados no sentido *upstream* e *downstream*.

2.2.2 GPON (Gigabit Passive Optical Network)

É uma tecnologia estabelecida pelo padrão ITU-T G.984 que permite a transferência de dados com taxas assíncronas, possibilitando uma largura de banda de 2,5Gbps de *downstream* e 1,25Gbps de *upstream*.

2.2.3 10G-EPON (10 Gigabit Ethernet Passive Optical Network)

É uma tecnologia padronizada pela norma IEEE 802.3av que é uma evolução do EPON, oferecendo taxas de até 10Gbps no sentido *downstream* e até 10 Gbps no sentido *upstream*, em seu modo síncrono.

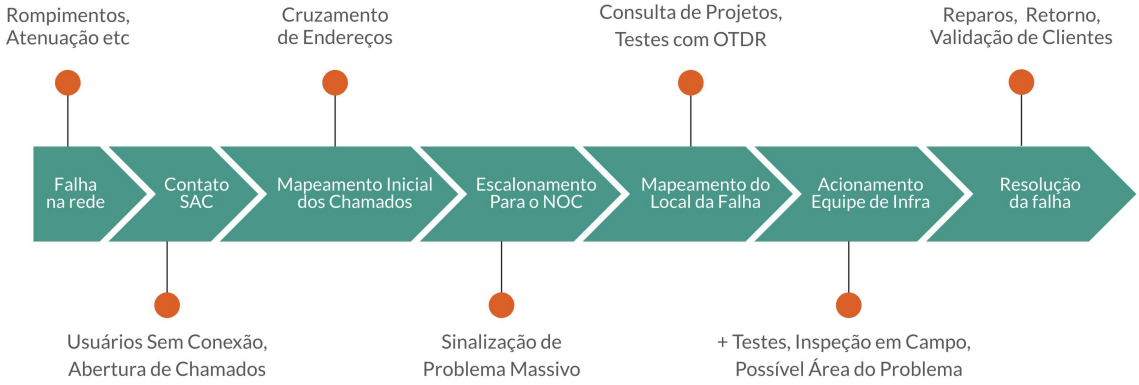
2.2.4 XGPON (10 Gigabit Passive Optical Network)

Evolução do padrão GPON, essa tecnologia é documentada pela norma ITU-T G.987. O XGPON traz uma comunicação com taxas assíncronas, possibilitando 10Gbps de *downstream* e 2,5Gbps de *upstream*.

2.3 Desafio do monitoramento de redes FTTH

Apesar das vantagens existentes, as redes FTTH trazem um grande desafio com relação ao seu monitoramento devido à sua natureza passiva. Sugumaran *et al.* (2021) afirmam que a impossibilidade de levar energia elétrica através dos cabos passivos da rede, inviabiliza a ativação de elementos ativos na mesma, dificultando a instalação de equipamentos para monitoramento de sinal. Essa característica dificulta a implementação de sistemas ativos que possam detectar eventos na rede, como rompimento de cabos ou atenuações dos mesmos. Para mitigar esses problemas, técnicas como uso de equipamentos do tipo OTDR e visita de campo para verificação visual da rede por equipes especializadas são comumente utilizadas, porém demandam um tempo considerável até a identificação da falha na rede. Essa deficiência no monitoramento, faz com que o cenário atual de gestão e resolução de eventos na rede FTTH do provedor de internet tenha basicamente sete passos, que podem ser visualizados na Figura 3.

Figura 3 – Cenário atual no gerenciamento de eventos da rede FTTH em um ISP



Fonte: Autoria própria (2024).

Ao analisar o cenário presente na Figura 3, é possível notar que, ao ocorrer alguma falha na rede, o usuário que é afetado pelo evento faz contato com o ISP para informá-la. O setor de atendimento então faz aberturas de chamados e cruzamento de dados de endereços no caso de vários contatos de clientes da mesma região. Logo em seguida, esses dados são repassados para o setor do NOC, que faz uma triagem mais precisa na tentativa de localizar a falha, podendo incluir processos de testes e consulta de projetos. Após a identificação do local aproximado, a equipe de campo especializada em resolver o caso é acionada, segue até o local informado, faz as tratativas e restabelece o sinal da rede. Esses processos acabam ocasionando uma demora na resolução do problema, visto que se faz necessário aguardar o retorno dos usuários, assim como empregar um tempo realizando filtros, testes e processos de triagem a fim de se localizar o ponto da falha.

A fim de se otimizar esse tempo de reparo, as redes Long Range Wide Area Network (LoRaWAN) podem ser empregadas para implantação de um sistema de monitoramento em tempo real, se utilizando da comunicação sem fio LoRa para troca de informações entre o elemento que captará os eventos da rede e o servidor central de monitoramento da mesma.

3 REDES DE SENSORES SEM FIO

As redes de sensores sem fio, ou Wireless Sensor Network (WSN), são compostas por pequenos dispositivos com capacidades de sensoriamento, processamento e comunicação sem fio. Estes dispositivos, chamados de nós sensores, são utilizados para monitorar ambientes e coletar dados em tempo real. Estas redes utilizam diversas tecnologias de comunicação para transmitir dados de sensores de forma eficiente e confiável, oferecendo opções para uso em setores como agricultura, saúde, monitoramento ambiental e segurança. A escolha da tecnologia ideal para os projetos, depende da análise de parâmetros como alcance, consumo de energia, taxa de dados e custo (PIZA *et al.*, 2013). As WSNs são classificadas em duas categorias principais: Low-Power Short Area Network (LPSAN) e Low-Power Wide Area Network (LPWAN) (MEKKI *et al.*, 2019), (ZANAJ *et al.*, 2021).

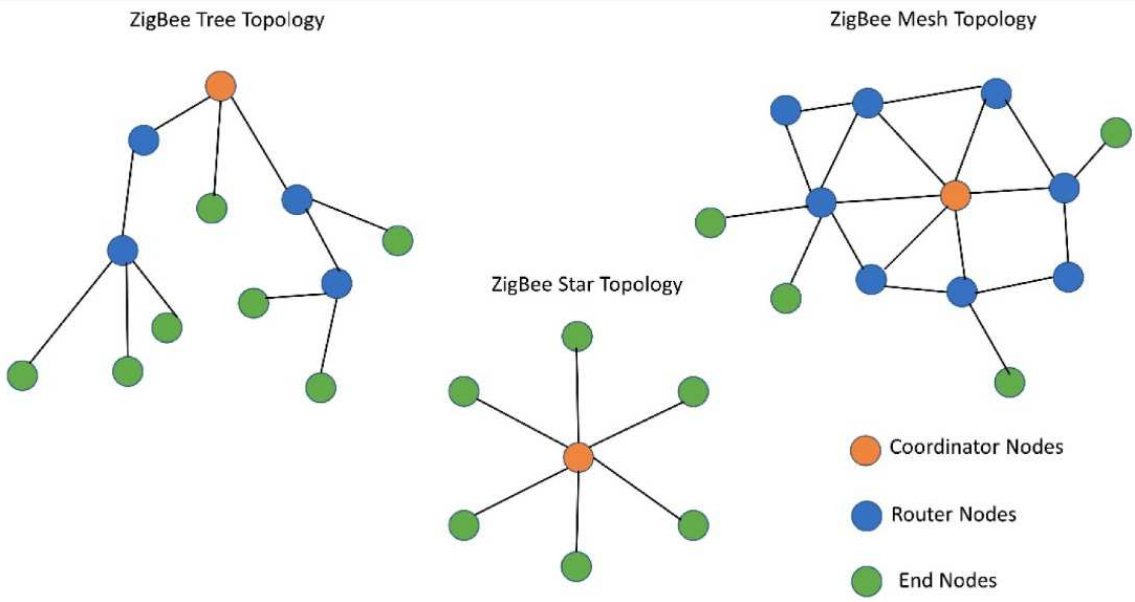
3.1 LPSAN (Low-Power Short Area Network)

A LPSAN trata-se de um tipo de rede utilizada para fornecer comunicações eficientes empregando o mínimo de consumo de energia em distâncias pequenas. Essas redes são ideais para dispositivos pequenos, alimentados por bateria, que necessitam de comunicação constante em áreas limitadas. LPSANs são altamente vantajosas em termos de eficiência energética e durabilidade da bateria, sendo especialmente adequadas para aplicações como monitoramento ambiental, dispositivos vestíveis e automação residencial. Entre as principais tecnologias LPSAN que utilizam a comunicação sem fio, destacam-se o ZigBee e o Bluetooth Low Energy (BLE) (ZANAJ *et al.*, 2021), (HAQUE *et al.*, 2022).

3.1.1 Zigbee

O ZigBee é um padrão de comunicação sem fio baseado na especificação do IEEE 802.15.4, projetado para fornecer uma solução robusta e de baixo custo, ideal para aplicações de monitoramento e sensoriamento. Essa tecnologia é particularmente eficiente em redes com topologias variadas, como estrela, árvore e mesh (Figura 4), o que facilita sua aplicação em diferentes cenários, tanto internos quanto externos, maximizando a eficiência energética e a confiabilidade da comunicação (NOREEN *et al.*, 2017), (HAQUE *et al.*, 2022).

Figura 4 – Topologias do padrão Zigbee



Fonte: Haque *et al.* (2022).

Uma das principais características do ZigBee é sua capacidade de operar com baixo consumo de energia, o que o torna ideal para dispositivos que necessitam de longos períodos de operação com baterias. Além disso, a tecnologia oferece uma taxa de dados de até 250 kbit/s, o que é suficiente para a maioria das aplicações de sensoriamento que transmitem dados de forma intermitente. A segurança também é um ponto forte do ZigBee, que utiliza criptografia Advanced Encryption Standard (AES) de 128 bits, proporcionando uma camada adicional de proteção para a rede (LIMA; NUNES, 2015), (ALVES; FILHO, 2020).

No contexto da automação, o ZigBee é amplamente utilizado para controlar luzes, fechaduras de portas, detectores de fumaça, ventiladores e outros dispositivos domésticos inteligentes. Em ambientes industriais, o ZigBee é usado para automação de controle de iluminação, sistemas de segurança e controle de acesso, assim como controle de Heating, Ventilating and Air Conditioning (HVAC). A tecnologia também é aplicada no monitoramento de saúde, onde sensores podem coletar e transmitir dados vitais de pacientes, como frequência cardíaca e níveis de glicose no sangue. Além disso, o ZigBee é utilizado em tecnologias verdes, como fazendas solares e redes de carregamento de veículos elétricos, assim como em redes de sensores para controle de iluminação pública e semáforos em cidades inteligentes (LIMA; NUNES, 2015), (SHI *et al.*, 2017), (ALVES; FILHO, 2020).

3.1.2 BLE (Bluetooth Low Energy)

Bluetooth Low Energy (BLE) é uma tecnologia de comunicação sem fio projetada para aplicações de curto alcance e baixo consumo de energia, sendo ideal para dispositivos móveis e *wearables*. O BLE se destaca por consumir significativamente menos energia em comparação com o Bluetooth clássico, o que é crucial para dispositivos alimentados por bateria, permitindo que funcionem por longos períodos sem a necessidade de recargas frequentes. Isso é alcançado graças ao ciclo de trabalho muito baixo do BLE, onde o rádio permanece desligado na maior parte do tempo, sendo ativado apenas quando necessário (KARVONEN *et al.*, 2020).

O BLE é considerado uma tecnologia flexível, permitindo o seu uso em diversas aplicações, desde a transmissão de dados simples de sensores até cenários mais complexos de troca de informações, suportando taxas de dados de até 1 Mbps (KARVONEN *et al.*, 2020). O alcance do BLE geralmente chega até 50 metros, mas pode ser ajustado mediante a modificação da potência de transmissão conforme as necessidades da aplicação. Outra característica importante do BLE é o tempo rápido de estabelecimento de conexões, o que é benéfico para aplicações que requerem troca intermitente de dados ou comunicação frequente em curtos intervalos (WOOLLEY, 2022).

BLE pode ser aplicado em rastreadores fitness, utilizados para monitoramento de atividades físicas e métricas de saúde, bem como em dispositivos médicos, permitindo o desenvolvimento de aparelhos portáteis para monitoramento de sinais vitais e envio de relatórios para outros dispositivos, como smartphones. Em automações residenciais, o BLE também pode ser utilizado nos sistemas de iluminação, termostatos e sistemas de segurança (KARVONEN *et al.*, 2020), (WOOLLEY, 2022).

3.2 LPWAN (Low-Power Wide Area Network)

A LPWAN é uma categoria de tecnologias de comunicação sem fio projetadas para fornecer conectividade de longo alcance com baixo consumo de energia, sendo ideal para conectar um grande número de dispositivos IoT distribuídos em áreas amplas (FARRELL, 2018), (ALIMI *et al.*, 2020). De acordo com Mekki *et al.* (2019), as redes LPWAN podem alcançar de 10 km até 40 km em zonas rurais e de 1 km à 5 km em zonas urbanas[1] com um baixo custo de implantação, tornando essas redes cada vez mais populares em comunidades industriais e de pesquisa. Ismail *et*

[1] A diferença de alcance se dá principalmente pela presença de barreiras físicas dentro das cidades, como edifícios e demais construções elevadas.

al. (2018) destacam que essas redes são cruciais para aplicações como monitoramento ambiental, agricultura inteligente e cidades inteligentes, assim como em aplicações que demandam soluções eficientes e escaláveis de monitoramento remoto, como automação industrial, rastreamento de ativos e monitoramento de redes.

De acordo com Mekki *et al.* (2019), Gu *et al.* (2020) e Zanaj *et al.* (2021), ganham destaques como tecnologias do tipo LPWAN o Narrowband Internet of Things (NB-IoT), o Long Term Evolution for Machines (LTE-M), as redes Sigfox e a tecnologia Long-Range (LoRa), cada uma com suas particularidades e aplicações específicas.

3.2.1 NB-IoT (Narrowband Internet of Things)

O NB-IoT (Narrowband IoT) é uma tecnologia LPWAN desenvolvida para redes celulares, operando em bandas 4G e 5G, com o objetivo de oferecer cobertura de longo alcance e baixo consumo de energia. Suas características incluem baixa taxa de dados, adequada para a transmissão de pequenos pacotes de dados, o que o torna ideal para aplicações que não exigem altas velocidades de transferência (ISMAIL *et al.*, 2018). O alcance do NB-IoT pode chegar a vários quilômetros, dependendo da infraestrutura da rede, sendo especialmente eficiente em termos de consumo de energia, permitindo uma longa duração da bateria dos dispositivos (MATZ *et al.*, 2020), (ALIMI *et al.*, 2020), (HOSSAIN; MARKENDAHL, 2021).

Uma das grandes vantagens do NB-IoT é sua excelente capacidade de penetração em ambientes interiores e subterrâneos, aumentando sua versatilidade em diferentes cenários de implantação. As principais aplicações do NB-IoT incluem medição remota, rastreamento de ativos, sistemas de monitoramento ambiental e soluções de Internet Of Things (IoT) industrial (ISMAIL *et al.*, 2018). Essa tecnologia é amplamente utilizada para monitorar e gerenciar operações em setores como agricultura, monitoramento de qualidade do ar e máquinas de venda automáticas inteligentes. Além disso, o NB-IoT oferece alta segurança, utilizando os mesmos recursos de criptografia e autenticação do Long Term Evolution (LTE), o que garante a proteção dos dados transmitidos. Com sua capacidade de suportar grandes volumes de dispositivos conectados, podendo chegar até 50 mil por célula (ISMAIL *et al.*, 2018), e a possibilidade de operar de forma eficiente em infraestruturas já existentes, o NB-IoT está se tornando uma escolha popular para soluções IoT em larga escala (ROUTRAY; MOHANTY, 2024).

3.2.2 LTE-M (Long Term Evolution for Machines)

O LTE-M é uma tecnologia de comunicação sem fio proveniente da tecnologia LTE (4G), projetada especificamente para aplicações de Internet das Coisas (IoT) e comunicação Machine To Machine (M2M), possibilitando dispositivos finais de menor custo quando comparado com outros ambientes LTE (BORKAR, 2020). Essa tecnologia se destaca por ser uma solução de área ampla e de baixa potência, oferecendo vantagens significativas quando comparada com as redes de celulares 2G e 3G (BORKAR, 2020), (HOSSAIN; MARKENDAHL, 2021).

Uma das principais características do LTE-M é sua maior largura de banda, que permite suportar taxas de dados de até 1 Mbit/s para *upload* e 375 kbit/s para *download*, em modo *half-duplex*. Isso o torna adequado para aplicações que requerem a transmissão de maiores volumes de dados (HOSSAIN; MARKENDAHL, 2021). Além disso, o LTE-M suporta modos de economia de energia, como Power Save Mode (PSM) e extended Discontinuous Reception (eDRX), que prolongam a vida útil da bateria dos dispositivos IoT por até 10 anos (BORKAR, 2020), (MILAROKOSTAS *et al.*, 2022).

Outra importante característica do LTE-M é o seu suporte total à mobilidade, o que permite a transferência de dados entre torres de celular sem interrupções, ideal para aplicações móveis como rastreamento de veículos, gestão de frotas e telemetria (ISMAIL *et al.*, 2018). A tecnologia também oferece excelente penetração de sinal em ambientes internos e subterrâneos, graças ao seu orçamento de link de até 155.7 dB, garantindo uma conectividade robusta mesmo em condições desafiadoras (MILAROKOSTAS *et al.*, 2022). De acordo com Borkar (2020), os principais candidatos a aplicações que exigem tais atributos são: transporte inteligente, serviços de saúde críticos e urgentes, *wearables* que monitoram medições vitais e aplicações industriais.

3.2.3 SigFox

O SigFox é uma rede LPWAN criada em 2009 e projetada especificamente para comunicação de baixo custo e baixa largura de banda. Essa rede se utiliza de frequências de rádio não licenciadas, sendo a primeira rede LPWAN proposta para uso em aplicações IoT (CENTENARO *et al.*, 2016), (ALIMI *et al.*, 2020). De acordo com Centenaro *et al.* (2016) e Aernouts *et al.* (2018), esse tipo de rede viabiliza a comunicação de diversos dispositivos com até 50Km de distância e baixo uso de energia, além de demandar um *hardware* de baixo custo e de possuir fácil implementação. Por outro lado, o SigFox tem uma limitação significativa em sua

taxa de transferência, possibilitando uma transmissão de no máximo 100bps (AERNOUTS *et al.*, 2018), (MEKKI *et al.*, 2019), (HOSSAIN; MARKENDAHL, 2021).

O SigFox também é extremamente eficiente em energia, o que prolonga significativamente a vida útil das baterias dos dispositivos conectados (SIGFOX, 2024). Além disso, utiliza uma infraestrutura proprietária global, garantindo uma operação confiável e segura através de uma rede dedicada, porém oferecendo menos flexibilidade em comparação com soluções baseadas em padrões abertos (ISMAIL *et al.*, 2018), (HOSSAIN; MARKENDAHL, 2021).

As aplicações típicas do SigFox incluem monitoramento de infraestruturas como pontes, estradas e edifícios para detecção de anomalias estruturais. Também é frequentemente utilizada para leitura remota de medidores de água, gás e eletricidade. Sistemas de alarme que utilizam SigFox podem enviar alertas em tempo real sobre atividades suspeitas ou falhas de segurança, beneficiando-se da comunicação de longo alcance e baixo consumo de energia para garantir a operação contínua e confiável dos dispositivos de segurança (MEKKI *et al.*, 2019), (HOSSAIN; MARKENDAHL, 2021).

