

Luan Cerilio de Oliveira Lima

**Simulação Computacional e Avaliação  
Experimental da Tecnologia LoRa - Conceito de  
Rádio Definido por Software**

Santo André, SP

2022

Luan Cerilio de Oliveira Lima

# **Simulação Computacional e Avaliação Experimental da Tecnologia LoRa - Conceito de Rádio Definido por Software**

**Monografia** apresentada ao curso de Engenharia de Informação da Universidade Federal do ABC, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Informação.

Universidade Federal do ABC  
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas  
Curso de Graduação em Engenharia de Informação

Orientador: Prof. Dr. Ivan Roberto Santana Casella

Santo André, SP

2022

# Agradecimentos

Agradeço primeiramente a minha mãe que me mostrou que a educação é o melhor caminho. Ju, minha companheira. Júlia e Dan, meus irmãos. Aos meus amigos de UFABC e de Tamandutech, em especial ao Gabriel, Jones, Fafa, Joninhas, Vitão, Jorginho, Pupin, Thi, Vi, Mestre Jerônimo, André, Pedro e Marce pelo companheirismo no trajeto. Meus amigos além da UF, que estiveram presentes na minha vida em algum momento nessa caminhada: Fê, Johw, Decco, Luã, Dani Dani, Roro, Bit, Mu, Felipe, Ericksen, Saulo. Ao Professor Ivan pela orientação neste trabalho e aos docentes do curso de Engenharia de Informação, especialmente à equipe do projeto de Robótica Pedagógica que tive o prazer de participar, que me mostraram como a Universidade pode ir além da sala de aula. Obrigado!

# Resumo

A Internet das Coisas, do inglês *Internet of Things* (IoT), pode ser definida como uma rede de objetos físicos que possuem certa "inteligência" para processar e gerar dados, com habilidades de ver, ouvir e executar tarefas, fazendo com que estes dispositivos conversem, compartilhem informações e coordenem decisões. Este ecossistema engloba aplicações e segmentos diversos, como transportes, indústria, varejo, veículos, casas inteligentes, agricultura e saúde, gerando um mercado de US\$ 1 trilhão até 2024. Um dos sistemas de comunicação líderes em IoT é a tecnologia LoRa (do inglês *Long Range*), que representa 75% do mercado de soluções IoT, enquanto os 25% restantes ficam para aplicações utilizando a rede 5G.

Porém a tecnologia LoRa tem sua camada física patenteada. Isso dificulta o aprimoramento da tecnologia, faltando flexibilidade para testar novas formas de modulação, codificação, novos protocolos ou padrões. Uma maneira de contornar tais dificuldades é por meio do Rádio Definido por Software, que permite implementar em software componentes de um sistema de comunicação que são tipicamente implementados em hardware.

Desta maneira, visando o crescente mercado de IoT e a tecnologia líder de mercado, este projeto pretende estudar a tecnologia LoRa, simular sua camada física em face a diferentes modelos de canais de comunicação e avaliar experimentalmente um receptor LoRa *open source* implementado em um dispositivo de Rádio Definido por Software.

# Abstract

The Internet of Things (IoT) can be defined as a network of physical objects that has a certain "intelligence" to process and generate data, with the ability to see, hear and perform tasks, making these devices communicate to each other, share information and coordinate decisions. This ecosystem encompasses diverse applications and segments, such as transportation, industry, retail, vehicles, smart homes, agriculture and healthcare, generating a US\$1 trillion market by 2024. One of the leading IoT communication systems is LoRa technology, which represents 75% of the IoT solutions market, while the remaining 25% are for applications using the 5G network.

But LoRa technology has its physical layer patented. This makes it difficult to improve the technology, lacking flexibility to test new forms of modulation, encoding, new protocols or standards. One way to circumvent such difficulties is through Software Defined Radio, which allows the implementation in software of components of a communication system that are typically implemented in hardware.

Therefore, targeting the growing IoT market and the market-leading technology, this project aims to study the LoRa technology, simulate its physical layer against different models of communication channels and experimentally evaluate a LoRa open source receiver implemented in a Software Defined Radio device.

# Lista de ilustrações

Figura 1 – Principais tecnologias de comunicação sem fio para IoT classificadas em cobertura e taxa de dados, adaptado de (CHEN et al., 2017). . . . .	12
Figura 2 – Pilha de protocolos da tecnologia LoRa (SEMTECH, 2020b). . . . .	17
Figura 3 – Arquitetura de rede típica LoRaWAN. . . . .	18
Figura 4 – Esquema de funcionamento de um dispositivo classe A. . . . .	19
Figura 5 – Funcionamento de um dispositivo classe B. . . . .	20
Figura 6 – Funcionamento de um dispositivo classe C. . . . .	20
Figura 7 – Bit ADR no quadro MAC. . . . .	21
Figura 8 – Ilustração das chaves de autenticação em uma rede <i>LoRaWAN</i> . . . . .	22
Figura 9 – Mecanismo pseudo-aleatório de mudança de canal. . . . .	22
Figura 10 – <i>Dwell time</i> e <i>hop time</i> . . . . .	23
Figura 11 – Estrutura de um quadro LoRa. Adaptado de (SEMTECH, 2020a). . . . .	26
Figura 12 – Espectrograma de um <i>upchirp</i> linear. . . . .	30
Figura 13 – Diferentes <i>chirps</i> LoRa e seus deslocamentos iniciais em frequência (CROCE et al., 2020). . . . .	31
Figura 14 – Espectrograma de um pacote LoRa (ROBYNS. et al., 2018). . . . .	31
Figura 15 – Diagrama em blocos do processo de demodulação. Adaptado de (FIALHO; AZEVEDO, 2018). . . . .	33
Figura 16 – Diagrama em blocos da transmissão de um sinal LoRa. Adaptado de (AFISIADIS, 2020). . . . .	33
Figura 17 – Diagrama em blocos de um sistema de SDR genérico (WYGLINSKI MAZIAR NEKOVEE, 2009). . . . .	36
Figura 18 – Fluxograma do sistema LoRa simulado. . . . .	39
Figura 19 – Representação de um modelo de canal Rayleigh. Adaptado de (MATLAB, 2022). . . . .	41
Figura 20 – Representação de um modelo de canal Rice. Adaptado de (MathWorks, 2022). . . . .	41
Figura 21 – Curva BER $\times$ SNR (dB) do sistema LoRa com SF=7 em um canal com desvanecimento plano lento do tipo Rice, com coeficiente K variante, sem codificação de canal. . . . .	42
Figura 22 – Curva BER $\times$ SNR (dB) do sistema LoRa com SF=12 em um canal com desvanecimento plano lento do tipo Rice, com coeficiente K variante, sem codificação de canal. . . . .	43
Figura 23 – Conteúdo exibido no display OLED da placa de desenvolvimento Heltec WiFi LoRa 32 ao enviar um pacote LoRa. . . . .	45