3.2.4 LoRa

LoRa (Long Range) é uma tecnologia desenvolvida pela empresa Semtech Corporation que opera na faixa de frequências Industrial, Scientific, and Medical (ISM), permitindo comunicação sem fio de longo alcance com baixo consumo de energia (MEKKI *et al.*, 2019), (ALIMI *et al.*, 2020). Utilizando a modulação Chirp Spread Spectrum (CSS), LoRa oferece alta sensibilidade do receptor e resistência a interferências, o que possibilita uma comunicação confiável mesmo em ambientes ruidosos (GEORGIOU; RAZA, 2017). A segurança também é um ponto forte da tecnologia LoRa, que permite aplicações que utilizam criptografia de ponta a ponta para garantir a proteção dos dados transmitidos, essencial em aplicações sensíveis como monitoramento de saúde e segurança pública. Essas características técnicas são essenciais para alcançar longas distâncias de transmissão, especialmente em áreas rurais, onde pode atingir até 20 km, e em áreas urbanas, com alcance de 2 à 5 km (SILVA *et al.*, 2017), (NOREEN *et al.*, 2017).

De acordo com Mekki *et al.* (2019), as aplicações de LoRa em IoT são diversas e abrangem desde a agricultura inteligente, monitorando parâmetros como umidade e temperatura do solo, até monitoramento ambiental e cidades inteligentes, sendo empregada para gerenciamento de sistemas de iluminação pública, coleta de lixo e monitoramento de qualidade do ar.

Em ambientes industriais, a tecnologia LoRa também é usada para monitoramento de máquinas e processos, detectando anomalias para prevenir falhas, sendo uma comunicação confiável e possuindo baixo consumo de energia (HAXHIBEQIRI *et al.*, 2018), (ALMUHAYA *et al.*, 2022).

Com base nessa tecnologia, a LoRa Alliance, uma associação aberta e sem fins lucrativos, promove o padrão LoRaWAN, um protocolo que funciona sobre a tecnologia LoRa, garantindo interoperabilidade entre dispositivos e redes que se baseiam no mesmo. Com mais de 400 membros globais, a aliança facilita a padronização e o crescimento das redes LoRaWAN, impulsionando a adoção da tecnologia em escala global (HAXHIBEQIRI *et al.*, 2018), (ALMUHAYA *et al.*, 2022).

4 PROTOCOLO LORAWAN

O LoRaWAN (Long Range Wide Area Network) é um padrão LPWAN aberto que permite comunicação de longo alcance com baixo consumo de energia, ideal para conectar dispositivos IoT em áreas extensas (SILVA *et al.*, 2017), (ALIPIO; BURES, 2023). Mekki *et al.* (2019) afirma em seu estudo que a comunicação em redes LoRaWAN pode alcançar distâncias de até 15 à 20 km em áreas rurais, enquanto em áreas urbanas, o alcance é geralmente entre 2 e 5 km, tornando o protocolo extremamente versátil para uma ampla gama de aplicações geográficas. Suportando taxas de dados que variam de 0,3 kbps a 50 kbps, é adequado para a transmissão de pequenas quantidades de dados típicas em aplicações de IoT que não necessitam de altas larguras de banda (SILVA *et al.*, 2017), (HOSSAIN; MARKENDAHL, 2021), (ALIPIO; BURES, 2023).

Esse padrão é projetado para operar com baterias por longos períodos sem necessidade de substituição frequente, podendo chegar a alguns anos de autonomia. Essa eficiência energética é crucial para dispositivos IoT implantados em locais remotos ou de difícil acesso (ERTÜRK *et al.*, 2019). O protocolo também utiliza criptografia de ponta a ponta para garantir a segurança dos dados transmitidos, essencial em aplicações sensíveis como monitoramento de saúde e segurança pública (NOREEN *et al.*, 2017), (GU *et al.*, 2020), (HOSSAIN; MARKENDAHL, 2021).

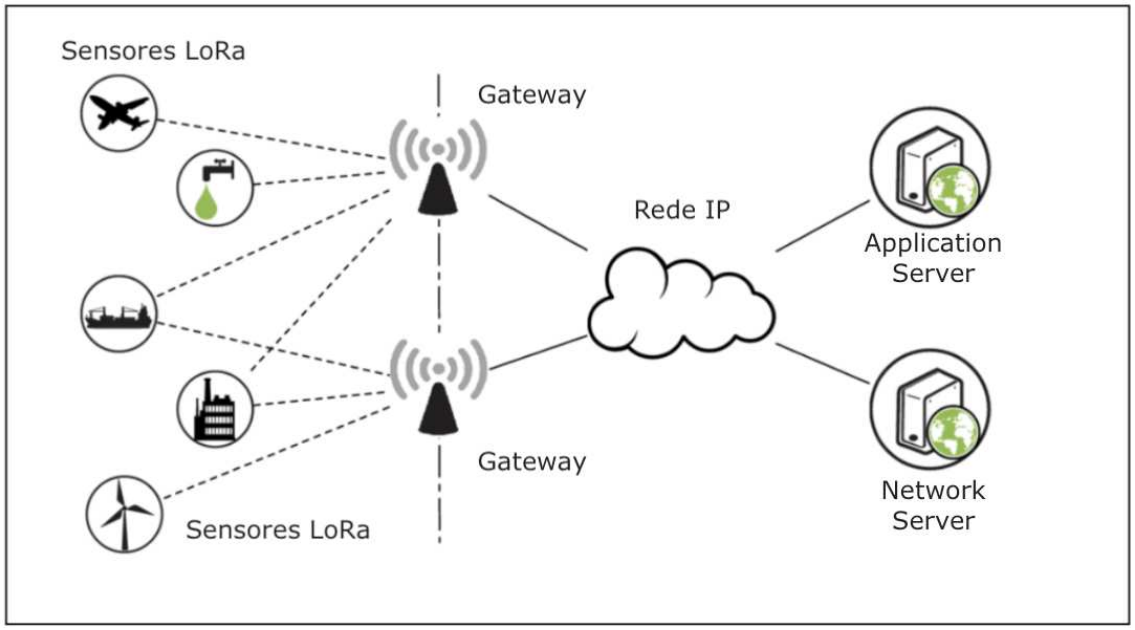
Por conta do baixo consumo de energia e da grande área de cobertura, as redes LoRaWAN são utilizadas para aplicações como: agricultura inteligente, onde sensores IoT podem monitorar condições ambientais e do solo; cidades inteligentes, para gerenciamento de recursos urbanos e infraestrutura; monitoramento ambiental, como qualidade do ar e detecção de vazamentos; controle industrial, para monitoramento de máquinas e processos (SILVA *et al.*, 2017), (HAXHIBEQIRI *et al.*, 2018).

4.1 Arquitetura de Rede LoRaWAN

A arquitetura de rede LoRaWAN (Figura 5) é composta por diversos componentes essenciais que trabalham juntos para possibilitar a comunicação de longo alcance com baixo consumo de energia, típica de redes LPWAN. A topologia de rede LoRaWAN é do tipo estrela, onde os nós finais (*end devices*) se comunicam diretamente com um ou mais *Gateways*, que por sua vez se conectam ao servidor de rede centralizado (SILVA *et al.*, 2017), (MILAROKOSTAS *et al.*, 2022). Esta configuração simplifica a infraestrutura e melhora a eficiência da comunicação,

permitindo que os nós finais mantenham um baixo consumo de energia, pois não precisam rotear dados para outros dispositivos (GU *et al.*, 2020).

Figura 5 – Arquitetura do protocolo LoRaWAN



Fonte: Adaptado de Alghamdi *et al.* (2022).

Os nós da rede, ou dispositivos finais, são sensores ou atuadores equipados com módulos de comunicação LoRaWAN. Esses dispositivos enviam e recebem pequenas quantidades de dados de forma intermitente, utilizando a modulação CSS para alcançar grandes distâncias com baixo consumo de energia, sendo projetados para operar com baterias por longos períodos sem necessidade de substituição (ERTÜRK *et al.*, 2019), (ALGHAMDI *et al.*, 2022).

Os *Gateways* por sua vez, atuam como pontes entre os dispositivos finais e o servidor de rede. Eles recebem as transmissões dos nós finais e as encaminham para o servidor de rede via conexão de alta velocidade, como *Ethernet*, 3G/4G ou *Wi-Fi*. Os *Gateways* são responsáveis por gerenciar a comunicação entre múltiplos dispositivos finais e podem suportar milhares de conexões simultâneas, garantindo escalabilidade da rede (ERTÜRK *et al.*, 2019), (ALMUHAYA *et al.*, 2022), (ALGHAMDI *et al.*, 2022).

O servidor de rede, ou *Network Server*, é a peça central da arquitetura LoRaWAN. Ele é responsável por gerenciar a rede, autenticando dispositivos, encaminhando mensagens recebidas de diferentes gateways, aplicando políticas de segurança e gerenciando as comunicações ascendentes (*uplink*) e descendentes (*downlink*). O servidor de rede também assegura a qualidade do serviço e a eficiência da utilização do espectro de frequência (AUGUSTIN *et al.*, 2016), (ALMUHAYA *et al.*, 2022), (ALGHAMDI *et al.*, 2022).

O servidor de aplicação, ou *Application Server*, processa e armazena os dados recebidos dos dispositivos finais. Ele permite que os dados sejam utilizados por diferentes aplicações IoT, como plataformas de monitoramento ambiental, sistemas de gerenciamento de cidades inteligentes ou soluções de agricultura de precisão. O servidor de aplicação pode incluir Application Programming Interface (API) para facilitar a integração com outros sistemas, permitindo a criação de dashboards e análises em tempo real para gerenciamento dos cenários (SILVA *et al.*, 2017), (ALGHAMDI *et al.*, 2022).

A arquitetura de rede LoRaWAN, com sua topologia estrela, componentes de baixo consumo energético e robustos servidores de rede e aplicação, oferece uma solução eficaz para a conectividade de dispositivos IoT em larga escala, sendo a escolha ideal para aplicações que vão desde monitoramento ambiental e agricultura inteligente até gerenciamento de infraestrutura urbana e controle industrial (SILVA *et al.*, 2017), (SUNDARAM *et al.*, 2019).

4.2 Modos de Ativação

Os dispositivos LoRaWAN podem se conectar à rede de duas maneiras principais: Activation By Personalization (ABP) e Over The Air Activation (OTAA). Cada método de ativação possui características, vantagens e desvantagens que afetam a forma como os dispositivos se autenticarão e estabelecerão uma comunicação segura com a rede (HAXHIBEQIRI *et al.*, 2018), (ROCHA *et al.*, 2023).

4.2.1 ABP (Activation By Personalization)

No método ABP, as chaves de sessão e o endereço do dispositivo são programados diretamente no dispositivo final antes da implantação. Este método permite uma conexão imediata do dispositivo à rede sem a necessidade de uma etapa inicial de autenticação. Os dispositivos LoRaWAN são configurados com um Device Address (DevAddr) fixo e com chaves de sessão que são codificadas no dispositivo final. Essas chaves de sessão incluem a Network Session Key (NwkSKey) e a Application Session Key (AppSKey) (HAXHIBEQIRI *et al.*, 2018), (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

A chave NwkSKey (Network Session Key) é utilizada para a autenticação de mensagens e para garantir a integridade dos dados na comunicação entre o dispositivo final e o servidor de rede. Ela é responsável por verificar se a mensagem recebida não foi alterada durante a

transmissão e se realmente provém de um dispositivo autorizado (HAXHIBEQIRI *et al.*, 2018), (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

A chave AppSKey (Application Session Key) é utilizada para criptografar e descriptografar os dados da carga útil da aplicação (payload). Esta chave garante a confidencialidade dos dados transmitidos entre o dispositivo final e o servidor de aplicação, protegendo a comunicação contra interceptações e acessos não autorizados (HAXHIBEQIRI *et al.*, 2018), (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

A ativação ABP (Activation By Personalization) no LoRaWAN é uma escolha prática para certas aplicações onde a simplicidade de configuração e a necessidade de evitar a complexidade do processo de junção inicial são desejáveis (HAXHIBEQIRI *et al.*, 2018), (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

4.2.2 OTAA (Over-The-Air Activation)

No método de ativação OTAA, os dispositivos finais estabelecem uma sessão segura com a rede LoRaWAN, durante a qual um endereço DevAddr dinâmico é atribuído e chaves de sessão são derivadas a partir de chaves raízes. Esse processo possibilita a renegociação periódica de chaves de sessão e contadores de frames, o que aumenta a segurança dos dados transmitidos e prolonga a vida útil dos dispositivos. Além disso, o OTAA facilita o ajuste dinâmico dos parâmetros de rede pelos dispositivos finais, melhorando significativamente a eficiência da comunicação e simplificando a configuração inicial (HAXHIBEQIRI *et al.*, 2018), (MILAROKOSTAS *et al.*, 2022).

A Application Key (AppKey) é uma chave de aplicação única para cada dispositivo final. Ela é utilizada durante o processo de ativação OTAA para derivar as chaves de sessão necessárias (NwkSKey e AppSKey) através de um processo de cálculo de criptografia específico (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

A chave NwkSKey é uma chave de sessão usada para criptografar e autenticar os dados entre o dispositivo final e a rede LoRaWAN. Ela é derivada a partir da AppKey e é negociada dinamicamente durante o processo de ativação OTAA. A NwkSKey é compartilhada entre o dispositivo final e o servidor de rede LoRaWAN para garantir a confidencialidade e a integridade dos dados transmitidos (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

A AppSKey é outra chave de sessão derivada da AppKey durante o processo OTAA. Ela é usada para criptografar e autenticar os dados entre o dispositivo final e a aplicação de

servidor de rede (como o ChirpStack), garantindo assim que os dados sejam protegidos e que a integridade das mensagens seja verificada após a decodificação (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

Em resumo, o método OTAA representa uma abordagem robusta e segura para a ativação e operação de dispositivos finais em redes LoRaWAN, proporcionando não apenas segurança aprimorada dos dispositivos, mas também uma maior flexibilidade e eficiência operacional (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

4.3 Modos de Operação

O protocolo LoRaWAN define três classes de dispositivos para atender a diferentes requisitos de comunicação e energia dos nós da rede. Cada classe tem suas próprias características de operação, que afetam a forma como os dispositivos transmitem e recebem dados (RAHMAN *et al.*, 2020), (ROCHA *et al.*, 2023).

4.3.1 Classe A

Os dispositivos da Classe A são ideais para aplicações que exigem o menor consumo de energia possível, voltada principalmente para dispositivos alimentados por bateria e que transmitem dados de forma esporádica. Nesse modo, os dispositivos transmitem dados de forma assíncrona e iniciam duas janelas de recepção imediatamente após a transmissão. Essa abordagem garante que o dispositivo permaneça a maior parte do tempo em modo de baixo consumo, ativando apenas quando necessário (ORTIZ *et al.*, 2018), (ROCHA *et al.*, 2023), (ROCHA *et al.*, 2023).

A alta eficiência energética máxima é um dos destaques dessa classe, tornando-a ideal para dispositivos com restrições de energia. Em contrapartida, temos a limitação da comunicação de *downlink* (do servidor para o dispositivo), que só pode ocorrer após o dispositivo ter enviado uma mensagem de *uplink* (do dispositivo para o servidor) (ROCHA *et al.*, 2023).

4.3.2 Classe B

A Classe B é projetada para dispositivos que necessitam de janelas de recepção periódicas, permitindo uma comunicação mais previsível entre o dispositivo final e o servidor de rede. Além das janelas de recepção de Classe A, os dispositivos Classe B abrem janelas

de recepção adicionais em horários sincronizados, utilizando para isso, dados enviados pelo Gateway (ORTIZ *et al.*, 2018), (MATNI *et al.*, 2020), (ROCHA *et al.*, 2023).

A comunicação downlink mais frequente e previsível sem comprometer significativamente a vida útil da bateria é um dos maiores benefícios dessa classe. Por outro lado, existe um consumo maior de energia quando comparado com a Classe A, pois o modo requer sincronização frequente com o Gateway (MATNI *et al.*, 2020), (ROCHA *et al.*, 2023).

4.3.3 Classe C

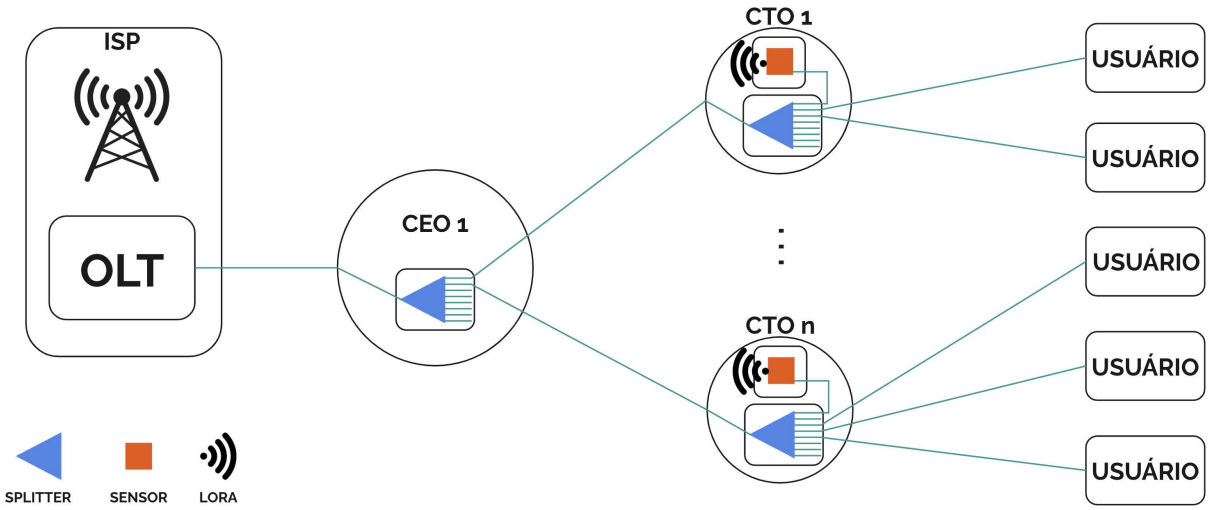
Os dispositivos da Classe C mantêm uma janela de recepção quase contínua, exceto durante a transmissão de dados, o que torna essa classe adequada para aplicações que exigem comunicação *downlink* constante. Essa característica faz com que os dispositivos dessa classe fiquem sempre prontos para receber mensagens, caso não estejam realizando transmissões (ORTIZ *et al.*, 2018), (MATNI *et al.*, 2020), (MILAROKOSTAS *et al.*, 2022), (ROCHA *et al.*, 2023).

A redução da latência na comunicação *downlink* dessa classe é um dos seus principais benefícios, tornando-a ideal para aplicações que necessitam de monitoramento em tempo real ou controle contínuo (MILAROKOSTAS *et al.*, 2022). Por outro lado, a disponibilidade para recepção de mensagens exige um consumo de energia significativamente maior, adequado apenas para dispositivos com uma fonte de energia contínua (MATNI *et al.*, 2020), (ROCHA *et al.*, 2023).

4.4 Aplicação em uma rede FTTH

Para se utilizar a rede LoRaWAN aplicada às redes FTTH para o monitoramento, se faz necessário incluir o *node* da rede de sensoriamento dentro de um componente da rede óptica. O componente escolhido foi a CTO, que fica mais próximo do usuário e está em maior número na rede, oferecendo a possibilidade de sensoriamento de uma maior quantidade de pontos da rede FTTH. A Figura 6 mostra um esquema dessa aplicação.