Figura 24 – Ambiente de desenvolvimento Arduino IDE mostrando o cabeçalho do código. . . . .	46
Figura 25 – RTL-SDR utilizado na avaliação experimental. . . . .	47
Figura 26 – <i>Setup</i> experimental em ambiente interno. . . . .	48
Figura 27 – <i>Setup</i> experimental do transmissor. . . . .	49
Figura 28 – Captura de um símbolo LoRa com SF=12 e BW=7.8 kHz utilizando o <i>software</i> SDR#. . . . .	50
Figura 29 – Fluxo gráfico inicial de identificação do sinal LoRa em GNU Radio. . .	51
Figura 30 – Visualização do espectro do sinal utilizando GNU Radio. . . . .	52
Figura 31 – Fluxo gráfico com o bloco LoRa Receiver adicionado. . . . .	52
Figura 32 – Decodificação de um sinal LoRa na console do <i>software</i> GNU Radio. .	53
Figura 33 – Fluxo gráfico definitivo de recepção e decodificação de mensagens LoRa.	53
Figura 34 – Pacotes LoRa sendo visualizados no <i>software</i> Wireshark. . . . .	54
Figura 35 – Taxa de erros de pacotes recebidos pelo receptor SDR em função da configuração em ambiente interno a 12 cm entre transmissor e receptor e parâmetros de <i>firmware</i> conforme tabela 14. . . . .	55
Figura 36 – <i>Setup</i> experimental do receptor para ambiente externo. . . . .	56
Figura 37 – Pontos de transmissão do sinal. . . . .	57
Figura 38 – Visualização em 2D obtida pelo Google Maps do ponto mais distante. .	58
Figura 39 – Taxa de erro de pacotes recebidos pelo receptor SDR em função da configuração em ambiente externo conforme tabela 16. . . . .	59

# Lista de tabelas

Tabela 1	– Sumário dos parâmetros de <i>uplink</i> da especificação LoRaWAN para o plano de frequência AU915-928 (THE THINGS NETWORKS, 2021).	24
Tabela 2	– Sumário dos parâmetros de <i>downlink</i> da especificação LoRaWAN para o plano de frequência AU915-928 (THE THINGS NETWORKS, 2021).	24
Tabela 3	– Sumário dos parâmetros de <i>uplink</i> da especificação LoRaWAN para o plano de frequência EU863-870 (THE THINGS NETWORKS, 2021).	24
Tabela 4	– Sumário dos parâmetros de <i>downlink</i> da especificação LoRaWAN para o plano de frequência EU863-870 (THE THINGS NETWORKS, 2021).	25
Tabela 5	– Descrição do preâmbulo de um quadro LoRa.	27
Tabela 6	– Descrição do <i>header</i> de um quadro LoRa.	27
Tabela 7	– Principais parâmetros da camada física LoRa e seus possíveis valores.	28
Tabela 8	– Quantidade de símbolos e sensibilidade de recepção na banda 125 <i>kHz</i> do transceptor Semtech SX1276 por <i>SF</i> , adaptado de (SEMTECH, 2020a).	29
Tabela 9	– Parâmetros utilizados nas simulações do sistema LoRa.	40
Tabela 10	– Parâmetros ajustáveis no <i>firmware</i> do dispositivo.	46
Tabela 11	– Principais características do <i>dongle</i> RLT-SDR (RTL-SDR, 2013)	47
Tabela 12	– Parâmetros utilizados no transmissor ao capturar os pacotes LoRa com SDR#.	49
Tabela 13	– Parâmetros utilizados no transmissor ao capturar os pacotes LoRa com GNU Radio.	51
Tabela 14	– Parâmetros definidos no <i>firmware</i> do dispositivo na análise em ambiente <i>indoor</i> .	54
Tabela 15	– Distâncias de linha de visada de cada ponto medido.	58
Tabela 16	– Parâmetros utilizados no transmissor ao executar o experimento externamente.	59

# Lista de abreviaturas e siglas

AWGN	Additive white Gaussian noise
ABP	Activation-By-Personalisation
ADC	Analog to Digital Conversion
ANATEL	Agência Nacional de Telecomunicações
API	Application Programming Interface
BER	Bit Error Rate
BW	Bandwidth
CSS	Chirp Spread Spectrum
CR	Coding Rate
CRC	Cyclic Redundancy Check
DAC	Digital to Analog Conversion
DC	Direct Current
DDC	Digital Down-Conversion
DUC	Digital Up-Conversion
FIR	Finite impulse response
FSK	Frequency-shift keying
ISM	Industrial, Scientific and Medical
IoT	Internet of Things
IF	Intermediate Frequency
IP	Internet Protocol
GPS	Global Positioning System
LOS	Line of Sight
MAC	Media Access Control

RFID	Radio-Frequency Identification
LPWAN	Low Power Wide Area Network
SF	Spreading Factor
SDR	Software-Defined Radio
SNR	Signal-to-Noise Ratio
SOC	System on Chip
PER	Packet Error Rate
OTAA	Over-The-Air-Activation
OOK	On-off keying
VANET	Vehicular Ad Hoc Network
XOR	Exclusive Or

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	Objetivos	13
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	14
1.2	Metodologia	14
1.2.1	Levantamento Bibliográfico	14
1.2.2	Simulações Computacionais	14
1.2.3	Avaliação Experimental	15
1.3	Revisão Bibliográfica	15
1.4	Organização deste Trabalho	16
<b>2</b>	<b>A CAMADA DE ENLACE LORAWAN</b>	<b>17</b>
2.1	Arquitetura de Rede	17
2.2	Classes de Dispositivos	18
2.2.1	Classe A	19
2.2.2	Classe B	19
2.2.3	Classe C	20
2.3	Taxa de Dados Adaptativa	20
2.4	Segurança	21
2.5	Mecanismos de Transmissão	22
2.6	Regulamentação	23
<b>3</b>	<b>A CAMADA FÍSICA LORA</b>	<b>26</b>
3.1	Estrutura de um Quadro LoRa	26
3.1.1	Preâmbulo	27
3.1.2	Header	27
3.1.3	Payload	27
3.2	Parâmetros da Camada Física	27
3.2.1	Símbolo e <i>Spreading Factor</i>	28
3.2.2	Período de Símbolo, Taxas de Símbolo e de Bit	28
3.3	Modulação	29
3.3.1	Descrição Matemática da Modulação LoRa	31
3.3.2	Demodulador Ótimo	32
3.4	Cadeia de Transmissão Completa de um Transceptor LoRa	33
3.4.1	<i>Whitening</i>	33
3.4.2	Codificação Hamming	34

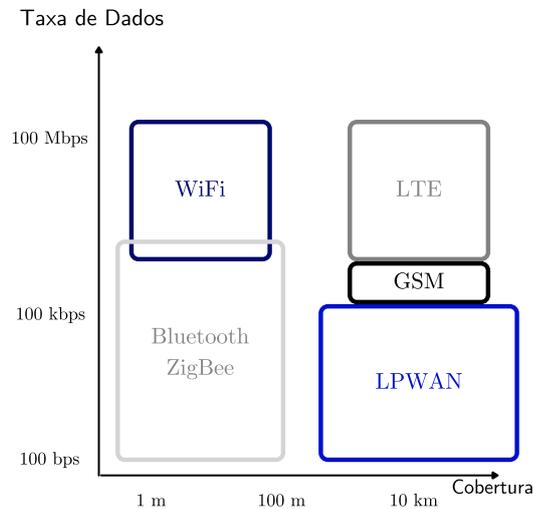
---

3.4.3	<i>Interleaving</i>	34
3.4.4	Mapeamento Gray	34
3.5	Taxa de Erro de Bit	34
<b>4</b>	<b>RÁDIO DEFINIDO POR SOFTWARE</b>	<b>36</b>
4.1	Principais Aplicações de SDR	37
4.1.1	Rádio Cognitivo	37
4.1.2	Redes de Comunicação Veiculares Dinâmicas	38
4.1.3	Comunicações por Satélite	38
4.1.4	Internet das Coisas	38
<b>5</b>	<b>SIMULAÇÕES DA CAMADA FÍSICA LORA</b>	<b>39</b>
5.1	Modelo do Sistema	39
5.1.1	Canal AWGN	40
5.1.2	Canal Rayleigh	40
5.1.3	Canal Rice	41
5.2	Desempenho do Sistema Simulado	41
<b>6</b>	<b>AVALIAÇÃO EXPERIMENTAL DA CAMADA FÍSICA LORA</b>	<b>45</b>
6.1	Hardware Utilizado	45
6.1.1	Heltec WiFi LoRa 32	45
6.1.2	RTL-SDR	46
6.2	Identificação de sinais LoRa utilizando SDR#	47
6.3	Identificação de sinais LoRa utilizando GNU Radio	50
6.4	Desempenho em Ambiente Interno	54
6.5	Desempenho em Ambiente Externo	56
<b>7</b>	<b>CONCLUSÃO</b>	<b>61</b>
	<b>REFERÊNCIAS</b>	<b>63</b>

# 1 Introdução

A Internet das Coisas, do inglês *Internet of Things* (IoT) é um termo que foi usado primeiramente por Kevin Ashton em 1999 durante uma apresentação na empresa Procter & Gamble (P&G). Na época, ele imaginava vincular a tecnologia de RFID (do inglês *Radio-frequency identification*) na cadeia de suprimentos da P&G à Internet, tópico em alta no momento (ASHTON, 2009). Anos após a primeira utilização do termo, novas tecnologias surgiram, permitindo que a ideia de IoT virasse realidade, podendo definir este conceito como uma rede de objetos físicos que possuem certa "inteligência" para processar e gerar dados, com habilidades de ver, ouvir e executar tarefas, fazendo com que estes dispositivos conversem, compartilhem informações e coordenem decisões.

Este ecossistema engloba aplicações e segmentos diversos, como transportes, indústria, varejo, veículos, casas inteligentes, agricultura e saúde (AL-FUQAHA et al., 2015). Para tornar reais aplicações tão distintas, existem diversas tecnologias de comunicação sem fio, classificadas em padrões diferentes de acordo com sua área de cobertura e taxa de transmissão de dados (CHEN et al., 2017), exemplificadas na figura 1.