Figura 6 – Aplicação da rede LoRaWAN em uma rede FTTH



Fonte: Autoria própria (2024).

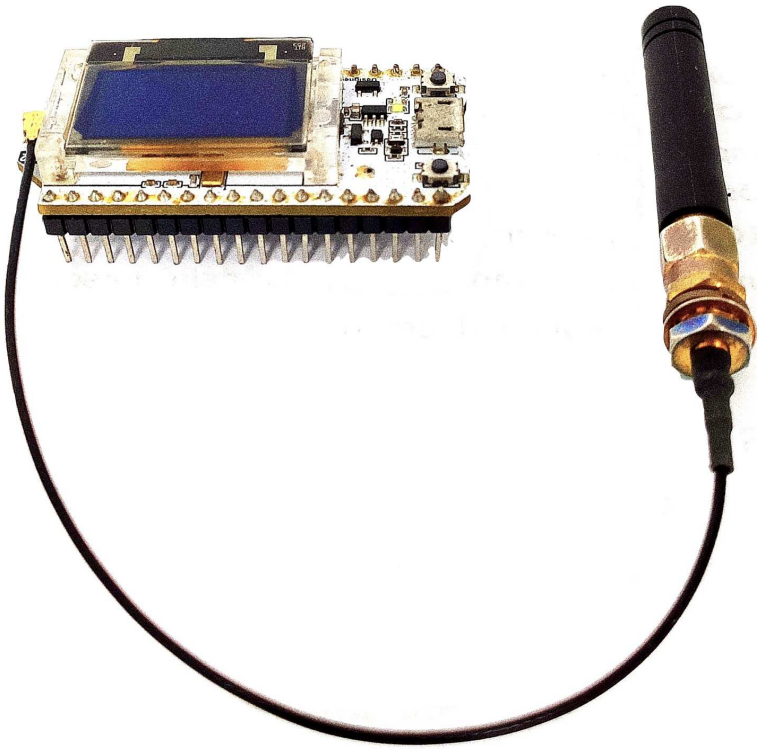
Para que o *node* consiga mensurar o sinal da rede óptica, o mesmo ficará acoplado a uma porta da CTO coletando sinal através de um equipamento de medição como o Power Meter LoRaWAN visto no trabalho do autor (NASCIMENTO, 2024). Para que o sensor não ocupe uma porta de atendimento da CTO, faz-se necessário o uso de um *splitter* desbalanceado de razão 1 entrada para 2 saídas e proporção de sinal de 1% para 99%, onde a saída de 99% do sinal será disponibilizada para atendimento de um usuário, e a saída de 1% será acoplada ao Power Meter LoRaWAN para medição do sinal.

5 MATERIAIS UTILIZADOS

5.1 Módulo de Transmissão ESP32 Wi-Fi LoRa

Nesta pesquisa, utilizamos como módulo de transmissão o microcontrolador ESP32 WiFi LoRa versão 2 (Figura 7), desenvolvido pela empresa Heltec. Este dispositivo é equipado com um chip SX1276, que possibilita a comunicação em uma rede LoRa, viabilizando comunicações em longas distâncias com baixo consumo de energia.

Figura 7 – Placa ESP32 com antena omnidirecional LoRa



Fonte: Autoria própria (2024)

O ESP32 WiFi LoRa v2 possui um microcontrolador dual-core Xtensa LX6, com interfaces Bluetooth, Wi-Fi e LoRa. Com uma frequência de operação de até 240Mhz e 520KB de memória Random Access Memory (RAM), o dispositivo oferece processamento eficiente e suporte para múltiplas tarefas. A placa também possui um display Oled 0,96"com resolução 128x64, que facilita a visualização em tempo real dos resultados dos testes realizados. Importante destacar que também é possível acoplar uma antena LoRa externa no ESP32, o que contribui para uma melhor qualidade do sinal de transmissão e recepção (HELTEC, 2024).

O rádio SX1276 integrado no dispositivo, é um transceptor LoRa desenvolvido pela Semtech. Este chip opera nas frequências 137MHz à 1020MHz, incluindo a banda 915MHz

homologada para uso no Brasil. A recepção de sinal pode chegar à -137dBm e sua potência de transmissão pode atingir +23dBm, permitindo comunicação em ambientes ruidosos e de longa distância (SEMTECH, 2024).

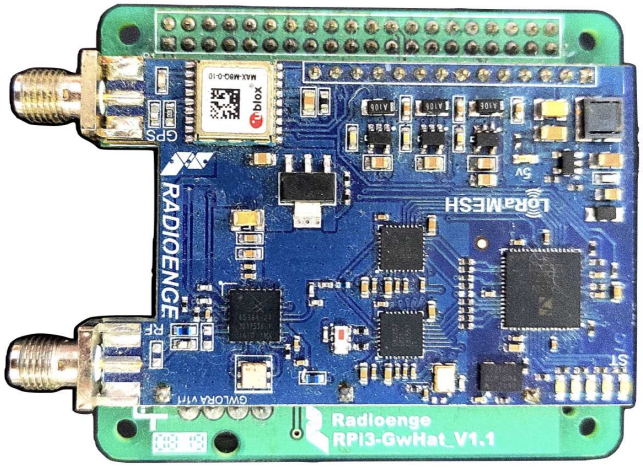
Nesta pesquisa, o dispositivo desempenhou o papel de nó da rede, responsável por receber o dado de sinal medido da rede FTTH.. Além disso, o ESP32 também foi responsável pelo envio desse indicador para um Gateway LoRa remoto, utilizando para essa comunicação o protocolo LoRaWAN e frequências da banda de 915Mhz.

O módulo foi utilizado em bancada de laboratório para a realização dos testes, mas na versão final do projeto, o mesmo ficará armazenado dentro de uma CTO (Caixa de Terminação Óptica) de uma rede de fibra óptica FTTH, com uma bateria como fonte de alimentação elétrica.

5.2 Gateway LoRaWAN

Para recepção dos dados provenientes do módulo de transmissão, utilizamos um um Gateway LoRa. A escolha da solução foi o modelo RD43HATGPS da empresa Radioenge (Figura 8), ao qual foi acoplada uma antena externa omnidirecional de 6dBi para a frequência de 915Mhz, além de outra antena dedicada à captação de sinal Global Positioning System (GPS).

Figura 8 – Gateway RD43HATGPS

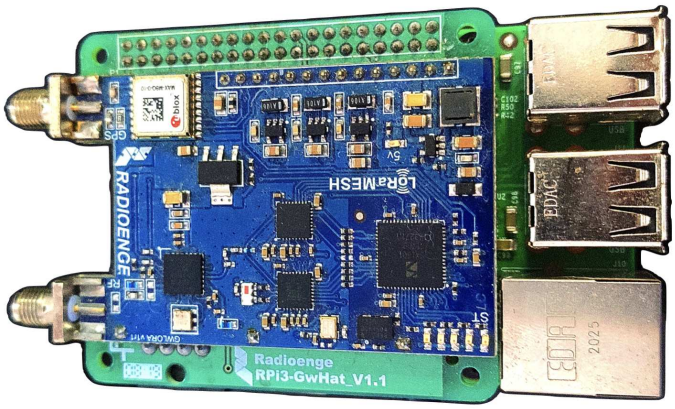


Fonte: Autoria própria (2024)

Para seu funcionamento, o RD43HATGPS foi integrado a um controlador Raspberry Pi 3 através da interface Serial Peripheral Interface (SPI) (Figura 9) e o conjunto foi alimentado por uma fonte DC de 5V e 3,5A. O Raspberry Pi 3 é responsável pelo processamento dos dados recebidos pelo Gateway e pelo envio dos mesmos ao servidor de rede. Este encaminhamento é realizado através da aplicação open-source Packet Forwarder, presente na Imagem de Sistema

Operacional (ISO) disponibilizada no site da fabricante do *gateway*.

Figura 9 – Gateway com Raspberry Pi 3



Fonte: Autoria própria (2024)

O Gateway faz uso da modulação CSS/LoRa, o que viabiliza uma comunicação de longo alcance para projetos de IoT. O dispositivo também conta com 8 canais simultâneos para recepção de dados, garantindo maior capacidade e eficiência no processamento de múltiplos dispositivos finais. Este Gateway da Radioenge também trabalha com o protocolo LoRaWAN, o que assegura interoperabilidade com outros dispositivos e sistemas.

O RD43HATGPS opera na faixa de frequência de 915MHz, adequada para operação no Brasil, e oferece largura de banda de 125 kHz à 500kHz, permitindo ajustes que otimizam a transmissão conforme as necessidade da aplicação. A sensibilidade de recepção do dispositivo é de até -137dBm, o que assegura comunicação com baixa quantidade de sinal provenientes dos nós remotos. Com relação à transmissão, o *gateway* consegue operar com até +27dBm de potência (RADIOENGE, 2024). A Figura 10 mostra o resumo das suas especificações técnicas.

Figura 10 – Especificações do Gateway RD43HATGPS

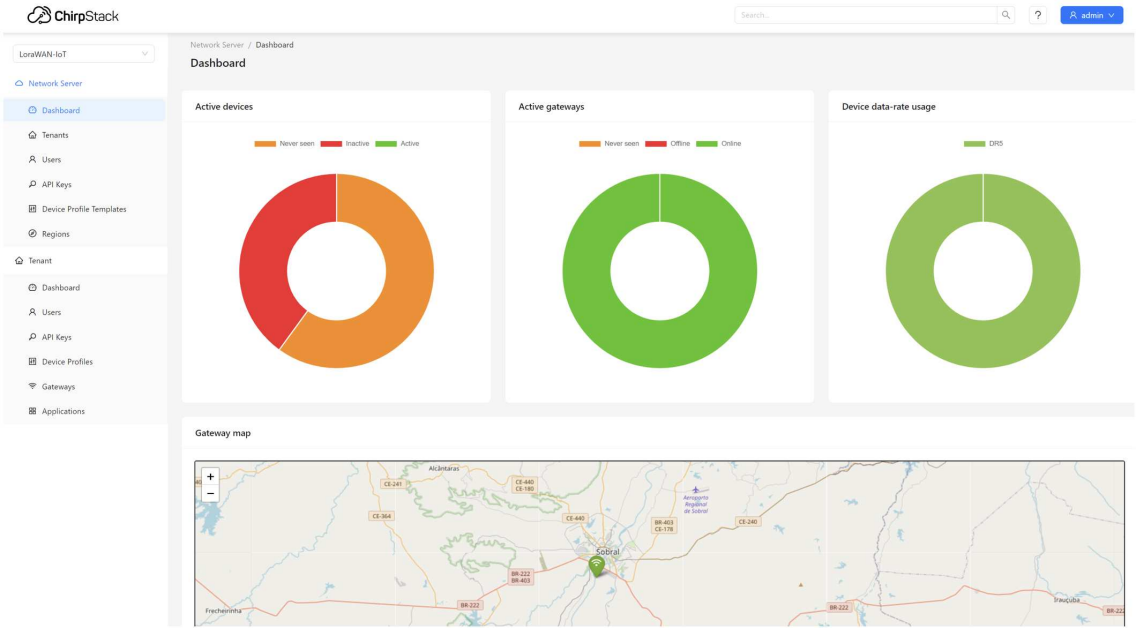
Especificação	Descrição
Frequência	902-928 Mhz Operação Half-Duplex (TDD)
Taxa de Dados RF	21,9 Kbps (efetivo máximo)
Largura de Banda	125 Khz / 250 Khz / 500 Khz
Potência de Saída	0,5 W / +27dBm
Tolerância de Frequência	~5,0 ppm
Modulação/Espalhamento	LoRa / CSS (Chirp Spread Spectrum)
Sensibilidade (BER 0,1%)	-137 dBm
Nível Máximo de Entrada	-20 dBm operação normal / 0 dBm máximo
Conector	SMA-M

Fonte: Autoria Própria (2024)

5.3 Servidor de Rede LoRaWAN

No presente estudo, adotou-se o sistema de código aberto ChirpStack como servidor de rede LoRa. O ChirpStack é uma plataforma que realiza o gerenciamento completo da rede LoRaWAN, comunicando-se com o Gateway para troca de informações e estabelecimento de parâmetros, como potência de transmissão, criptografia dos dados, taxa de transmissão e payload contendo as medições de sinal da rede óptica enviadas pelo ESP32. A aplicação também suporte gerenciamento WEB, como pode ser visualizado na Figura 11.

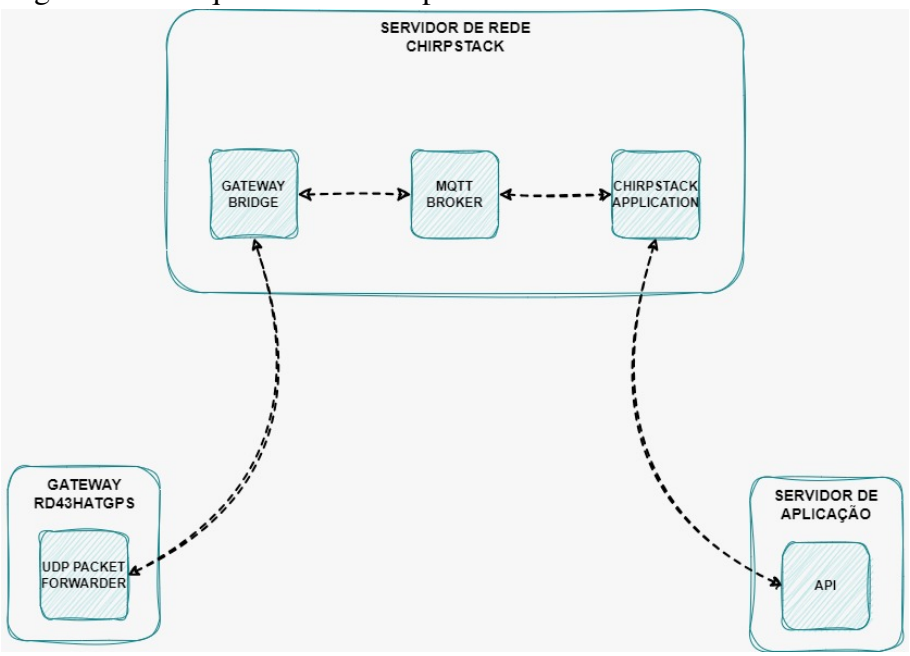
Figura 11 – Dashboard inicial do ChirpStack



Fonte: Autoria Própria (2024)

Para suportar a operação do ChirpStack, foram utilizadas algumas aplicações essenciais. O Broker Message Queuing Telemetry Transport (MQTT) Mosquitto e o Mosquitto-clients v2.0.11 foram responsáveis pela comunicação MQTT, que é o protocolo de mensagens utilizado para a transmissão de dados entre os dispositivos e o servidor, assim como entre o servidor de rede e servidor de aplicação. O banco de dados Postgresql v14.11 foi escolhido para armazenar e gerenciar os dados coletados e as configurações da rede. O ChirpStack Gateway Bridge v4.0.10 atuou como encaminhador de pacotes, viabilizando a comunicação entre o Gateway e Broker MQTT. A aplicação principal foi o ChirpStack v4.6.0, que gerencia todas as operações do servidor de rede (CHIRPSTACK, 2024). Um esquema de funcionamento do ChirpStack, contendo os elementos da rede e sua arquitetura, pode ser visualizado na Figura 12.

Figura 12 – Arquitetura do ChirpStack



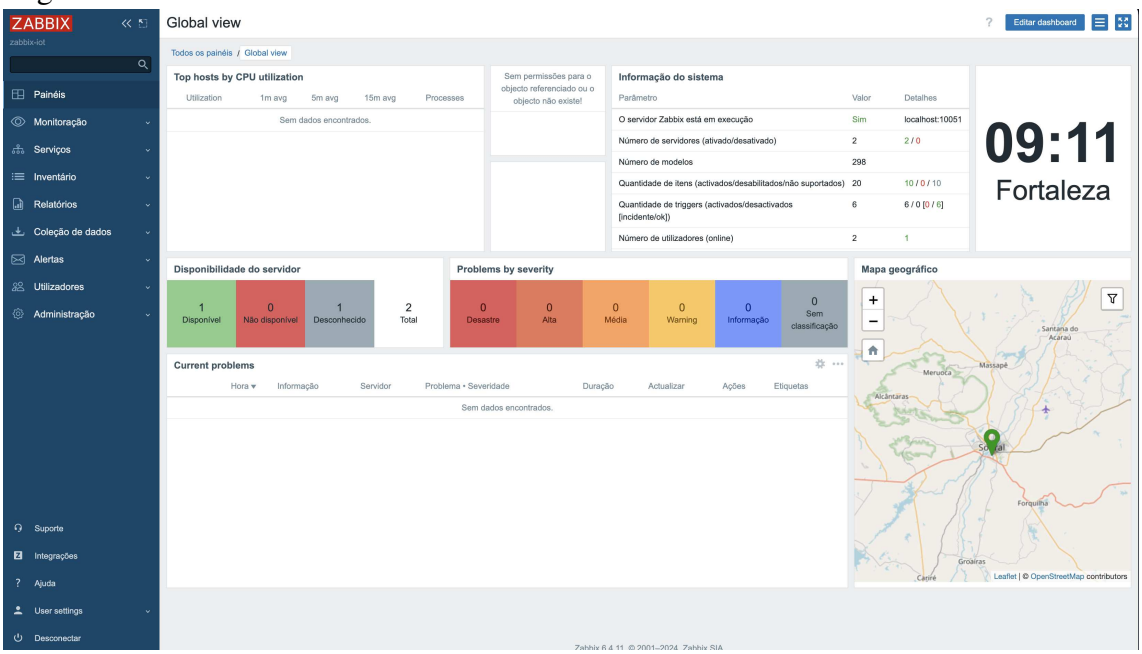
Fonte: Autoria Própria (2024)

A instalação foi realizada em uma máquina virtual, utilizando o programa VM Virtual Box e sistema operacional Ubuntu 22.04.3 LTS. Para assegurar o desempenho adequado, foram reservados 10 GB de espaço em disco, 2.048 MB de memória RAM e 1 CPU 2,8Ghz. Essa configuração foi suficiente para garantir a operação estável e eficiente do servidor de rede durante os testes.

5.4 Servidor de Aplicação

Para recepção das informações provenientes do servidor de rede, foi essencial implementar um sistema de armazenamento e tratamento de dados capaz de gerenciar os indicadores da rede. Neste estudo, optou-se pelo Zabbix, uma plataforma open source amplamente utilizada para gerenciamento de redes, servidores e dispositivos IoT. A flexibilidade proporcionada pelo Zabbix, através de sua capacidade de comunicação via API com diversas soluções, possibilitou a integração eficaz com o servidor de rede ChirpStack. Esta integração viabilizou a coleta contínua de dados da rede monitorada, assim como sua organização em Dashboards interativos para análises e tomadas de decisão. A Figura 13 mostra um Dashboard elaborado para esta pesquisa, com alguns gráficos de informações a respeito da rede LoRa.

Figura 13 – Tela de Dashboards no servidor Zabbix



Fonte: Autoria Própria (2024)

O Zabbix Server v6.4.11 foi a versão utilizada, suportada por um banco de dados MySQL v8.0.36 e hospedado em um servidor WEB Apache v2.4.52. Essa configuração proporcionou um ambiente estável para o armazenamento e processamento dos dados coletados durante os testes. A instalação da aplicação foi realizada em uma máquina virtual utilizando a mesma quantidade de recursos reservados para o servidor de rede.

6 DESENVOLVIMENTO

6.1 Configuração do node ESP32

Para configuração do ESP32, utilizamos a aplicação Arduíno IDE v2.3.2., assim como as bibliotecas *TTN_esp32.h v*0.1.6 e *TTN_CayenneLPP.h v*0.2.0 para coleta e envio dos dados pela rede LoRa. A biblioteca *heltec.h v1.1.5* também foi utilizada para exibição dos dados no display do ESP32, o que facilitou os testes realizados. Embora essas bibliotecas tenham sido inicialmente desenvolvidas para uso com o servidor The Things Network (TTN), elas possibilitaram mudanças em alguns parâmetros para que a comunicação fosse realizada com o servidor ChirpStack. As principais alterações se deram no arquivo *lmic_project_config.h*, no qual foi necessária a inserção dos códigos *#define CFG_au*915 1 e *#define CFG_sx*1276*_radio* 1 (Figura 14), para garantir a compatibilidade com o *hardware* SX1276 do ESP32 e a comunicação LoRa ocorresse no range de frequência homologada pela Anatel.

Figura 14 – Configurações da biblioteca TTN

lmic_project_config.h

```
1  // project-specific definitions
2  //#define CFG_eu868 1
3  //#define LMIC_REGION_eu868 1
4  #define CFG_au915 1
5  //#define CFG_au921 1
6  //#define CFG_as923 1
7  // #define LMIC_COUNTRY_CODE LMIC_COUNTRY_CODE_JP   /* for as923-JP */
8  //#define CFG_kr920 1
9  //#define CFG_in866 1
10 #define CFG_sx1276_radio 1
11 //#define LMIC_USE_INTERRUPTS
12 #define LMIC_LORAWAN_SPEC_VERSION    LMIC_LORAWAN_SPEC_VERSION_1_0_3
13 #define LMIC_MAX_FRAME_LENGTH 64
14 #define hal_init LMICHAL_init
```

Fonte: Autoria Própria (2024)

Como método de autenticação da rede LoRaWAN, foi utilizado o protocolo ABP (Activation By Personalization), que requer uso dos parâmetros:

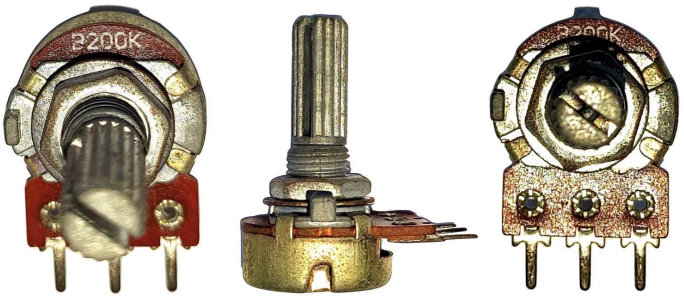
a) DevAddr: Um endereço de 32 bits composto pelo identificador de rede, utilizado para diferenciar endereços em redes próximas, e pelo endereço de rede do nó, que pode ser gerado aleatoriamente;
b) NwkSKey: Uma chave de sessão da rede de tamanho 128 bits, utilizada para interação entre o ESP32 e o ChirpStack;
c) AppSKey: Uma chave de sessão da aplicação de tamanho 128 bits, utilizada para criptografia da carga útil de dados, ou payload.

Nesse método de ativação, a combinação das chaves NwkSKey e AppSKey é única por dispositivo, permanecendo a mesma durante toda a comunicação. A geração inicial dessas

chaves se deu no servidor ChirpStack, posteriormente as chaves foram copiadas e inseridas no código do ESP32 em variáveis correspondentes. Utilizando então os métodos *ttn.begin()* e *ttn.personalize()*, a comunicação foi iniciada entre o ESP32 e o ChirpStack.

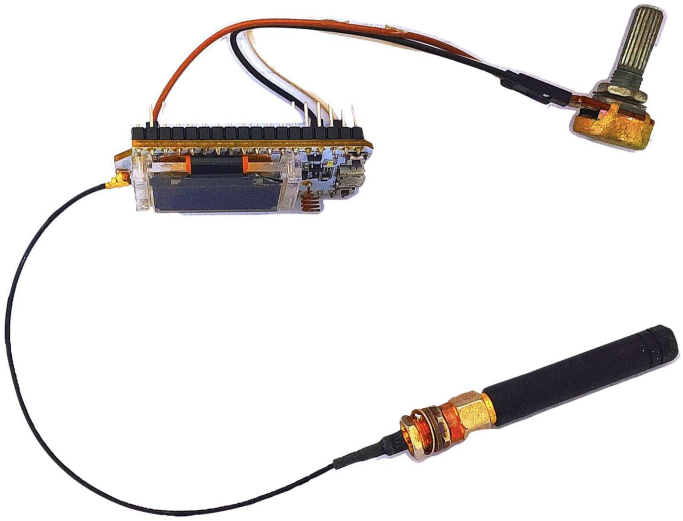
Para a simulação do sensor de coleta de sinal, utilizamos um potenciômetro B200K (Figura 15) acoplado aos pinos analógicos da placa ESP32, como mostra a Figura 16.

Figura 15 – Potenciômetro B200K



Fonte: Autoria Própria (2024)

Figura 16 – ESP32 com potenciômetro



Fonte: Autoria Própria (2024)

No código implementado, foi utilizada a função textitanalogRead(SINAL) para leitura do sinal do potenciômetro. A função *map()* por sua vez, foi necessária para transformar o sinal analógico em um valor representativo de sinal da rede óptica, para os testes utilizamos valores entre -50 e +5. A partir de então, a função $lpp.addAnalogInput()$ foi utilizada para inserir o valor do sinal no *buffer*. Em seguida, foi chamada a função $ttn.sendBytes()$ para enviar os dados via LoRa para o Gateway e o servidor de rede, permitindo as simulações de sinal através do ajuste do potenciômetro. A Figura 17 mostra o código implementado, com algumas funções adicionais para monitoramento dos parâmetros na tela do ESP32.

Figura 17 – Código implementado no ESP32

LORA-ABP-POTENCIOMETRO.ino

```
#include <TTN_esp32.h>
#include <heltec.h>
#include "TTN_CayenneLPP.h"

const char* devAddr = "260D7932"; // Change to TTN Device Address
const char* nwkSKey = "1A6737DD116CF2CBDE2443C97E6F41CD"; // Change to TTN Network Session Key
const char* appSKey = "ACA956656CB8C50826DFF29622596219"; // Change to TTN Application Session Key

TTN_esp32 ttn ;
TTN_CayenneLPP lpp;

#define BOTAO 13

void setup()
{
    pinMode(BOTAO, INPUT);
    Serial.begin(115200);
    Heltec.begin(true, true, true);
    Heltec.display->setFont(ArialMT_Plain_24);
    Heltec.display->clear();
    ttn.begin();
    ttn.personalize(devAddr, nwkSKey, appSKey);
}

void loop()
{
    float sinal = analogRead(BOTAO);
    sinal = map(sinal, 0, 4095, -50, 5);
    lpp.reset();
    lpp.addAnalogInput(1, sinal);
    if (ttn.sendBytes(lpp.getBuffer(), lpp.getSize()))
    {
      Serial.println("SINAL: " + String(sinal));
    }
    ttn.showStatus();
    if (ttn.isRunning()) {
      Heltec.display->drawString(0, 0, "TTN: OK");
      Heltec.display->drawString(0, 30, "SIG: " + String(sinal));
    } else {
      Heltec.display->drawString(0, 0, "TTN: OFF");
    }
    Heltec.display->display();
    delay(1000);
    Heltec.display->clear();
}
```

Fonte: Autoria Própria (2024)

Estas configurações garantiram o pleno funcionamento do *node* ESP32, atuando na recepção do valor de sinal e na transmissão do mesmo através da rede LoRaWAN.

6.2 Configuração do Gateway RD43HATGPS

Para utilização do *Gateway*, inicialmente foi realizada a instalação do sistema disponibilizado pela Radioenge em um Raspberry Pi 3, através de um cartão de memória MicroSD 16G. Este sistema já inclui todos os programas necessários para o funcionamento do dispositivo e já vêm pré-configurado, facilitando a instalação e configuração inicial.

Após sua instalação, o *Gateway* foi conectado à rede local através de um cabo em sua interface *ethernet*. A partir de então, o dispositivo recebeu um endereço de Internet Protocol (IP) via Dynamic Host Configuration Protocol (DHCP), possibilitando o acesso ao mesmo por Secure Shell (SSH). A configuração inicial realizada foi a geração do ID do *Gateway* através do método $update\_gwid.sh$, presente no diretório $opt/LoRa/packet\_forwarder/lora\_pkt\_fwd/$. Esse

identificador é único, construído a partir do endereço Media Access Control (MAC) do Raspberry Pi. Posteriormente, esse ID foi inserido no campo *gateway_ID* do arquivo *global_conf.json*, para registro do dispositivo durante a comunicação com o servidor de rede Chirpstack.

Outra configuração realizada no arquivo *global_conf.json*, foi a alteração das frequências para a banda de 915Mhz à 928Mhz, que são permitidas para uso no Brasil e compatíveis com àquelas do rádio SX1276 presente no ESP32. Esta etapa garantiu que a comunicação ocorresse dentro das regulamentações locais da Anatel.

Como última configuração no mesmo arquivo, foram adicionados os dados do servidor de rede Chirpstack para que o Gateway estabelecesse a comunicação. Na presente pesquisa, os dois dispositivos foram adicionados à mesma rede local, possibilitando o uso do IP no arquivo. Também foi utilizada a porta padrão 1700 User Datagram Protocol (UDP) para o tráfego de entrada e saída entre *gateway* e servidor de rede.

Após a realização destas configurações, o dispositivo foi reiniciado para que os serviços atualizassem seus parâmetros de funcionamento. Este processo garantiu que o *gateway* estivesse corretamente configurado para receber e encaminhar os dados coletados pelos nós da rede para o servidor ChirpStack, utiliziando o protocolo LoRaWAN.

6.3 Configuração do Servidor de Rede ChirpStack

A instalação do ChirpStack e das aplicações necessárias para o seu funcionamento, foram realizadas conforme a documentação oficial do projeto (CHIRPSTACK, 2024).

Inicialmente, configuramos a região de frequência no arquivo *chirpstack.toml*, habilitando as opções *au*915_0 e *au*915_1 dentro da chave *enabled_regions* (Figura 18). Esses parâmetros são essenciais para garantir que o sistema opere dentro das regulamentações de frequência locais. Os detalhes específicos de cada uma destas frequências estão pré-configurados em arquivos presentes no mesmo diretórios, o que facilita a configuração inicial.

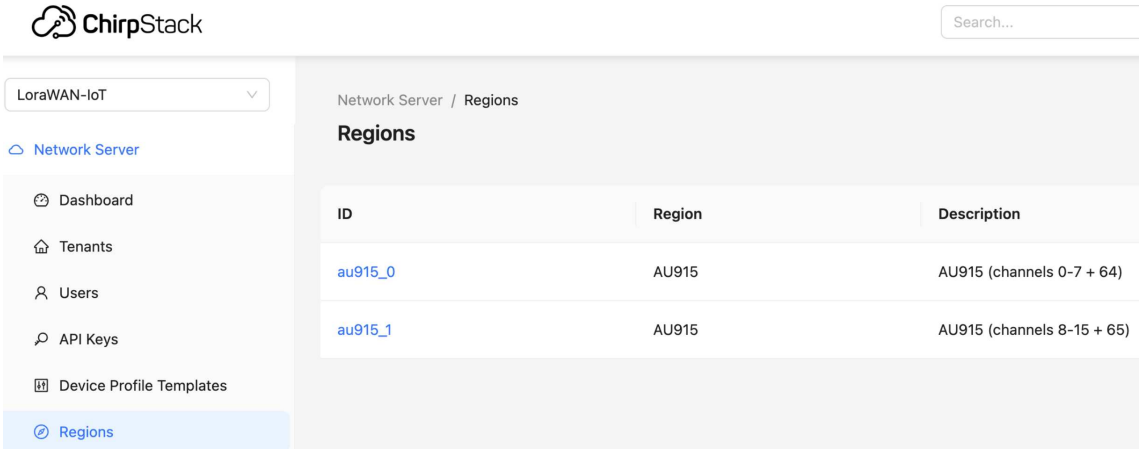
Figura 18 – Configurações regionais no ChirpStack

```
# Network related configuration.
[network]

  # Network identifier (NetID, 3 bytes) encoded as HEX (e.g. 010203).
  net_id="000000"

  # Enabled regions.
  #
  # Multiple regions can be enabled simultaneously. Each region must match
  # the 'name' parameter of the region configuration in '[[regions]]'.
  enabled_regions=[
     "au915_0",
     "au915_1",
  ]
```

Fonte: Autoria Própria (2024)

Após estas configurações, é possível visualizar e escolher os parâmetros de região através da interface WEB de gerenciamento Chirpstack, como pode ser visto na Figura 19.

Figura 19 – Configurações regionais das frequências no ChirpStack



Fonte: Autoria Própria (2024)

No mesmo arquivo *chirpstack.toml*, configuramos alguns parâmetros do protocolo MQTT (Message Queuing Telemetry Transport), que é responsável pela troca de mensagens entre os dispositivos da rede LoRa, o ChirpStack e servidor de aplicação (Figura 20).

Figura 20 – Configurações MQTT no ChirpStack

```
# Integration configuration.
[integration]
# Payload marshaler.
#
# This defines how the MQTT payloads are encoded. Valid options are:
# * protobuf:  Protobuf encoding
# * json:      JSON encoding (for debugging)
marshaler="json"

  # MQTT integration configuration.
  [integration.mqtt]
  # Event topic template.
  event_topic_template="au915_1/gateway/{{ .GatewayID }}/event/{{ .EventType }}"

  # State topic template.
  #
  # States are sent by the gateway as retained MQTT messages (by default)
  # so that the last message will be stored by the MQTT broker. When set to
  # a blank string, this feature will be disabled. This feature is only
  # supported when using the generic authentication type.
  state_topic_template="au915_1/gateway/{{ .GatewayID }}/state/{{ .StateType }}"
```

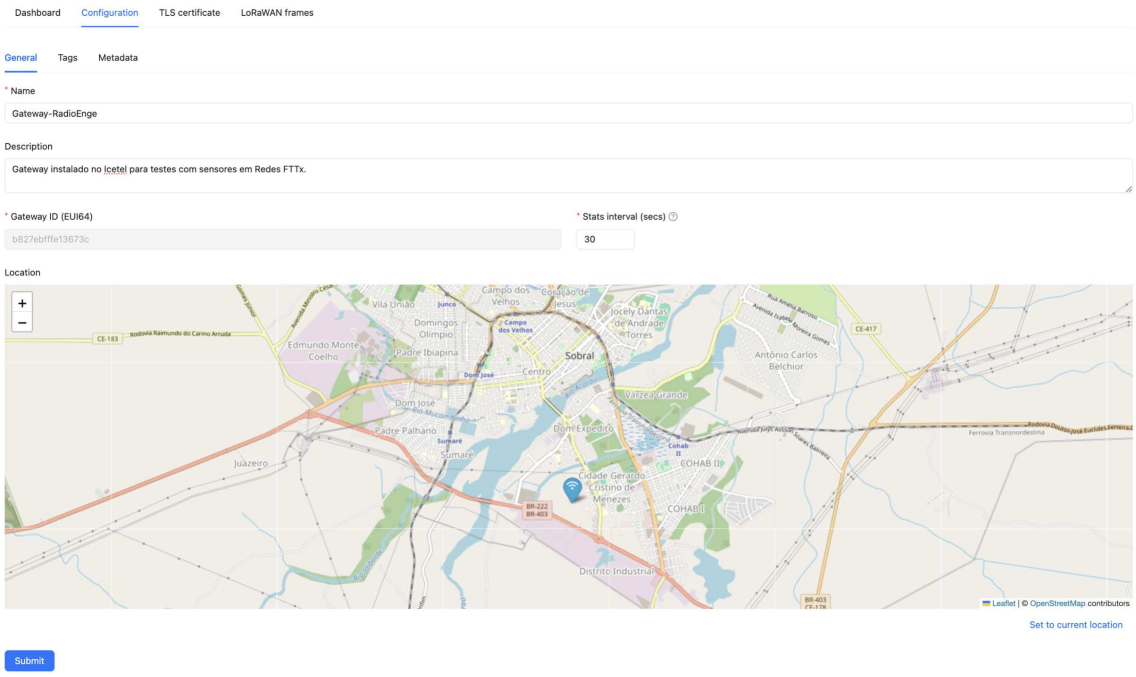
Fonte: Autoria Própria (2024)

O MQTT é um protocolo de comunicação leve, ideal para dispositivos IoT, que facilita a transmissão de dados de maneira eficiente e segura. As configurações dos tópicos MQTT são essenciais para a publicação dos dados pelo *Gateway*, bem como para a coleta destes dados pelo servidor de aplicação. Também foram configurados neste arquivo parâmetros como

usuário, senha e porta de acesso. Os arquivos contendo todas as configurações estão presentes nos Apêndices A e B desta dissertação.

O próximo passo realizado foi o cadastro do Gateway da rede no servidor ChirpStack (Figura 21). Essa atividade foi realizada acessando a interface WEB de gerenciamento pelo IP do servidor ChirpStack e a porta padrão 8080. Para o cadastro, foi necessário criar um nome para o Gateway e inserir o ID único, presente no arquivo de configuração do Raspberry Pi, mencionado anteriormente. O ChirpStack utiliza esse ID para identificar e autenticar o Gateway na rede.

Figura 21 – Tela de cadastro do Gateway no ChirpStack

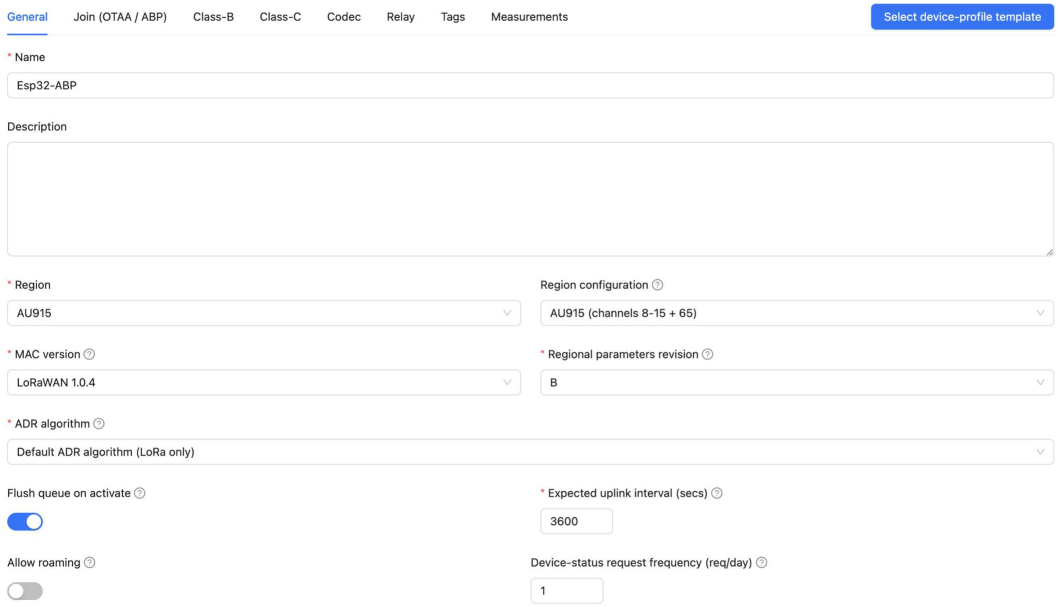


Fonte: Autoria Própria (2024)

Após a realização do cadastro, o Chirpstack reconhece pacotes provenientes do Gateway e a região de frequência utilizada pelo mesmo, exibindo o status de ativo caso o mesmo continue em comunicação com o ChirpStack.