**Figura 1** – Principais tecnologias de comunicação sem fio para IoT classificadas em cobertura e taxa de dados, adaptado de (CHEN et al., 2017).

Dentre as principais tecnologias e padrões de comunicação, as Redes de Longo Alcance e Baixa Potência (do inglês *Low Power Wide Area Network*, LPWAN) surgiram recentemente como complemento aos padrões existentes. Enquanto tecnologias de curto alcance como Bluetooth ou ZigBee são desenhadas para cobrir pequenas áreas e tecnologias celulares são implementadas para prover alta taxa de dados e cobrir grandes áreas, LPWANs têm como intenção cobrir grandes áreas com o menor custo energético possível a uma menor taxa de dados (RAZA; KULKARNI; SOORIYABANDARA, 2017), sendo LoRa,

Sigfox NB-IoT as mais populares (MEKKI et al., 2018).

A tecnologia Sigfox emprega modulação D-BPSK (do inglês *Differential Binary Phase-Shift Keying*) de banda ultra-estreita, utiliza espectro de frequência não licenciado abaixo de 1 GHz e é operada globalmente pela empresa francesa de mesmo nome da tecnologia. NB-IoT é uma tecnologia de banda estreita padronizada pela organização 3GPP (*3rd Generation Partnership Project*), baseada em LTE (*Long-Term Evolution*) (ZAYAS; MERINO, 2017). LoRa emprega modulação CSS (do inglês *Chirp Spread Spectrum*), utiliza espectro de frequência não licenciado abaixo de 1 GHz, padronizada e complementada por uma camada de enlace denominada LoRaWAN (do inglês *Long Range Wide Area Network*), especificada e de código aberto, mantida pela *LoRa Alliance*. No contexto deste projeto, a tecnologia LoRa será estudada.

De acordo com a Global Data (GLOBAL DATA, 2021), o mercado global de IoT ultrapassará a marca de US\$ 1 trilhão até 2024 e movimentará mais de 30 bilhões de dólares na América Latina até 2023 (DINO, 2021). Segundo a LoRa Alliance, LoRa representa 75% do mercado de soluções IoT, enquanto os 25% restantes ficam para aplicações em 5G que demandam maior largura de banda e baixa latência (LORA ALLIANCE, 2021b).

Porém, a tecnologia LoRa tem sua camada física patenteada. Isso dificulta seu aprimoramento, faltando flexibilidade para testar novas formas de modulação, codificação, novos protocolos ou padrões. Uma maneira de contornar tais dificuldades é por meio do conceito de Rádio Definido por Software (do inglês *Software-Defined Radio*, SDR), que permite implementar em *software* componentes de um sistema de comunicação que são tipicamente implementados em *hardware*.

Desta maneira, visando o crescente mercado de IoT e a tecnologia líder de mercado, este projeto pretende estudar a tecnologia LoRa, simular sua camada física em face a diferentes modelos de canais de comunicação e avaliar experimentalmente um receptor LoRa *open source* implementado em um dispositivo SDR.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Analisar a camada física LoRa por meio de análise matemática e simulações numéricas, implementar em plataforma de SDR as capacidades básicas de um receptor LoRa e avaliar seu desempenho.

### 1.1.2 Objetivos Específicos

- Compreender a modulação LoRa e seus principais parâmetros por meio de análise matemática.
- Compreender a cadeia de transmissão completa LoRa por meio dos processos de engenharia reversa já publicados.
- Traçar a curva de taxa de erros de bit teórica de LoRa em MATLAB para diferentes *Spreading Factors*.
- Simular os componentes básicos da camada física LoRa em MATLAB em um canal AWGN (do inglês *Additive White Gaussian Noise*).
- Simular os componentes básicos da camada física LoRa em MATLAB em canais sujeitos a desvanecimento plano.
- Implementar em plataforma de SDR o receptor *open source* LoRa.
- Realizar medições do sistema físico e avaliar seu desempenho em ambientes internos e externos.

## 1.2 Metodologia

A metodologia utilizada pode ser separada em três blocos, como segue abaixo.

### 1.2.1 Levantamento Bibliográfico

Por meio de levantamento bibliográfico, pretende-se compreender a tecnologia LoRa como um todo, começando pela principal especificação relacionada à tecnologia, denominada LoRaWAN. Então será analisada a camada física LoRa, a composição de seu quadro, os principais parâmetros e outros processos que trazem robustez a cadeia de transmissão completa e identificados por meio de engenharia reversa já que a camada física é patenteada, como *whitening*, codificação de canal, *interleaving* e mapeamento *Gray*, além de seu desempenho teórico no que diz respeito a taxa de erros de bit.

### 1.2.2 Simulações Computacionais

Por meio de análises numéricas, pretende-se simular e comparar os resultados teóricos com os computacionais, utilizando o *software* MATLAB, implementando a camada física LoRa com seus componentes básicos, ou seja, gerando bits aleatórios que serão mapeados em código *Gray* e introduzidos na modulação *Chirp Spread Spectrum* em um canal AWGN. Então, pretende-se analisar este modelo de camada física em modelos de canais sujeitos a desvanecimento plano.

### 1.2.3 Avaliação Experimental

Por fim, a avaliação experimental será feita implementando em plataforma de SDR o receptor *open source* LoRa, testando seu alcance e a taxa de erros de pacote (PER, *Packet Error Rate*). Como ferramentas serão utilizados os dispositivos RTL-SDR (RTL-SDR, 2013), Heltec ESP32 LoRa (HELTEC, 2019) e os *softwares* Arduino, SDR Sharp e GNU Radio.