Em seguida, configuramos os parâmetros relacionados ao Node ESP32. Inicialmente criamos um Device Profile no ChirpStack, especificando os dados regionais, o algoritmo de Adaptative Data Rate (ADR) e versão do protocolo LoRaWAN utilizado (Figura 22). O ADR é uma funcionalidade importante que ajusta automaticamente a taxa de dados do dispositivo para otimizar a comunicação e economizar energia. Na ativação da classe de operação, utilizamos apenas a Classe A nesta pesquisa, pois a ideia inicial é viabilizar a comunicação na rede, podendo a Classe B também ser utilizada posteriormente na solução como uma melhoria.

Figura 22 – Configuração do Device Profile no ChirpStack

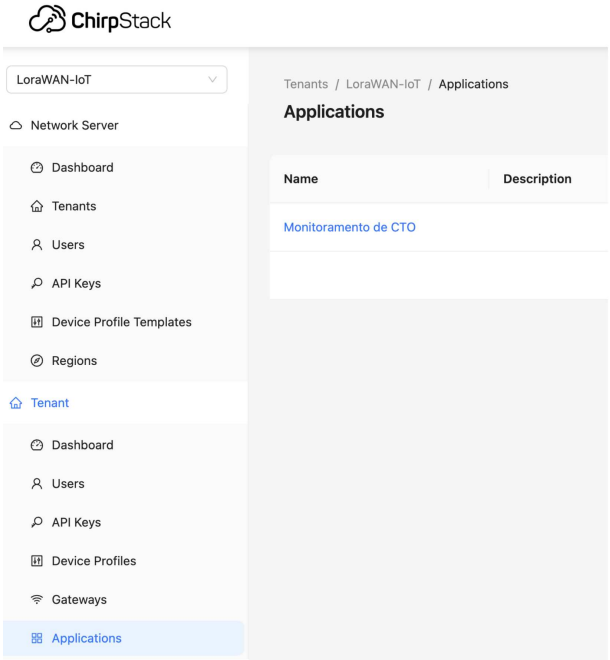


Fonte: Autoria Própria (2024)

Na seção *Codec*, configuramos o Cayenne Low Power Payload (LPP), utilizado pelo ESP32 para envio dos dados medidos do sensor. O Cayenne LPP é um codec eficiente para codificação de dados de sensores IoT. Na seção *Measurements* configuramos a variável *analoginput*_1, utilizada no ESP32 para a medição e armazenamento temporário do sinal lido.

Para gerar os dados de autenticação e incluir o *node* ESP32 no ChirpStack, criamos uma seção para o projeto dentro da opção *Applications*, como é mostrado na Figura 23.

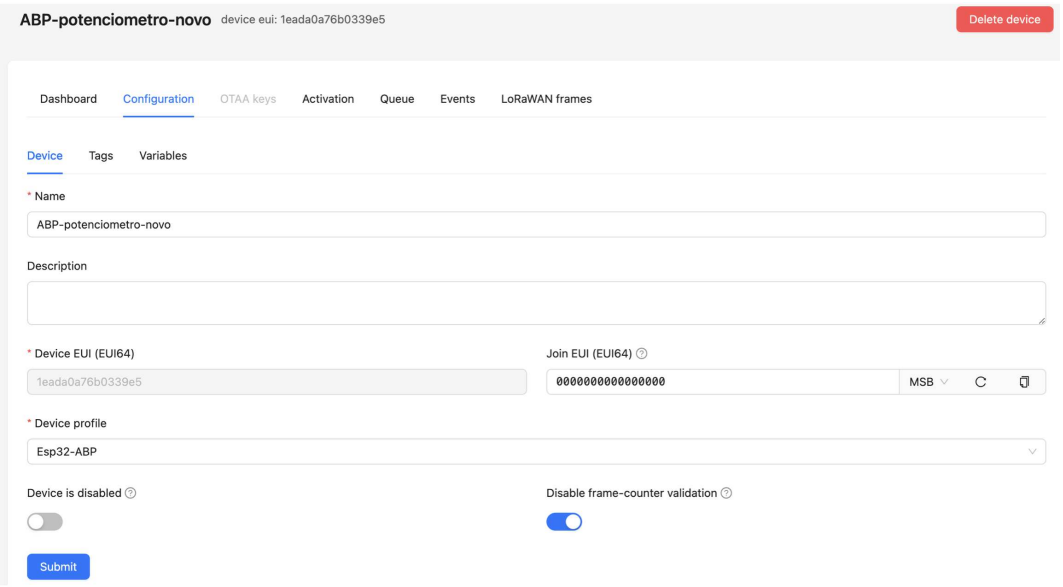
Figura 23 – Cadastro de app no ChirpStack



Fonte: Autoria Própria (2024)

Dentro da aplicação criada, é possível adicionar diversos dispositivos ou *nodes*. No caso desta pesquisa, geramos o Device End Unique Identifier (EUI) no cadastro do ESP32 e adicionamos um nome. Também foi necessário selecionar o Device Profile que foi criado anteriormente, como pode ser visualizado na Figura 24.

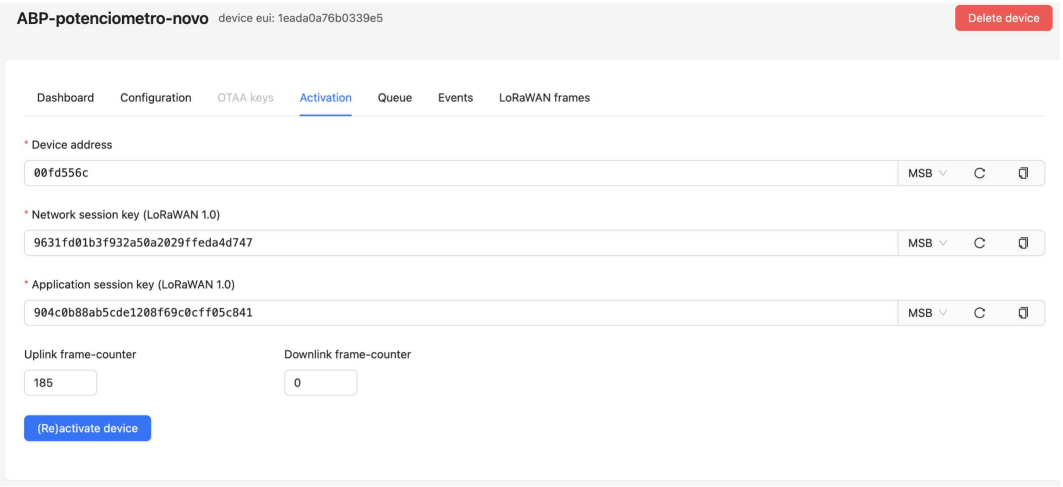
Figura 24 – Cadastro de um device no ChirpStack



Fonte: Autoria Própria (2024)

Após a criação do *device*, realizamos a geração das chaves de ativação do modo ABP (Activation By Personalization), dentro da seção *Activation*. O ABP é um método de ativação que permite configurar diretamente os parâmetros de sessão do dispositivo, facilitando a conexão com a rede. Os campos de geração das chaves podem ser visualizados na Figura 25.

Figura 25 – Configurações de ativação de device no ChirpStack



Fonte: Autoria Própria (2024)

Estas chaves foram então configuradas dentro do código do ESP32, nas variáveis DevAddr, NwkSKey e AppSKey mencionadas anteriormente. Ao inserir o código na placa, o dispositivo passou a se conectar ao ChirpStack através do *gateway*.

6.4 Configuração do Servidor de Aplicação Zabbix

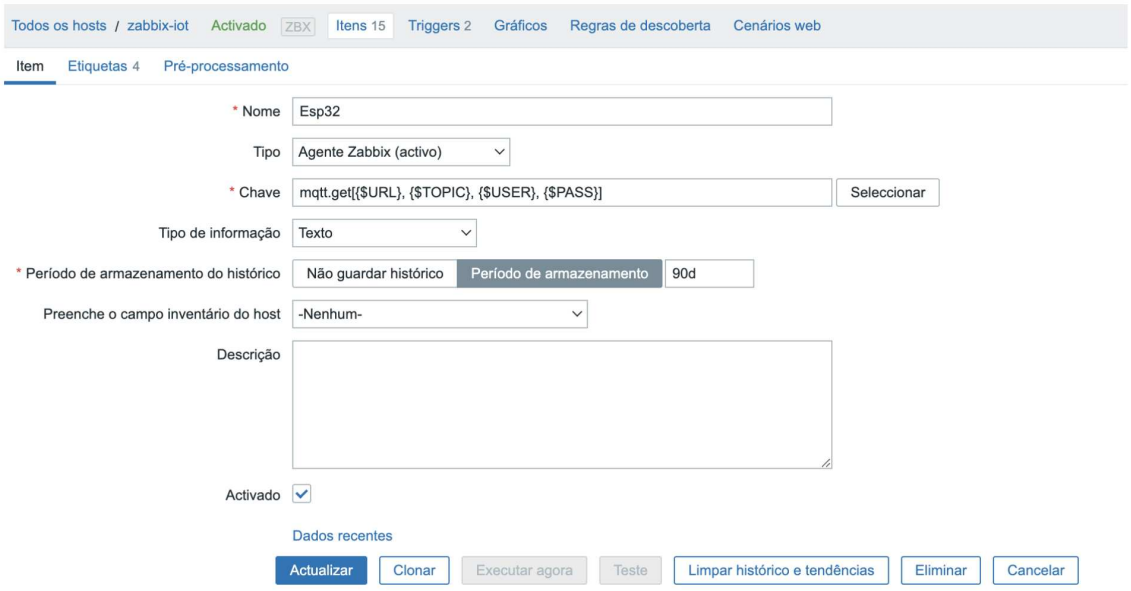
A instalação da plataforma Zabbix e das aplicações necessárias para o seu funcionamento, foram realizadas conforme documentação do site oficial da mesma (ZABBIX, 2024). Dentro da ferramenta, foi utilizado o *Agent Active*, uma solução que permite a coleta proativa de dados, ideal para ambientes onde o monitoramento contínuo e em tempo real é crucial.

Inicialmente foi realizado o cadastro de um servidor no menu *Coleção de Dados* > *Servidores*. O host criado recebeu o nome zabbix-iot, sendo então adicionado ao grupo de hosts *Zabbix Servers*, que já vem pré-configurado na instalação do sistema. Também foi necessário associar o template Zabbix Agente Active no cadastro desta instância.

O uso do Zabbix Agent Active foi necessário para obter mensagens MQTT do servidor de rede ChirpStack de maneira ativa. Esse agente permite estabelecer conexões bidirecionais, onde o Zabbix pode enviar comandos e receber dados de dispositivos através do protocolo MQTT. Isso é fundamental para a coleta eficiente e confiável de informações da rede LoRaWAN, garantindo que os dados sejam atualizados em tempo real no monitoramento do Zabbix.

Após a criação do servidor, foi acessada a seção *Itens* para criar os dispositivos IoT e definir os dados a serem coletados. O primeiro item criado foi o Esp32, que serviu como base para a coleta de dados geral deste dispositivo. O tipo de agente utilizado no cadastro foi o *Agente Zabbix (active)*. A chave deste item foi configurada com o método nativo *mqtt.get[$URL, $TOPIC, $USER, $PASS]*, permitindo a conexão via MQTT com o ChirpStack para a coleta de dados da rede LoRaWAN. Na seção *Etiqueta*, foram preenchidos os valores das chaves conforme as configurações mencionadas nos tópicos anteriores desta pesquisa (Figura 26).

Figura 26 – Cadastro de item na interface do Zabbix



Fonte: Autoria Própria (2024)

Inicialmente testamos a comunicação MQTT sem parâmetros de usuário e senha. A Uniform Resource Locator (URl) utilizada foi o IP e a Porta de serviço do Broker MQTT presente no ChirpStack. O tópico utilizado foi a chave #, que retorna todos os dados disponíveis de todos os tópicos utilizados pelo Broker MQTT. A Figura 27 mostra a configuração das chaves utilizadas pelo método *mqtt.get* do Zabbix.

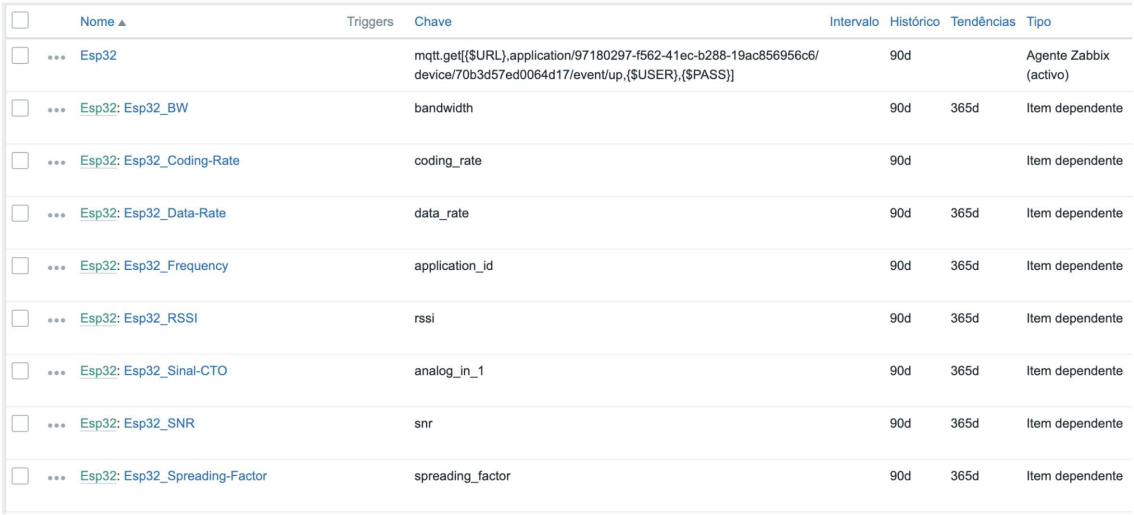
Figura 27 – Configuração das chaves MQTT na interface do Zabbix

Fonte: Autoria Própria (2024)

Após o cadastro do item, foi então estabelecida a comunicação do Zabbix com os outros elementos da rede, possibilitando a coleta de dados brutos no formato JavaScript Object Notation (JSON) via mensagens MQTT. Com essas informações, foi possível a criação de mais itens no Zabbix, cada um com indicadores específicos. Os próximos itens criados, foram do tipo

Item Dependente, no qual selecionamos o ESP32 como item principal e, a partir deste, o item secundário realizou um segundo filtro de dados para coletar a informação desejada. A Figura 28 mostra a lista de itens criados para a coleta de informações da rede.

Figura 28 – Lista de itens criados no Zabbix Server



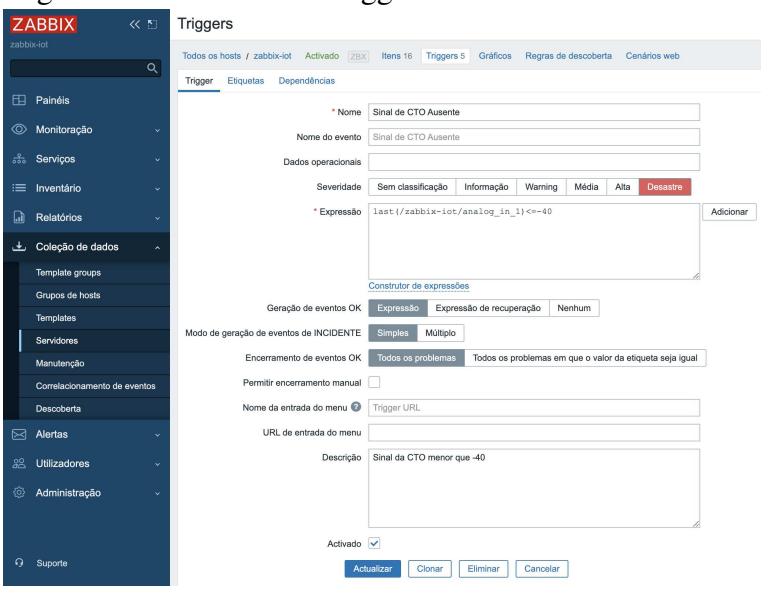
Nome ▲	Triggers	Chave	Intervalo	Histórico	Tendências	Tipo
Esp32		mqtt.get[{$URL},application/97180297-f562-41ec-b288-19ac856956c6/device/70b3d57ed0064d17/event/up,{$USER},{$PASS}]		90d		Agente Zabbix (activo)
Esp32: Esp32_BW		bandwidth		90d	365d	Item dependente
Esp32: Esp32_Coding-Rate		coding_rate		90d		Item dependente
Esp32: Esp32_Data-Rate		data_rate		90d	365d	Item dependente
Esp32: Esp32_Frequency		application_id		90d	365d	Item dependente
Esp32: Esp32_RSSI		rssi		90d	365d	Item dependente
Esp32: Esp32_Sinal-CTO		analog_in_1		90d	365d	Item dependente
Esp32: Esp32_SNR		snr		90d	365d	Item dependente
Esp32: Esp32_Spreading-Factor		spreading_factor		90d	365d	Item dependente

Fonte: Autoria Própria (2024)

Para cada item dependente criado, foi informada a chave JSON onde o dado se encontrava. O principal dado para esta pesquisa, foi o sinal da CTO, que se encontra dentro da chave *analog_in_1*. A configuração detalhada desses itens garantiu uma coleta precisa e organizada dos dados da rede LoRaWAN, facilitando a análise e monitoramento contínuo através do Zabbix.

Outra funcionalidade implementada nesta pesquisa, foi a criação de um alerta no Zabbix, para que o NOC seja informado no caso de algum evento na rede. Para isso, inicialmente foi realizado o cadastro de um *trigger*, que funciona como um gatilho que gera um evento no servidor e também uma ação escolhida. A Figura 29 mostra a tela cadastro através do menu *Coleção de Dados* > *Servidores* > *Triggers*.

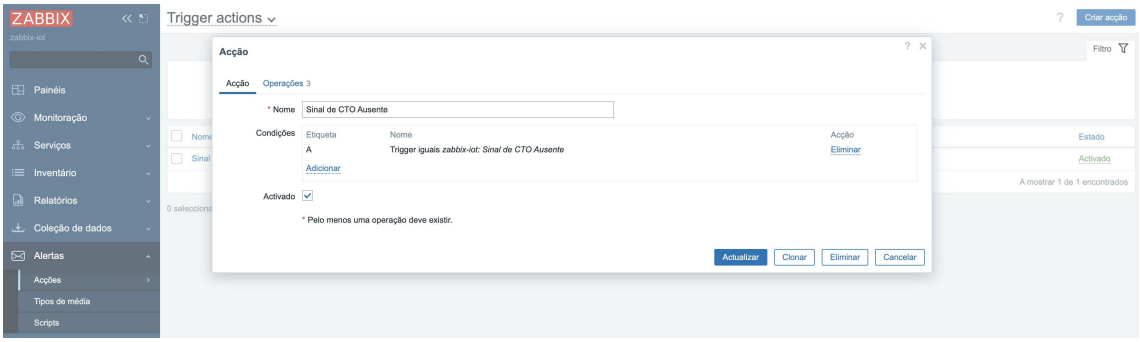
Figura 29 – Cadastro de trigger no Zabbix



Fonte: Autoria Própria (2024)

Nesse cadastro, utilizamos a expressão *last(/zabbix-iot/analog_in_1)<=-40* para indicar a geração do alerta de ausência de sinal no caso do valor medido ficar menor ou igual à -40dBm. Também selecionamos o tipo de alerta como "Desastre", que é a de maior prioridade dentro do Zabbix. Posteriormente, realizamos o cadastro de uma ação a ser tomada no caso desse alerta ser emitido. A Figura 30 mostra a tela de cadastro de uma ação.

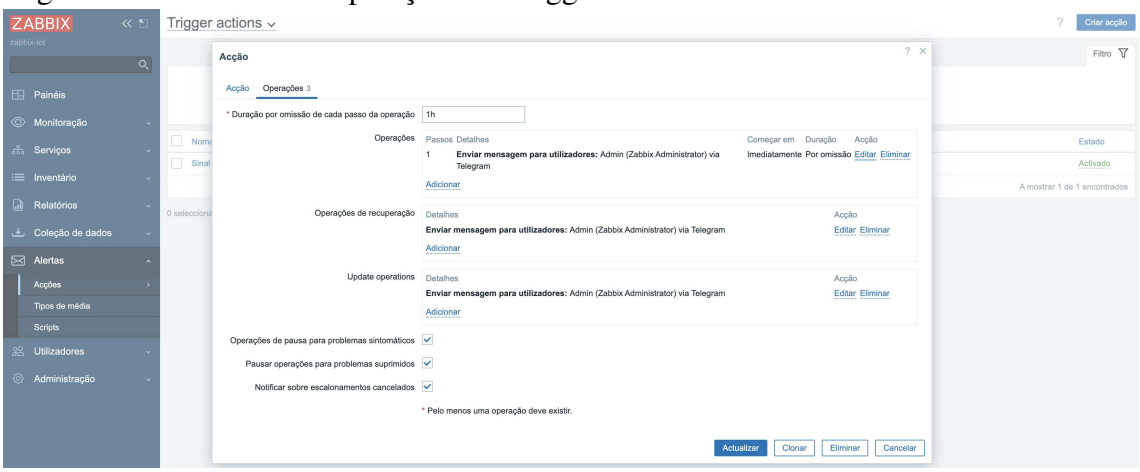
Figura 30 – Cadastro de Trigger Action no Zabbix



Fonte: Autoria Própria (2024)

Para o cadastro, se faz necessário selecionar a condição que vai gerar a ação, onde colocamos o *trigger* criado anteriormente. Na aba "Operações"selecionamos a ação de envio do alerta via Telegram, na criação, atualização e resolução do mesmo (Figura 31).

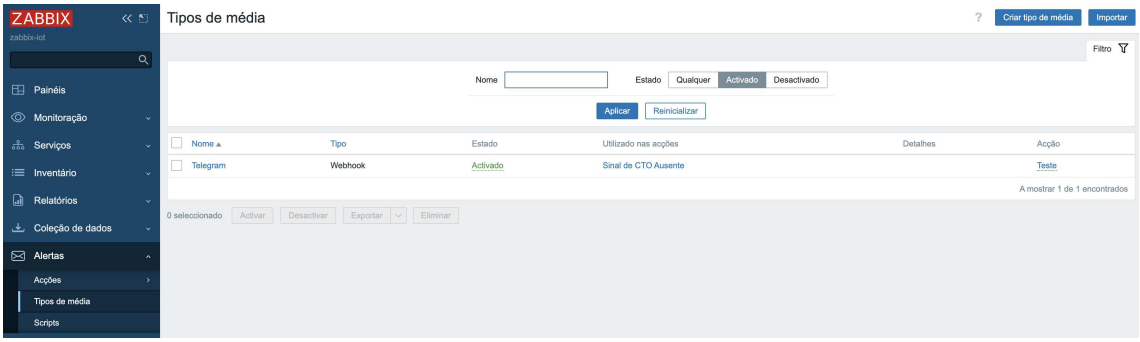
Figura 31 – Cadastro de operações no Trigger Action



Fonte: Autoria Própria (2024)

Para a realização do envio de mensagens ao Telegram, foi necessário configurar no Zabbix alguns parâmetros, localizados no menu *Alertas* > *Tipos de Mídia* > *Telegram*, como mostra a Figura 32.

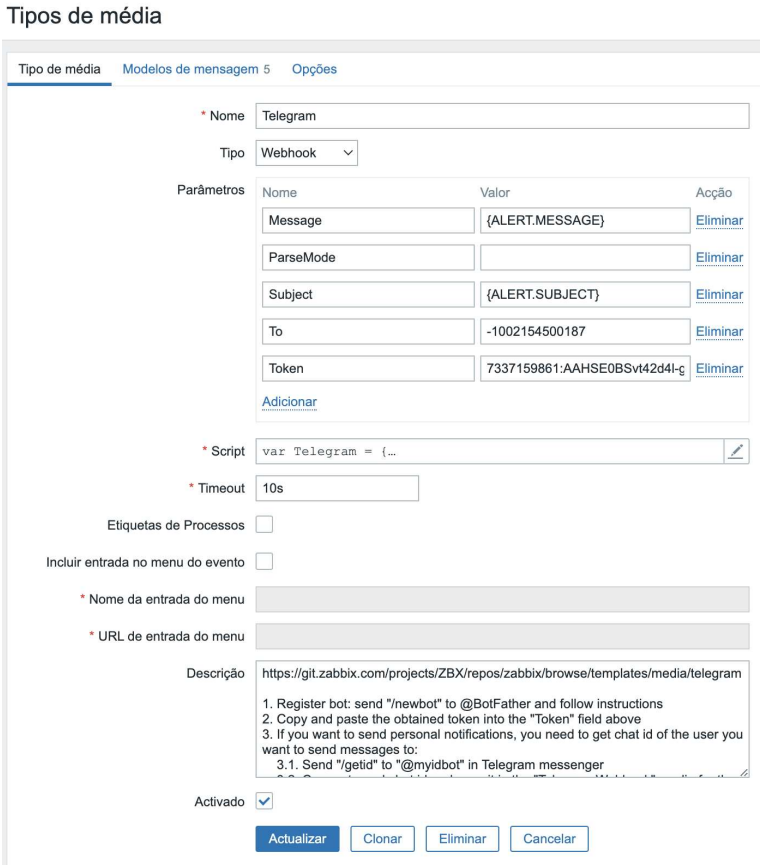
Figura 32 – Configuração do Telegram no Zabbix



Fonte: Autoria Própria (2024)

O *Token* é coletado no momento da criação de um *bot* no Telegram, e o campo *To* recebe um ID do grupo do mensageiro onde serão enviadas as mensagens geradas pelo Zabbix (Figura 33). Para a geração desses identificadores, foi utilizado o @*BotFather* no Telegram na criação do *bot*, posteriormente foi criado um grupo no mensageiro para adicionar o *bot* e os colaboradores do NOC responsáveis pelo monitoramento da rede. Também foi necessário configurar o ID do grupo no usuário administrador do Zabbix que enviará as mensagens.

Figura 33 – Configuração do Telegram no Zabbix



Fonte: Autoria Própria (2024)

Além disso, o Zabbix permite a integração via API com diversas plataformas, facilitando a automação de tarefas e interação com outros sistemas de gerenciamento. A versatilidade do Zabbix em lidar com diferentes protocolos de comunicação e dispositivos IoT o torna uma escolha robusta para projetos de monitoramento e gerenciamento de redes como a rede LoRaWAN.

7 RESULTADOS E DISCUSSÕES

7.1 Escolha dos componentes da solução

7.1.1 Tecnologia WSN

Para a implementação desta pesquisa, foi escolhido o tipo de rede LPWAN, pois os cenários encontrados das redes FTTH podem ter quilômetros de distância, tornando necessária a escolha de uma rede de longo alcance. Dentro destas redes, foi necessário considerar para a pesquisa, características como: eficiência energética, custos envolvidos e latência.

Considerando os custos envolvidos, as tecnologias LTE-M e NB-IoT são menos vantajosas, pois se utilizam de espectros licenciados para o seu funcionamento, operando em redes de operadoras móveis, demandando um maior custo de implantação e de manutenção (CARVALHO; MIERS, 2021). A tecnologia SigFox é proprietária, o que demanda uma assinatura para o seu uso que pode impactar no orçamento dos projetos. Assim, nesse aspecto as redes LoRaWAN se mostram mais vantajosas.

Com relação a latência, é necessária a escolha de uma tecnologia que consiga se comunicar em um menor tempo possível, para que o servidor de aplicação receba em tempo real a informação dos eventos ocorridos na rede FTTH. A tecnologia SigFox se mostra a menos vantajosa, proporcionando latência entre 1 e 30 segundos. NB-IoT oferece tempo de resposta entre 1,6 e 10 segundos. As opções LTE-M e LoRaWAN se mostram mais vantajosas nesse quesito, oferecendo latência entre 50 e 100 milissegundos e 61 à 371 milissegundos respectivamente (CARVALHO; MIERS, 2021).

O parâmetro de eficiência energética é fundamental para o cenário desta pesquisa, pois os sensores ficarão localizados dentro de CTOs da rede FTTH, que não possui alimentação elétrica. Nesse quesito, a tecnologia LTE-M mostra-se a menos vantajosa, com maior custo energético nas transmissões. NB-IoT e SigFox também demandam um maior uso energético, quando comparados com LoRaWAN, sendo este último o mais indicado para o cenário desta pesquisa, contribuindo para uma maior autonomia das baterias que alimentam os nós da rede (SINGH *et al.*, 2020).

A partir das análises realizadas, o protocolo LoRaWAN foi escolhido como melhor opção. Além de ser o mais vantajoso no requisito energético, também possui um tempo de resposta interessante e menores custos na implantação e manutenção da rede. Outro fator

importante das redes LoRaWAN é sua flexibilidade, pois não dependem de redes fechadas de operadoras, se mostrando uma opção mais interessante para adequação aos diferentes tipos e tamanhos projetos. Um resumo de algumas características pode ser visualizado na Figura 34.

Figura 34 – Características de Tecnologias LPWAN

Característica	SigFox	LoRaWAN	LTE-M	NB-IoT
Cobertura	< 12 Km	< 10 Km	< 10 Km	< 15 Km
Espectro	900 Mhz	900 Mhz	7 - 900 Mhz	8 - 900 Mhz
Taxa de Dados	100 à 600 bps	200 bps à 50 Kbps	< 1 Mbps	< 144 Kbps
Latência	1 à 30 s	61 à 371 ms	50 - 100 ms	1,6 à 10 s

Fonte: Adaptado de (CARVALHO; MIERS, 2021)

7.1.2 Node

Foi escolhido para o nó da rede o dispositivo Heltec WiFi LoRa 32 V2 na frequência LoRa de 915Mhz, que integra um microprocessador ESP32 e possui interface de rede LoRa com +20dBm de potência de transmissão e -139dBm de sensibilidade de recepção. Possui também interfaces Bluetooth e Wi-Fi b/g/n, além de funcionalidades como Analog to Digital Converter (ADC) e Digital to Analog Converter (DAC) (HELTEC, 2024). Essas características viabilizam e simplificam a comunicação na rede LoRa, pois esse dispositivo dispensa uso de interfaces extras para atuar nessa demanda. Além desses destaques, quando comparamos os recursos de processamento, memória e arquitetura, o ESP32 leva grande vantagem sobre outros dispositivos utilizados para esse uso, como o Arduíno Uno R3. Mais detalhes podem ser vistos na Figura 35.

Figura 35 – Características ESP32 x Arduino

Característica	Arduíno Uno R3	ESP 32
CPU	ATmega328	Xtensa LX6 DualCore
Clock	20 Mhz	240 Mhz
Arquitetura	8 bit	32 bit
RAM	2 KB SRAM	520 KB SRAM

Fonte: Adaptado de (PASTÓRIO *et al.*, 2021)

7.1.3 Gateway

Neste projeto foi utilizado o Gateway da Radioenge modelo RD43HATGPS, que opera na frequência permitida no Brasil e é homologado pela Anatel, possuindo atualmente fabricação nacional. Este equipamento possui uma vasta documentação e comunidade web que contribuem para a implantação do mesmo, assim como a integração com outras soluções. No aspecto alcance, o RD43HATGPS possui +27dBm de potência e uma antena externa de 6dBi, proporcionando uma larga cobertura de área, essencial para a viabilização desse projeto (RADIOENGE, 2024). Também é importante ressaltar que o mesmo foi projetado para funcionar acoplado à um Raspberry Pi 3, através de sua interface SPI, o que possibilita a instalação de um Servidor de Rede e um Servidor de Aplicação nesse mesmo dispositivo, caso seja necessário. Outro ponto importante, é a existência de um módulo GPS integrado ao Gateway, que facilita a gerência do mesmo, assim como a sincronização do relógio para maior precisão de tempo na comunicação com o Servidor de Rede. Na parte de comunicação lógica dos dados, esse Gateway utiliza o programa Packet Forwarder, que possui diversas vantagens como personalização e controle dos dados, pois se trata de um código aberto.

7.1.4 Network Server

A escolha da melhor solução para um servidor de rede LoRaWAN é uma etapa fundamental do projeto, influenciando diretamente em parâmetros como escalabilidade, segurança, confiabilidade e eficiência do negócio. Destacam-se como opções no mercado, as plataformas The Things Network (TTN) e a ChirpStack, pois ambas oferecem funcionalidades para implantação e gerenciamento dos dispositivos LoRa.

Com relação à escalabilidade, faz-se necessária a escolha de um servidor de rede que atenda cenários com tamanhos distintos, facilitando possíveis expansões futuras do negócio. Nesse aspecto, a plataforma TTN disponibiliza em sua versão gratuita um limite de até 10 Gateways e 10 nodes cadastrados na rede, o que possibilita a realização de testes e validações, mas se mostra ineficaz para cenários maiores. Existe ainda a possibilidade de contratação do plano Plus dessa plataforma, que aumenta o limite para 1.000 dispositivos, sendo necessário aumentar o plano em caso de uma demanda maior. Já a plataforma ChirpStack, por ser uma opção open-source, mostra-se uma solução adaptável aos mais distintos tamanhos de projetos. Os limites de quantidade de dispositivos gerenciados por essa plataforma, serão proporcionais à quantidade de

recursos disponibilizados no servidor, podendo ser realizados aumentos destes recursos à medida que o projeto demande a gerência de mais dispositivos (TTN, 2024), (CHIRPSTACK, 2024).

Quanto à segurança e confiabilidade dos dados, a versão gratuita da plataforma TTN mostra-se bem limitada, pois a rede LoRaWAN gerenciada pela mesma torna-se automaticamente pública, elevando os riscos de acesso aos dispositivos e dados trafegados pelos mesmos, além de oferecer poucas opções de protocolos de segurança, mesmo em sua versão mais completa. Na plataforma ChirpStack, é possível realizar a implementação de diversos protocolos de segurança, tanto relacionados à comunicação da rede, quanto nos acessos ao servidor, pois a ferramenta possui código aberto para livre personalização. Também é importante destacar a transparência existente na comunicação dos processos e fluxo de dados no ChirpStack, possibilitando análises e rastreamento dos dados de forma mais precisa (TTN, 2024), (CHIRPSTACK, 2024).

Considerando o parâmetro eficiência da solução, faz-se necessário salientar que esse projeto visa atender cenários que podem ter diferentes níveis de exigências relacionadas à desempenho, frequência de comunicação entre Gateway e nós, tempo de coleta dos dados, entre outros. Nesse ponto, a plataforma TTN possui uma limitação considerável na sua versão gratuita, por não garantir disponibilidade do servidor, tornando-se necessário o pagamento do plano Plus para que haja 99,9% de disponibilidade garantida. Com relação ao servidor ChirpStack, existe a possibilidade do aumento personalizado de recursos como memória e processamento do servidor, além de melhorias no ambiente, como implementação de redundância de link e servidor backup, elementos que contribuem diretamente para a eficiência da solução (TTN, 2024), (CHIRPSTACK, 2024).

7.1.5 Application Server

O Servidor de Aplicação é o componente que se integra ao Servidor de Rede para a coleta de dados, com a finalidade de armazenamento, processamento e apresentação dos mesmos. A escolha de uma solução para desempenhar esse papel está diretamente relacionada aos requisitos específicos de cada projeto. Nesse contexto, as opções consideradas para atender as demandas desta pesquisa incluem as plataformas Ubidots, TagoIO e Zabbix.

Considerando a necessidade de adaptação a diferentes cenários e redes, é fundamental que o servidor de aplicação seja altamente flexível. Neste aspecto, o Zabbix destaca-se como a melhor opção devido à sua natureza open-source, permitindo a realização de configurações para representar diversas realidades e integrar-se a aplicações existentes conforme necessário.

As plataformas Ubidots e TagoIO não são abertas, logo apresentam limitações de personalização, principalmente em suas versões gratuitas (TAGOIO, 2024), (UBIDOTS, 2024), (ZABBIX, 2024).

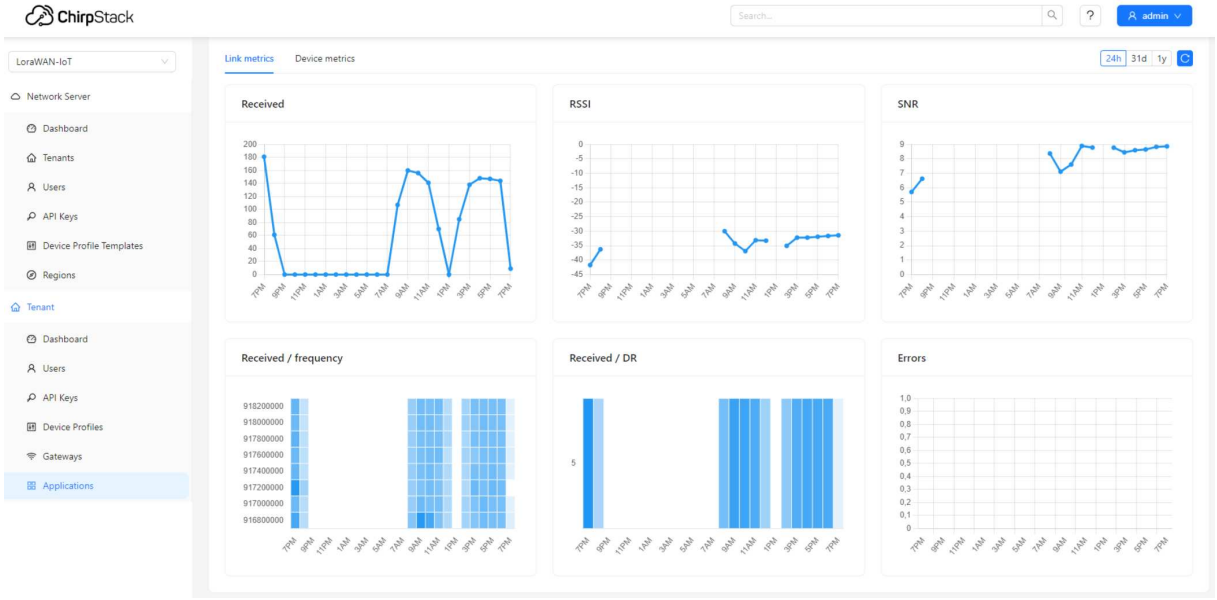
Outro ponto importante do projeto é a escalabilidade, pois a solução proposta deve conseguir suportar a gerência de dados em redes de tamanhos distintos, possibilitando o crescimento na quantidade de nós, Gateways e das informações gerenciadas. Considerando as aplicações propostas, o Zabbix também se sobressai neste aspecto, pois seus limites de armazenamento e processamento, são proporcionais à quantidade de recursos disponibilizados pelo servidor em que o mesmo está hospedado, devendo este, ser proporcional à cada cenário. Já as outras duas aplicações possuem limites específicos, como quantidade de dispositivos, quantidade de dados coletados e frequência de coleta, demandando a aquisição de pacotes de serviço mais completos para atender as necessidades (TAGOIO, 2024), (UBIDOTS, 2024), (ZABBIX, 2024).