## 1.3 Revisão Bibliográfica

A seguir serão descritos os principais estudos relacionados a este trabalho.

**Decoding LoRa: Realizing a Modern LPWAN with SDR** (KNIGHT; SEEGER, 2016): Os autores Matthew Knight e Balint Seeger foram uns dos primeiros a realizar um receptor da camada física LoRa em uma plataforma de SDR utilizando o *toolkit* GNU Radio por meio de engenharia reversa, deixando-a disponível no GitHub. O estudo apresenta tanto os parâmetros necessários para a demodulação, quanto para decodificação. Porém, a ordem em que as operações de *dewhitening* e *deinterleaving* são realizadas diferem da apresentada na patente da tecnologia. Além disso, segundo (ROBYNS. et al., 2018), o decodificador possui limitações, decodificando apenas *frames* curtos transmitidos sem um cabeçalho.

**A Multi-Channel Software Decoder for the LoRa Modulation Scheme** (ROBYNS. et al., 2018): Neste trabalho, é proposta uma análise da camada física LoRa e um decodificador multicanais (simultâneo) baseado em SDR, utilizando o *framework* GNU Radio e disponibilizado publicamente. Os autores utilizaram três diferentes plataformas de SDR e três transceptores comerciais LoRa distintos, comprovando compatibilidade de recepção com dois transceptores para uma SNR (do inglês *Signal-to-Noise Ratio*) de pelo menos 20 dB, afim de atingir uma PER (do inglês *Packet Error Rate*) nula, independente da plataforma de SDR adotada. Um aspecto de destaque deste trabalho é a proposta de utilização do algoritmo Schmidl-Cox em diferentes versões para sincronização e detecção dos sinais e uma nova técnica de demodulação, que remove a necessidade de correção de erros em frequência.

**Towards an SDR implementation of LoRa: Reverse-engineering, demodulation strategies and assessment over Rayleigh channel** (MARQUET; MONTAVONT; PAPADOPOULOS, 2020): O artigo propõe realizar uma análise detalhada dos processos de engenharia reversa da camada física LoRa, além de apresentar as expressões matemáticas minuciosas referentes a cada estágio do transceptor, complementando o trabalho dos autores (KNIGHT; SEEGER, 2016) e (ROBYNS. et al., 2018) expostos anteriormente. Além disso, simulações do sistema são feitas tomando como parâmetro a SNR para diferentes técnicas de modulação (CSS, M-FSK, LoRa e FSK-LoRa) em quatro

canais distintos (AWGN, seletivos em frequência, seletivos no tempo e Rayleigh).

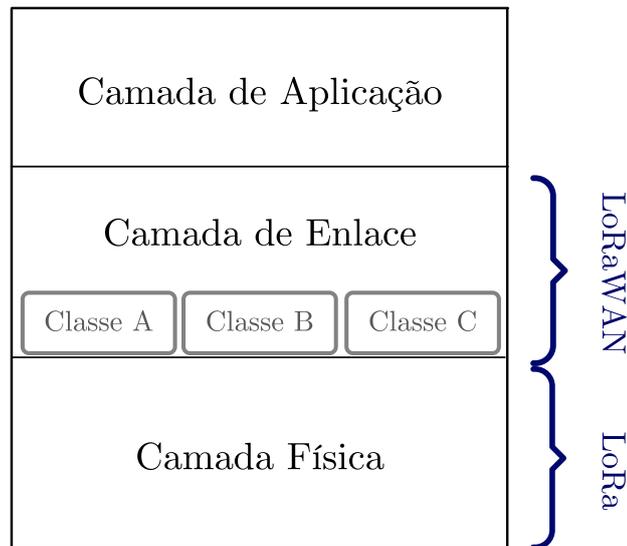
**An Open-Source LoRa Physical Layer Prototype on GNU Radio** (TAPPAREL et al., 2020): O protótipo proposto pelos autores é uma implementação em SDR completa, um transceptor compatível com a camada física LoRa, também utilizando o GNU Radio. A contribuição principal se dá no estudo, estimação e correção do desvio temporal de amostragem e do desvio de frequência da portadora. Os autores possibilitaram a operação para SNR menores e corrigiram a ordem de codificação e *whitening*, colaborando com o trabalho desenvolvido por (ROBYNS. et al., 2018) e (KNIGHT; SEEBER, 2016), respectivamente.

## 1.4 Organização deste Trabalho

A estrutura deste trabalho, deste capítulo em diante, é a descrita a seguir. No segundo capítulo, são apresentadas as principais características da especificação LoRaWAN, como a arquitetura de rede, classes de dispositivos, taxa de dados adaptativa, segurança, mecanismos de transmissão e regulamentação. No terceiro capítulo é apresentado o referencial teórico da camada física LoRa, tais como seus principais parâmetros e modulação. No quarto capítulo, é apresentado o referencial teórico do conceito de SDR, definindo seus aspectos técnicos e aplicações. Já no quinto capítulo, são apresentados os resultados das simulações e análises numéricas realizadas. Então, no sexto capítulo, são apresentados o receptor LoRa implementado em SDR e os resultados obtidos. Por fim, o sétimo capítulo é dedicado a conclusão e trabalhos futuros.