Quanto à segurança dos dados, as aplicações Ubidots e TagoIO oferecem medidas de segurança padrão, exigindo a aquisição de planos mais completos para acesso à mais recursos, como duplo fator de autenticação para os usuários. Também é importante ressaltar que ambas as plataformas mantêm os dados armazenados em nuvem, acessíveis da internet, que oferece um risco maior por conta da exposição à ataques provenientes desta. Já a aplicação Zabbix possibilita a personalização de ferramentas de segurança, podendo ser implementados processos específicos para cada cenário, além de funcionar na rede local, reduzindo os riscos de ataques oriundos da internet. Além disso, o Zabbix apresenta total transparência devido à sua natureza open-source, permitindo uma análise completa de seus processos, ao contrário das outras plataformas, que possuem processos fechados, impossibilitando o rastreamento dos dados tratados (TAGOIO, 2024), (UBIDOTS, 2024), (ZABBIX, 2024).

7.2 Dashboards gerenciais

Visando o gerenciamento das informações e indicadores do sistema desenvolvido, foram criados *dashboards* gerenciais, contendo gráficos de acompanhamento dos dados. O primeiro *dashboard* foi criado no servidor de rede Chirpstack, como mostra a Figura 36.

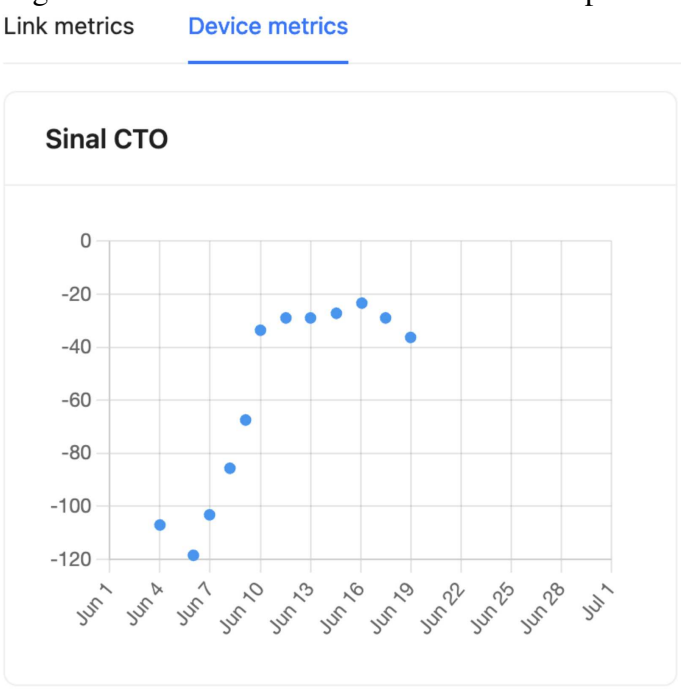
Figura 36 – Dashboard gerencial no Chirpstack



Fonte: Autoria própria (2024).

Nos gráficos do Chirpstack, é possível visualizar dados da rede LoRa, como a quantidade de dados recebidos no gráfico *received* e o histórico de sinal de comunicação entre o nó e o Gateway, no gráfico RSSI. Essas informações já permitem um nível de gerenciamento e acompanhamento da rede, possibilitando inclusive o acompanhamento do sinal medido da CTO, através do gráfico exibido na Figura 37.

Figura 37 – Gráfico de sinal da CTO no Chirpstack



Fonte: Autoria própria (2024).

Embora os gráficos do Chirpstack mostrem algumas informações sobre a rede, os mesmos possuem grande limitação com relação à manipulação e personalização, problemas resolvidos pelo monitoramento com o Zabbix. Para a implementação dos *dashboards* no Zabbix, foi necessário um tratamento inicial dos dados brutos coletados via protocolo MQTT. A Figura 38 mostra uma mensagem recebida no Zabbix com os dados coletados, na qual é possível observar e identificar a presença das informações desejadas.

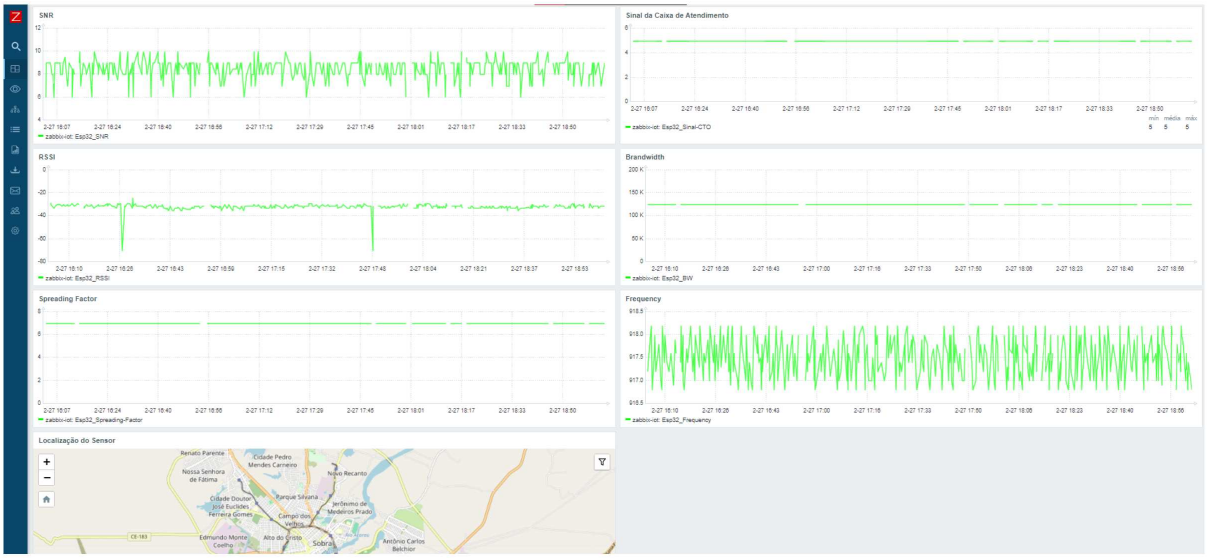
Figura 38 – Mensagem MQTT recebida no Zabbix

Timestamp	Valor
18-04-2024 17:07:48	{"deduplicationId":"11cc10d5-60dc-488c-9076-153911f56c35","time":"2024-04-18T20:08:09.862+00:00","deviceInfo":{"tenantId":"e4cea21b-3a8b-410e-8c7c-ba884458c4c3","tenantName":"LoraWAN-IoT","applicationId":"97180297-f562-41ec-b288-19ac856956c6","applicationName":"Monitoramento de CTO","deviceProfileId":"0b9a2f96-f6d0-4b5c-9722-5a58f6f4c701","deviceProfileName":"OTAA3","deviceName":"OTAA","devEui":"70b3d57ed0064d17","deviceClassEnabled":"CLASS_A","tags":{}},"devAddr":"260d7932","adr":true,"dr":5,"fCnt":1128,"fPort":1,"confirmed":false,"data":"AQLseA==","object":{"analogInput":{"1":-50.0}},"rxInfo":[{"gatewayId":"b827ebfffe13673c","uplinkId":37551,"nsTime":"2024-04-18T20:07:48.131732116+00:00","timeSinceGpsEpoch":"1397506107.862s","rssi":-91,"snr":8.0,"channel":6,"rfChain":1,"location":{"latitude":-3.70596,"longitude":-40.34564,"altitude":58.0},"context":"qTs0BA==","metadata":{"region_config_id":"au915_1","region_common_name":"AU915"},"crcStatus":"CRC_OK"}],"txInfo":{"frequency":918000000,"modulation":{"lora":{"bandwidth":125000,"spreadingFactor":7,"codeRate":"CR_4_5"}}}}

Fonte: Autoria própria (2024).

Após o tratamento e separação dos dados importantes, é possível realizar a implementação de gráficos personalizados no Zabbix, onde o usuário define os períodos dos gráficos, cores, formatos, limites e outras características. A Figura 39 mostra um *dashboard* criado com os principais indicadores da rede, assim como a informação do sinal da CTO da rede FTTH.

Figura 39 – Dashboard gerencial no Zabbix



Fonte: Autoria própria (2024).

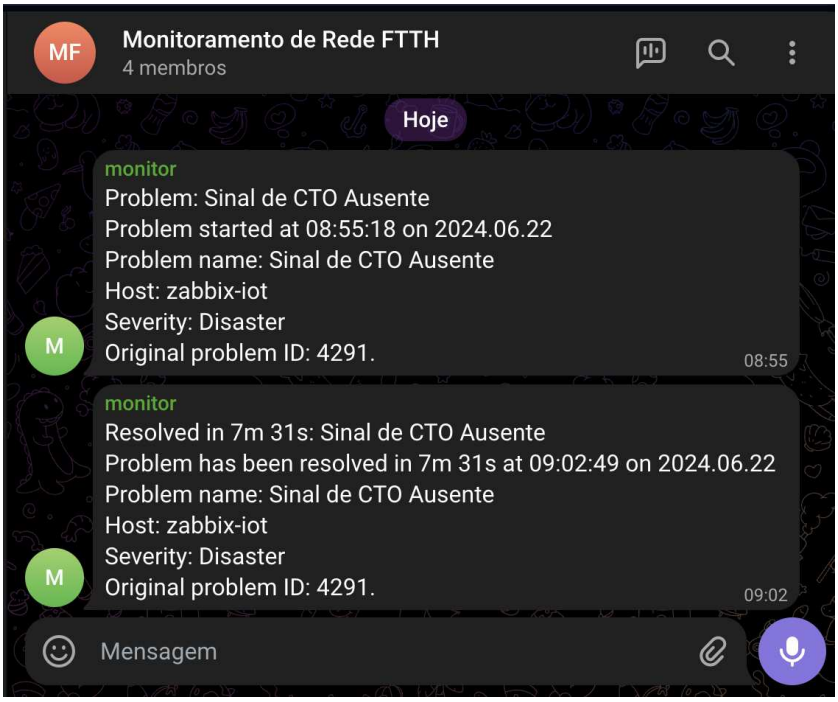
Com esse gerenciamento, já é possível acompanhar o funcionamento da rede LoRa, assim como o monitoramento do sinal da CTO. Também é possível estabelecer gatilhos ou *triggers*, que podem, por exemplo, alterar a cor do gráfico no caso de uma medição fora do limite estabelecido de sinal. A partir dos gráficos, também é possível acompanhar um histórico

conforme período estabelecido pelo usuário, podendo ser utilizado para acompanhamento de eventos da rede e também para comparação de informações no caso de uma manutenção na rede. Também é importante salientar que o monitoramento do sinal da CTO pode proporcionar ao ISP uma atuação em casos onde o problema está iniciando e ainda não afeta conexão de clientes, por exemplo quando o sinal foi ateunado em determinado ponto, mas ainda está dentro do limite de comunicação do equipamento do cliente. Nesses casos, o provedor de internet consegue detectar pelos gráficos, agendar uma manutenção programada naquele ponto, comunicar ao cliente com antecedência e atuar em horários de menor impacto para os usuários.

7.3 Alertas no Telegram

O servidor de aplicação Zabbix também possibilita a integração com inúmeros aplicativos de mensagens, como SMS, Whatsapp e Telegram. Essa comunicação pode ser utilizada para envio automático de alertas. Nesta dissertação foi implementado um serviço de envio automático de alertas pelo Zabbix para o Telegram. No caso de algum evento na rede, ou alguma alteração que faça o indicador 'sinal da CTO' sair do limite pré-estabelecido, o Zabbix automaticamente envia uma mensagem personalizada para um grupo do Telegram, como pode ser visualizado na Figura 40.

Figura 40 – Alertas no Telegram



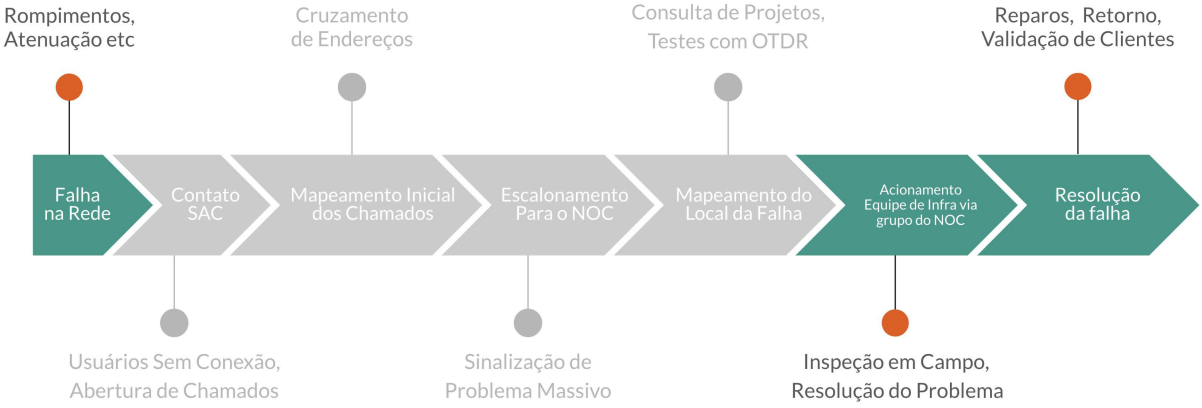
Fonte: Autoria própria (2024).

O grupo do Telegram pode ser criado para que os responsáveis pelo monitoramento da rede estejam nele, fazendo com que os eventos da rede sejam comunicados em tempo real para o NOC. As mensagens são personalizadas, podendo conter dados como o ID da CTO, localização, tempo de início do evento, severidade entre outros. O Zabbix também envia uma mensagem automática para o Telegram quando o evento é resolvido, informando o tempo total de indisponibilidade.

7.4 Melhoria no processo de gerenciamento de incidentes na rede FTTH

Conforme visto anteriormente, o cenário atual de gerenciamento de incidentes na rede FTTH pelos ISPs consiste em sete etapas (Figura 41). No caso de ocorrência de algum evento na rede que afete a conexão de usuários, o cenário atual depende do contato ativo do usuário para reportar a falta de conexão, assim como de processos de triagem e classificação de incidentes pelo setor do Serviço de Atendimento ao Consumidor (SAC) e do NOC do ISP.

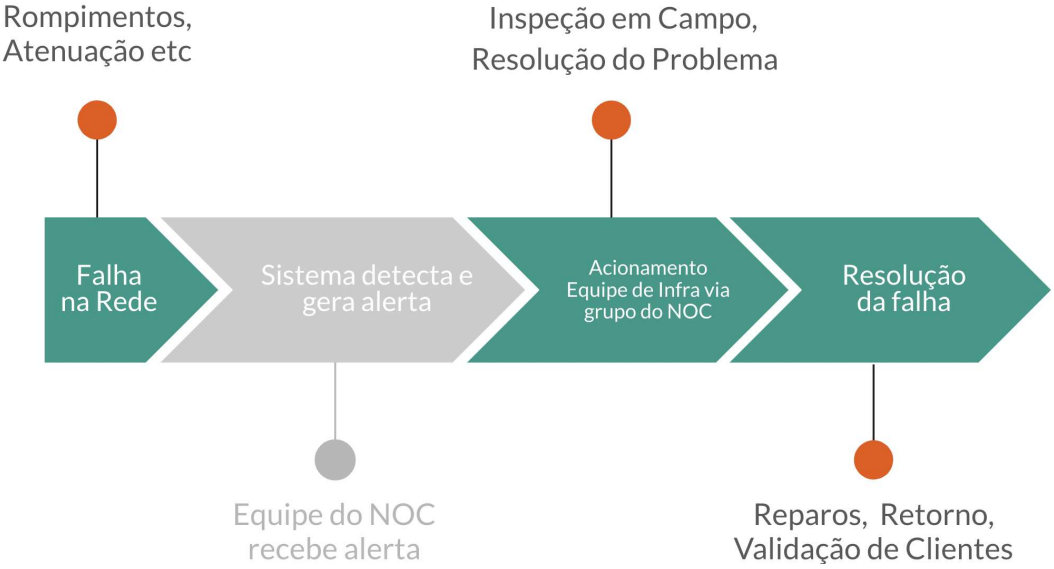
Figura 41 – Cenário atual de gerenciamento de incidentes de redes FTTH no ISP



Fonte: Autoria própria (2024).

Com a implementação da solução proposta, o ISP contará com um novo cenário, no qual as etapas de recepção, registro e triagem dos eventos serão substituídos pela etapa de detecção e comunicação automática do sistema proposto, resultando na redução de sete para quatro etapas no processo. Um esboço do novo cenário é exibido na Figura 42.

Figura 42 – Novo cenário com a solução proposta



Fonte: Autoria própria (2024).

Com essa atualização do cenário, o ISP passa a ter proatividade na detecção e resolução dos problemas da rede FTTH, pois o mesmo ficará ciente dos eventos da rede em tempo real e conseguirá tomar uma ação de forma mais rápida e assertiva. Tendo o conhecimento dos eventos da rede, o provedor de internet pode melhorar processos internos, implementando por exemplo, mensagens automáticas de alertas na central telefônica, assim, quando o usuário fizer contato para informar a falta de conexão, ouvirá uma mensagem informando que o provedor já está ciente da situação e que há equipes solucionando o caso, com determinado prazo para restauração da sua conexão. Este processo atuará diretamente na redução de custos com ligações telefônicas e atendentes, pois o usuário ficando ciente da situação não precisará mais falar com um atendente e encerrará sua ligação mais cedo quando comparado com o cenário anterior.

8 CONCLUSÃO

O monitoramento das redes ópticas FTTH é um desafio para os ISPs, tornando-se essencial para a garantia de qualidade e desempenho das conexões exigidas atualmente. A ausência desse acompanhamento contribui com a redução do indicador de disponibilidade da rede, demandando um tempo considerável entre mapeamento e resolução de problemas ocorridos na mesma. Por tanto, o presente estudo apresentou a implementação de um sistema capaz de coletar informações de forma remota sobre a rede, contribuindo positivamente com seu gerenciamento, através de um servidor de aplicação.

O uso da tecnologia LoRa e do protocolo LoRaWAN possibilitou a montagem de uma infraestrutura de comunicação com um servidor de rede ChirpStack, servindo como base para transmissão de indicadores da rede óptica FTTH. Para que as informações fossem armazenadas e tratadas, utilizou-se a plataforma Zabbix, que facilita a criação de relatórios de sinais e variações, bem como a geração de alarmes em casos de eventos na rede. A partir da integração desses elementos foi possível estabelecer um sistema de fácil gerenciamento e análise dos dados, incluindo a coleta em tempo real dos indicadores que servem de fonte para tomadas de decisão pelo setor de NOC de um ISP , otimizando processos e melhorando a qualidade do serviço prestado pela empresa.

REFERÊNCIAS

ABDELLAOUI, Z.; DIEUDONNE, Y.; ALEYA, A. Design, implementation and evaluation of a fiber to the home (ftth) access network based on a giga passive optical network gpon. **Array**, Elsevier, v. 10, p. 100058, 2021.

AERNOUTS, M.; BERKVENS, R.; VLAENDEREN, K. V.; WEYN, M. Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas. **Data**, MDPI, v. 3, n. 2, p. 13, 2018.

ALGHAMDI, A. M.; KHAIRULLAH, E. F.; MOJAMED, M. M. A. Lorawan performance analysis for a water monitoring and leakage detection system in a housing complex. **Sensors**, MDPI, v. 22, n. 19, p. 7188, 2022.

ALIMI, K. O. A.; OUAHADA, K.; ABU-MAHFOUZ, A. M.; RIMER, S. A survey on the security of low power wide area networks: Threats, challenges, and potential solutions. **Sensors**, MDPI, v. 20, n. 20, p. 5800, 2020.

ALIPIO, M.; BURES, M. Current testing and performance evaluation methodologies of lora and lorawan in iot applications: Classification, issues, and future directives. **Internet of Things**, Elsevier, p. 101053, 2023.

ALMUHAYA, M. A.; JABBAR, W. A.; SULAIMAN, N.; ABDULMALEK, S. A survey on lorawan technology: Recent trends, opportunities, simulation tools and future directions. **Electronics**, MDPI, v. 11, n. 1, p. 164, 2022.

ALVES, C.; FILHO, J. da S. Um estudo sobre o modelo zigbee de rede sem fio ieee 802.15. 4. **Revista Eletrônica de Iniciação Científica em Computação**, v. 18, n. 2, 2020.

AUGUSTIN, A.; YI, J.; CLAUSEN, T.; TOWNSLEY, W. M. A study of lora: Long range & low power networks for the internet of things. **Sensors**, MDPI, v. 16, n. 9, p. 1466, 2016.

BORKAR, S. R. Long-term evolution for machines (lte-m). In: **LPWAN technologies for IoT and M2M applications**. [S.l.]: Elsevier, 2020. p. 145–166.

CARVALHO, D. F. de; MIERS, C. C. Uma proposta de estudo comparativo de nb-iot vs lorawan para aplicaçao em redes iiot privadas para automaçao e monitoramento de processos. In: SBC. **Escola Regional de Redes de Computadores (ERRC)**. [S.l.], 2021. p. 19–24.

CENTENARO, M.; VANGELISTA, L.; ZANELLA, A.; ZORZI, M. Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios. **IEEE Wireless Communications**, v. 23, October 2016.

CHIRPSTACK. **Introduction - ChirpStack open-source LoRaWAN® Network Server documentation**. 2024. <https://www.chirpstack.io/docs/>. (Accessed on 01/12/2024).

ENRIKO, I. K. A.; GUSTIYANA, F. N.; GIRI, G. C. Lora gateway coverage and capacity analysis for supporting monitoring passive infrastructure fiber optic in urban area. **Elinvo (Electronics, Informatics, and Vocational Education)**, v. 8, n. 2, p. 164–170, 2023.

ERTÜRK, M. A.; AYDIN, M. A.; BÜYÜKAKKAŞLAR, M. T.; EVIRGEN, H. A survey on lorawan architecture, protocol and technologies. **Future internet**, MDPI, v. 11, n. 10, p. 216, 2019.

FARRELL, S. **Low-power wide area network (lpwan) overview**. [S.l.], 2018.

FILGUEIRAS, G. G.; PESSOA, C. R. M. Ftth em redes opticas passivas. **Engenharias On-line**, v. 1, n. 2, p. 34–42, 2015.

GEORGIOU, O.; RAZA, U. Low power wide area network analysis: Can lora scale? **IEEE Wireless Communications Letters**, IEEE, v. 6, n. 2, p. 162–165, 2017.

GU, F.; NIU, J.; JIANG, L.; LIU, X.; ATIQUZZAMAN, M. Survey of the low power wide area network technologies. **Journal of Network and Computer Applications**, Elsevier, v. 149, p. 102459, 2020.

HAMMADI, Y. I. Fiber bragg grating-based monitoring system for fiber to the home (ftth) passive optical network. **Journal of Optical Communications**, De Gruyter, v. 43, n. 4, p. 573–583, 2022.

HAQUE, K. F.; ABDELGAWAD, A.; YELAMARTHI, K. Comprehensive performance analysis of zigbee communication: an experimental approach with xbee s2c module. **Sensors**, MDPI, v. 22, n. 9, p. 3245, 2022.

HAXHIBEQIRI, J.; POORTER, E. D.; MOERMAN, n.; HOEBEKE, J. A survey of lorawan for iot: From technology to application. **Sensors**, Mdpi, v. 18, n. 11, p. 3995, 2018.

HELTEC. **WiFi Lora32.pdf**. 2024. <https://resource.heltec.cn/download/WiFi_LoRa_32/WiFi%20Lora32.pdf>. (Accessed on 02/24/2024).

HOSSAIN, M. I.; MARKENDAHL, J. I. Comparison of lpwan technologies: Cost structure and scalability. **Wireless Personal Communications**, Springer, v. 121, n. 1, p. 887–903, 2021.

ISMAIL, D.; RAHMAN, M.; SAIFULLAH, A. Low-power wide-area networks: opportunities, challenges, and directions. In: . [S.l.: s.n.], 2018. p. 1–6.

KARVONEN, H.; MIKHAYLOV, K.; ACHARYA, D.; RAHMAN, M. M. Performance evaluation of bluetooth low energy technology under interference. In: SPRINGER. **13th EAI International Conference on Body Area Networks 13**. [S.l.], 2020. p. 147–156.

LIMA, M. S.; NUNES, M. A. S. N. Estudo prospectivo relativo ao padrão zigbee para redes sem fio e sua utilização na computação em nuvens. **Scientia Plena**, v. 11, n. 1, 2015.

MATNI, N.; MORAES, J.; RIKER, A.; OLIVEIRA, H.; ROSÁRIO, D.; CERQUEIRA, E. Modelo de otimizaçao de alocaçao de recursos em lorawan para aplicaçoes de internet das coisas. In: SBC. **Anais do XXV Workshop de Gerência e Operação de Redes e Serviços**. [S.l.], 2020. p. 43–56.

MATZ, A. P.; FERNANDEZ-PRIETO, J.-A.; CAÑADA-BAGO, J.; BIRKEL, U. A systematic analysis of narrowband iot quality of service. **Sensors**, MDPI, v. 20, n. 6, p. 1636, 2020.

MEKKI, K.; BAJIC, E.; CHAXEL, F.; MEYER, F. A comparative study of lpwan technologies for large-scale iot deployment. **ICT express**, Elsevier, v. 5, n. 1, p. 1–7, 2019.

MILAROKOSTAS, C.; TSOLKAS, D.; PASSAS, N.; MERAKOS, L. A comprehensive study on lpwans with a focus on the potential of lora/lorawan systems. **IEEE Communications Surveys & Tutorials**, IEEE, v. 25, n. 1, p. 825–867, 2022.

NASCIMENTO, D. F. Sistema de medição de potência óptica lorawan. 2024.

NOREEN, U.; BOUNCEUR, A.; CLAVIER, L. A study of lora low power and wide area network technology. In: IEEE. **2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)**. [S.l.], 2017. p. 1–6.

ORTIZ, F. M.; CRUZ, P.; COUTO, R. d. S.; COSTA, L. H. M. Caracterização de uma rede sem-fio de baixa potência e longo alcance para internet das coisas. In: SBC. **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**. [S.l.], 2018. p. 1159–1172.

PASTÓRIO, A.; ROSSATO, J.; SÁ, J.; SPANHOL, F.; RODRIGUES, L.; CAMARGO, E. 2 fundamentos de lorawan–teoria e prática. **Sociedade Brasileira de Computação**, 2021.

PIZA, L. V.; ARCE, A. I. C.; TECH, A. R. B.; COSTA, E. J. X. Ensinando os fundamentos de redes de sensores sem fio usando um sistema simples. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 35, p. 01–07, 2013.

RADIOENGE. **Gateway LoRaWAN | Radioenge**. 2024. <https://www.radioenge.com.br/produto/gateway-lorawan/>. (Accessed on 02/22/2024).

RAHMAN, H. U.; AHMAD, M.; AHMAD, H.; HABIB, M. A. Lorawan: state of the art, challenges, protocols and research issues. In: IEEE. **2020 IEEE 23rd International Multitopic Conference (INMIC)**. [S.l.], 2020. p. 1–6.

RIDHO, C. S.; YUSUF, S. A. N. A.; ANDRA, D. N. S. S.; APRIONO, C. Perancangan jaringan fiber to the home (ftth) pada perumahan di daerah urban. **J. Nas. Tek. Elektro**, v. 3, 2020.

ROCHA, A. M. D.; OLIVEIRA, M. A. D.; FM, P. J.; CAVALHEIRO, G. G. H. Abp vs. otaa activation of lora devices: an experimental study in a rural context. In: IEEE. **2023 International Conference on Computing, Networking and Communications (ICNC)**. [S.l.], 2023. p. 630–634.

ROUTRAY, S. K.; MOHANTY, S. Narrowband iot: Principles, potentials, and applications. **International Journal of Hyperconnectivity and the Internet of Things (IJHIoT)**, IGI Global, v. 8, n. 1, p. 1–13, 2024.

SALAMI, I.; ADEWOLE, A. Comparative analysis of gpon and epon optical communication network. 2022.

SEMTECH. **LoRa Connect Transceiver, SX1276, 137MHz to 1020MHz | Semtech**. 2024. <https://www.semtech.com/products/wireless-rf/lora-connect/sx1276#documentation>. (Accessed on 01/24/2024).

SHI, G.; LI, K.; SHI, G.; LI, K. Fundamentals of zigbee and wifi. **Signal interference in WiFi and ZigBee networks**, Springer, p. 9–27, 2017.

SIGFOX. **Sigfox Support**. 2024. <https://support.sigfox.com/docs>. (Accessed on 01/22/2024).

SILVA, J. de C.; RODRIGUES, J. J.; ALBERTI, A. M.; SOLIC, P.; AQUINO, A. L. Lorawan—a low power wan protocol for internet of things: A review and opportunities. In: IEEE. **2017 2nd International multidisciplinary conference on computer and energy science (SpliTech)**. [S.l.], 2017. p. 1–6.

SINGH, R. K.; PULUCKUL, P. P.; BERKVENS, R.; WEYN, M. Energy consumption analysis of lpwan technologies and lifetime estimation for iot application. **Sensors**, MDPI, v. 20, n. 17, p. 4794, 2020.

SUGUMARAN, S.; LAKSHMI, D. N.; CHOUDHARY, S. An overview of ftth for optical network. **Advances in Smart Communication and Imaging Systems: Select Proceedings of MedCom 2020**, Springer, p. 41–51, 2021.

SUNDARAM, J. P. S.; DU, W.; ZHAO, Z. A survey on lora networking: Research problems, current solutions and open issues. **arXiv e-prints**, p. arXiv–1908, 2019.

TAGOIO. **IoT Cloud Platform | TagoIO**. 2024. <https://tago.io/>. (Accessed on 01/20/2024).

TANG, B.-h.; ZHOU, Z.-x. The design of communication network optical fiber cable condition monitoring system based on distributed optical fiber sensor. In: IEEE. **2018 International Conference on Electronics Technology (ICET)**. [S.l.], 2018. p. 97–101.

TTN. **Pricing Plans | The Things Stack**. 2024. <https://www.thethingsindustries.com/stack/plans/>. (Accessed on 01/12/2024).

UBIDOTS. **Ubidots — Powerful but simple Industrial IoT**. 2024. <https://ubidots.com/>. (Accessed on 01/20/2024).

WOOLLEY, M. The bluetooth low energy primer. **Bluetooth Blog**, v. 15, p. 2022, 2022.

ZABBIX. **Zabbix :: The Enterprise-Class Open Source Network Monitoring Solution**. 2024. <https://www.zabbix.com/>. (Accessed on 01/20/2024).

ZANAJ, E.; CASO, G.; NARDIS, L. D.; MOHAMMADPOUR, A.; ALAY, Ö.; BENEDETTO, M.-G. D. Energy efficiency in short and wide-area iot technologies—a survey. **Technologies**, MDPI, v. 9, n. 1, p. 22, 2021.

APÊNDICE A – ARQUIVO DE CONFIGURAÇÃO CHIRPSTACK.TOML

```
# Network related configuration.
[network]

  # Network identifier (NetID, 3 bytes) encoded as HEX (e.g. 010203).
  net_id="000000"

  # Enabled regions.
  # Multiple regions can be enabled simultaneously. Each region must
  # match the 'name' parameter of the region configuration in
  # '[[regions]]'.
  enabled_regions=[ "au915_0", "au915_1", ]

# API interface configuration.
[api]

  # interface:port to bind the API interface to.
  bind="0.0.0.0:8080"

  # Secret.
  secret=""

[integration]
  enabled=["mqtt"]

  [integration.mqtt]
    server="tcp://127.0.0.1:1883/"
    json=true
    username=""
    password=""
```

APÊNDICE B – ARQUIVO DE CONFIGURAÇÃO REGIONS_AU915_0.TOML

```
[[regions]]
  # ID is an use-defined identifier for this region.
  id="au915_0"

  # Description is a short description for this region.
  description="AU915 (channels 0-7 + 64)"

  # Common-name of this region as defined by the LoRa Alliance.
  common_name="AU915"

  # Gateway configuration.
  [regions.gateway]

    # Force gateways as private.
    force_gws_private=false

    # Gateway backend configuration.
    [regions.gateway.backend]

      # The enabled backend type.
      enabled="mqtt"

      # MQTT configuration.
      [regions.gateway.backend.mqtt]

        # Topic prefix.
        # The topic prefix can be used to define the region of the gateway.
        # Note, there is no need to add a trailing '/' to the prefix. The
        # trailing '/' is automatically added to prefix if configured.
        topic_prefix="au915_0"
```

```
# MQTT server (e.g. scheme://host:port where scheme is tcp, ssl
# or ws)
server="tcp://localhost:1883"

# Connect with the given username (optional)
username=""

# Connect with the given password (optional)
password=""

# Quality of service level
# 0: at most once. 1: at least once. 2: exactly once.
# Note: an increase of this value will decrease the performance.
qos=0

# Clean session
# Set the "clean session" flag in the connect message when this
# client connects to an MQTT broker. By setting this flag you
# are indicating that no messages saved by the broker for this
# client should be delivered
clean_session=false

# Client ID
# Set the client id to be used by this client when connecting to
# the MQTT broker. A client id must be no longer than 23
# characters. If left blank a random id will be generated by
# ChirpStack.
client_id=""

# Keep alive interval.
# This defines the maximum time that that should pass without
# communication between the client and server.
```

```
    keep_alive_interval="30s"

    # CA certificate file (optional)
    # Use this when setting up a secure connection (when server uses
    # ssl://) but the certificate used by the server is not trusted
    # by any CA certificate on the server (e.g. when self generated).
    ca_cert=""

    # TLS certificate file (optional)
    tls_cert=""

    # TLS key file (optional)
    tls_key=""

# Gateway channel configuration.
# Note: this configuration is only used in case the gateway is using
# the ChirpStack Concentratord daemon. In any other case, this
# configuration is ignored.
[[regions.gateway.channels]]
  frequency=915200000
  bandwidth=125000
  modulation="LORA"
  spreading_factors=[7, 8, 9, 10, 11, 12]

[[regions.gateway.channels]]
  frequency=915400000
  bandwidth=125000
  modulation="LORA"
  spreading_factors=[7, 8, 9, 10, 11, 12]

[[regions.gateway.channels]]
  frequency=915600000
```

```
  bandwidth=125000
  modulation="LORA"
  spreading_factors=[7, 8, 9, 10, 11, 12]

[[regions.gateway.channels]]
  frequency=915800000
  bandwidth=125000
  modulation="LORA"
  spreading_factors=[7, 8, 9, 10, 11, 12]

[[regions.gateway.channels]]
  frequency=916000000
  bandwidth=125000
  modulation="LORA"
  spreading_factors=[7, 8, 9, 10, 11, 12]

[[regions.gateway.channels]]
  frequency=916200000
  bandwidth=125000
  modulation="LORA"
  spreading_factors=[7, 8, 9, 10, 11, 12]

[[regions.gateway.channels]]
  frequency=916400000
  bandwidth=125000
  modulation="LORA"
  spreading_factors=[7, 8, 9, 10, 11, 12]

[[regions.gateway.channels]]
  frequency=916600000
  bandwidth=125000
  modulation="LORA"
```

```
    spreading_factors=[7, 8, 9, 10, 11, 12]

  [[regions.gateway.channels]]
    frequency=915900000
    bandwidth=500000
    modulation="LORA"
    spreading_factors=[8]

# Region specific network configuration.
[regions.network]

  # Installation margin (dB) used by the ADR engine.
  # A higher number means that the network-server will keep more
  # margin, resulting in a lower data-rate but decreasing the
  # chance that the device gets disconnected because it is unable
  # to reach one of the surrounded gateways.
  installation_margin=10

  # RX window (Class-A).
  # Set this to: 0: RX1 / RX2. 1: RX1 only. 2: RX2 only
  rx_window=0

  # RX1 delay (1 - 15 seconds).
  rx1_delay=1

  # RX1 data-rate offset
  rx1_dr_offset=0

  # RX2 data-rate
  rx2_dr=8

  # RX2 frequency (Hz)
```

```
rx2_frequency=923300000

# Prefer RX2 on RX1 data-rate less than.
# Prefer RX2 over RX1 based on the RX1 data-rate. When the RX1
# data-rate is smaller than the configured value, then the Network
# Server will first try to schedule the downlink for RX2, failing
# that (e.g. the gateway has already a payload scheduled at the RX2
# timing) it will try RX1.
rx2_prefer_on_rx1_dr_lt=0

# Prefer RX2 on link budget.
# When the link-budget is better for RX2 than for RX1, the Network
# Server will first try to schedule the downlink in RX2, failing
# that it will try RX1.
rx2_prefer_on_link_budget=false

# Downlink TX Power (in dBm EIRP)
# When set to -1, the downlink TX Power from the configured band will
# be used.
# Please consult the LoRaWAN Regional Parameters and local
# regulations for valid and legal options. Note that the configured
# TX Power must be supported by your gateway(s).
downlink_tx_power=-1

# ADR is disabled.
adr_disabled=false

# Minimum data-rate.
min_dr=0

# Maximum data-rate.
max_dr=5
```

```
# Enabled uplink channels.
# Use this when ony a sub-set of the by default enabled channels are
# being used. For example when only using the first 8 channels of
# the US band.
# Note: when left blank / empty array, all channels will be enabled.
enabled_uplink_channels=[0, 1, 2, 3, 4, 5, 6, 7, 64]

# Rejoin-request configuration (LoRaWAN 1.1)
[regions.network.rejoin_request]

  # Request devices to periodically send rejoin-requests.
  enabled=false

  # The device must send a rejoin-request type 0 at least every
  # 2^(max_count_n + 4) uplink messages. Valid values are 0 to 15.
  max_count_n=0

  # The device must send a rejoin-request type 0 at least every
  # 2^(max_time_n + 10) seconds. Valid values are 0 to 15.
  # 0  = roughly 17 minutes. 15 = about 1 year
  max_time_n=0

# Class-B configuration.
[regions.network.class_b]
# Ping-slot data-rate.
  ping_slot_dr=8
  # Ping-slot frequency (Hz)
  # set this to 0 to use the default frequency plan for the
  # configured region (which could be frequency hopping).
  ping_slot_frequency=0
```