## 2 A Camada de Enlace LoRaWAN

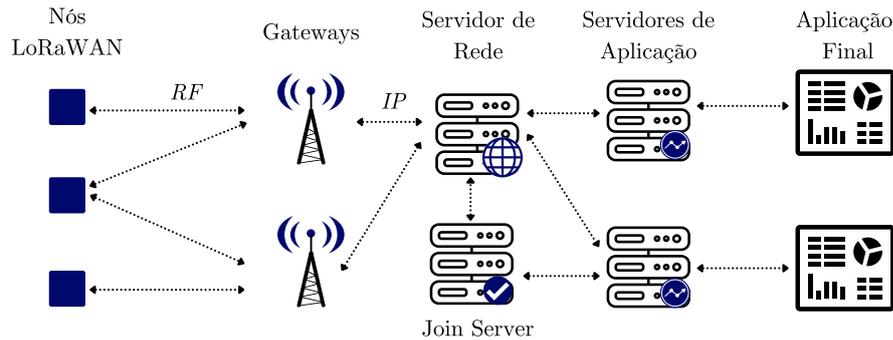
A especificação LoRaWAN é um padrão de LPWAN desenhado para conectar dispositivos sem fio operados a bateria, mantido pela organização *LoRa Alliance* ([LORA ALLIANCE, 2021a](#)). Como apresentado na figura 2, LoRaWAN define a camada de enlace, ou seja, o protocolo de comunicação e a arquitetura de rede, enquanto LoRa especifica a camada física definida para comunicação de longa distância.



**Figura 2** – Pilha de protocolos da tecnologia LoRa ([SEMTECH, 2020b](#)).

### 2.1 Arquitetura de Rede

Uma rede LoRaWAN é tipicamente implantada em uma topologia estrela-estrela, composta por *gateways* que retransmitem mensagens entre os dispositivos (também chamados de nós) e o Servidor de Rede central (*uplink*), que por sua vez garante a autenticidade dos nós e integridade das mensagens; Servidores de Aplicação que lidam com a interpretação dos dados dos sensores, segurança de rede e geração de *payload* para envio de pacotes do Servidor de Rede em direção ao dispositivo (*downlink*); *Join Server* que processa e responde os *frames* de *join-request* vindos dos nós e sinaliza ao Servidor de Rede qual Servidor de Aplicação o dispositivo deve se conectar ([SEMTECH, 2020b](#)). A figura 3 apresenta os principais componentes de uma rede LoRaWAN típica.



**Figura 3** – Arquitetura de rede típica LoRaWAN.

A comunicação entre o *gateway* e o dispositivo é bidirecional, ou seja, o dispositivo tanto envia como recebe informações do *gateway*. Além disso, o dispositivo realiza um *broadcast* de sua informação para todos os *gateways* próximos, fazendo com que seja possível que a mesma mensagem seja recebida por mais de um *gateway*. Os *gateways* atuam como uma ponte transparente entre o dispositivo e o Servidor de Rede, convertendo pacotes de Radiofrequência em IP (*Internet Protocol*) e vice versa. Por sua vez, o Servidor de Rede coleta as mensagens de todos os *gateways*, filtra os dados duplicados e determina qual *gateway* possui a melhor recepção. Então, o Servidor de Rede encaminha o pacote ao Servidor de Aplicação correto, onde o usuário final processará os dados do dispositivo. Por fim, o Servidor de Aplicação pode responder ao dispositivo e o Servidor de Rede determinará qual *gateway* utilizar para realizar o *broadcast* da informação em direção ao dispositivo (LORA ALLIANCE, 2020).

## 2.2 Classes de Dispositivos

Os dispositivos em uma rede LoRaWAN atendem diferentes aplicações com requisitos e perfis distintos, procurando o equilíbrio entre a latência na comunicação e a vida útil do dispositivo. Desta maneira, a especificação LoRaWAN define três classes de dispositivos: A, de menor consumo de energia e maior latência, onde o *uplink* é seguido de até duas janelas de recepção (*downlink*); B, também chamado de *beacon*, com as mesmas funcionalidades de A somadas à janelas extras de recepção; C, também com as mesmas funcionalidades de A, porém com janela de recepção contínua após o *uplink*, mais adequado para aplicações com atuadores e latência mínima. Desta maneira, um dispositivo LoRaWAN possui ao menos as funcionalidades da classe A necessariamente (LORA ALLIANCE, 2020). A seguir, serão detalhadas as características de cada uma dessas classes de dispositivos

### 2.2.1 Classe A

Dispositivos classe A passam a maior parte de seu tempo em modo de baixo consumo, com seu módulo de RF desligado (*sleep mode*). Quem inicia o processo de comunicação é o dispositivo, por meio de *uplink*. Podem ser programados para enviar um *uplink* de tempos em tempos ou ser ativados por um evento, como a mudança de um parâmetro no ambiente monitorado pelo sensor. Então, após o *uplink*, o dispositivo abre uma janela de recepção (RX1). Caso não receba nada em RX1, o dispositivo dorme por um breve período de tempo e abre uma segunda janela de recepção (RX2), finalizando seu ciclo (SEMTECH, 2020b), como exemplificado na figura 4.

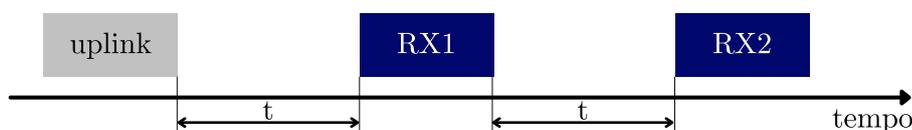
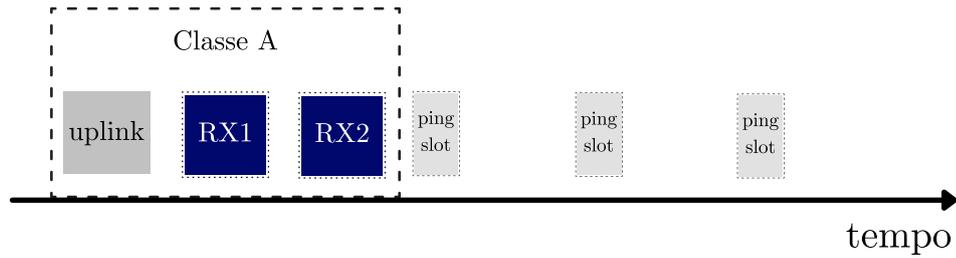


Figura 4 – Esquema de funcionamento de um dispositivo classe A.

### 2.2.2 Classe B

Além das janelas de recepção da classe A, dispositivos classe B possuem janelas extras de recepção regulares, exemplificadas na figura 5. Quem controla a classe do dispositivo é o Servidor de Aplicação, ou seja, por meio de *downlinks* é possível alterar o modo do dispositivo. Para que uma rede LoRaWAN suporte dispositivos classe B, todos os *gateways* na rede precisam ter embutido um GPS (do inglês *Global Positioning System*) como fonte de tempo, de maneira que estes estejam sincronizados no tempo. Então, os *gateways* na rede realizam o *broadcast* periódico de mensagens contendo informações de tempo e localização. Baseados nas informações enviadas pelos *gateways*, os dispositivos classe B abrem janelas de recepção periódicas, chamadas de *ping slots* (LORA ALLIANCE, 2020), de forma a alinhar seu *clock* interno com a referência da rede. Qualquer *ping slot* pode ser utilizado pela rede para iniciar um *downlink*. Quando operando no modo classe B, os dispositivos devem seguir três parâmetros para o *ping slot* de acordo com a região em questão (LORA ALLIANCE, 2021c), sendo eles a periodicidade, a taxa de dados e o canal.

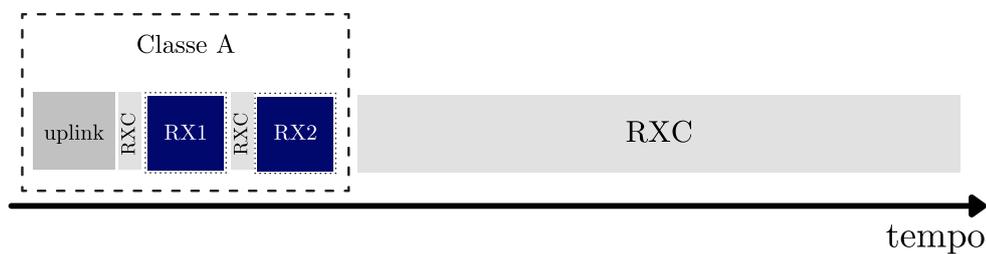
Os *frames* de *downlink* podem ser encaminhados pela rede como *unicast* ou *multicast*. No caso de *unicast*, o *downlink* é enviado a um dispositivo específico, enquanto que no *multicast*, os *frames* são enviados para um grupo de dispositivos, identificados por meio de um endereço de *multicast* e chaves de criptografia (LORA ALLIANCE, 2020).



**Figura 5** – Funcionamento de um dispositivo classe B.

### 2.2.3 Classe C

Já os dispositivos que operam em classe C são utilizados geralmente em aplicações em que a fonte de energia disponível não é limitada. Deste modo, adicionalmente às capacidades da classe A, há janelas de recepção contínuas, tanto para *unicast* quanto para *multicast*. Além das janelas de recepção RX1 e RX2 da classe A, há uma janela de recepção denominada RXC. O dispositivo não estará disponível para *downlink* em RXC apenas quando estiver realizando *uplink* ou *downlink* em RX1 ou RX2. Então, existem janelas de recepção RXC mesmo entre o *uplink* e RX1 e entre RX1 e RX2. Após RX2, o dispositivo escutará em RXC continuamente (LORA ALLIANCE, 2020), como mostra a figura 6.

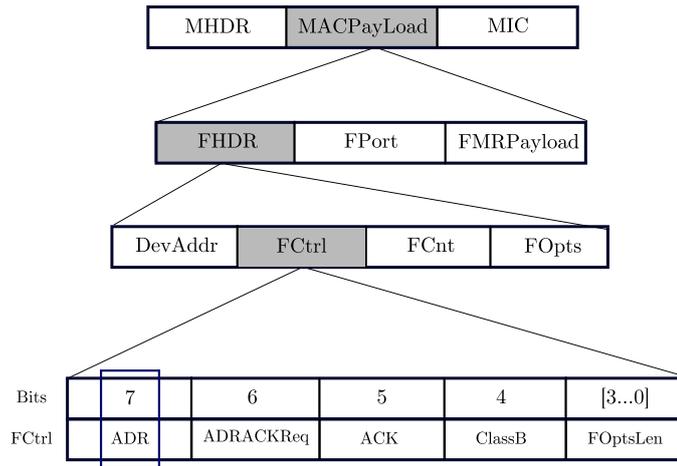


**Figura 6** – Funcionamento de um dispositivo classe C.

## 2.3 Taxa de Dados Adaptativa

O mecanismo de taxa de dados adaptativa empregado na especificação LoRaWAN possibilita que dispositivos otimizem a comunicação com a rede, utilizando a maior taxa de dados e a menor potência de transmissão possível.

O próprio dispositivo define se habilitará, ou não, o controle da taxa de dados adaptativa durante o *uplink*, por meio do sétimo bit no octeto de controle de quadro FCtrl do quadro MAC, denominado ADR, apresentado em detalhes na especificação LoRaWAN (LORA ALLIANCE, 2020), como ilustrado na figura 7.



**Figura 7** – Bit ADR no quadro MAC.

Se o dispositivo habilitar o bit ADR durante o *uplink*, o Servidor Central poderá controlar diversos de seus parâmetros por meio de comandos MAC no *downlink*, tais como a taxa de dados e a potência de transmissão. Para definir os novos parâmetros, o Servidor de Rede baseia-se na estimativa do *link budget* e no limite da relação sinal-ruído para a qual o pacote consegue ser decodificado corretamente.

Caso o dispositivo controle sua própria taxa de dados ( $ADR = 0$ ), ele deve verificar periodicamente se ainda há conectividade com a rede através dos seguintes passos (HAXHIBEQIRI et al., 2018):

- A cada *uplink* um contador de *acknowledgement* é incrementado ( $ADRACKCnt$ ).
- Quando o contador é maior que o limite ( $ADR\_ACK\_LIMIT$ ), o dispositivo habilita o bit de solicitação de resposta do Servidor de Rede ( $ADRACKReq$ ) e aguarda a confirmação da rede.
- Se não há resposta após um determinado período de tempo ( $ADR\_ACK\_DELAY$ ), o dispositivo primeiramente aumentará sua potência de transmissão e então decrementará sua taxa de dados.
- Se ainda assim não obtiver resposta, continuará o processo até atingir a menor taxa de dados e a maior potência de transmissão.

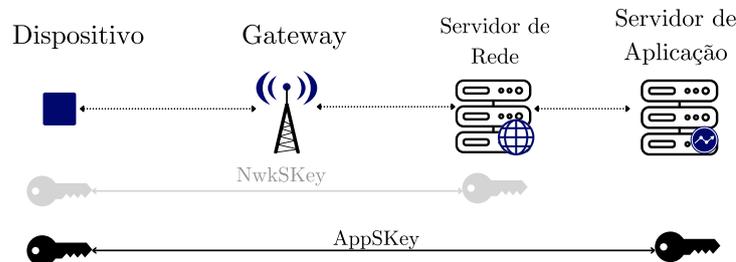
## 2.4 Segurança

A especificação LoRaWAN define que seus dispositivos devem encriptar seu *payload* e *header* com o algoritmo *Advanced Encryption Standard* em duas camadas:

- Chave única de 128 *bits* do dispositivo ao Servidor de Aplicação utilizando a *AppSkey*.

- Chave única de 128 *bits* do dispositivo ao Servidor de Rede, utilizando a *NwkSKey*.

Na figura 8 é mostrada a extensão das chaves criptográficas numa rede LoRaWAN, indo do dispositivo ao Servidor de Rede e de Aplicação.



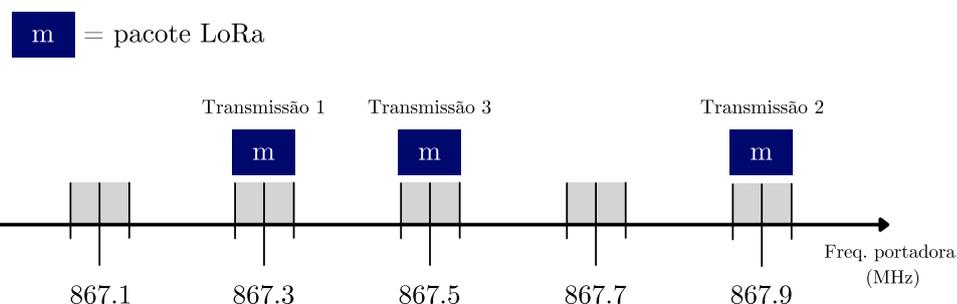
**Figura 8** – Ilustração das chaves de autenticação em uma rede *LoRaWAN*.

Além disso, a integridade da mensagem é garantida por meio do *Message Integrity Code* (MIC). E, para evitar ataques de repetição, LoRaWAN implementa um *frame counter* em seus pacotes, onde o valor de 32 bits reservado no *uplink* (FCntUp) e no *downlink* (FCntDown) é incrementado a cada transmissão ou recepção de dados.

Todavia, antes mesmo de realizar uma comunicação segura, dispositivos LoRaWAN devem ser ativados na rede. Para tanto, redes LoRaWAN empregam um dos dois métodos seguintes para realizar a ativação: *Over-The-Air-Activation* (OTAA) e *Activation-By-Personalisation* (ABP), descritos em sua especificação (LORA ALLIANCE, 2020).

## 2.5 Mecanismos de Transmissão

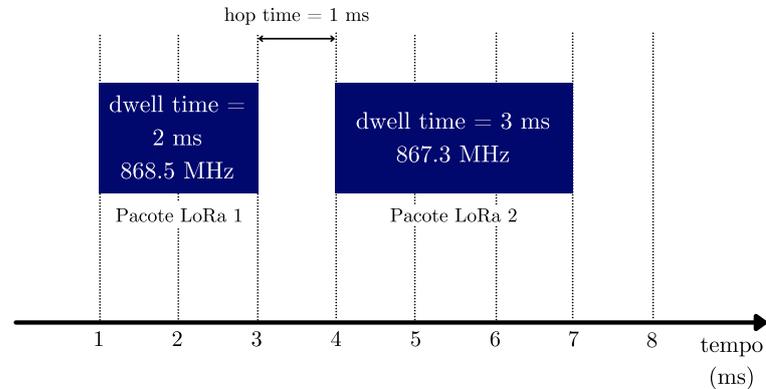
Um dispositivo pode transmitir em qualquer canal disponível a qualquer momento, utilizando qualquer taxa de dados. Além disso, a cada transmissão, o canal é trocado de forma pseudo-aleatória, garantindo maior robustez ao sistema (LORA ALLIANCE, 2020). A figura 9 ilustra o mecanismo de troca de frequência entre uma transmissão e outra.



**Figura 9** – Mecanismo pseudo-aleatório de mudança de canal.

Outros parâmetros importantes na transmissão são o *dwell time* e o *hop time*, que são o tempo necessário para transmitir em uma determinada frequência e o tempo

necessário para trocar de uma frequência para outra, respectivamente. Na figura 10 podemos observar dois pacotes LoRa distintos e seus respectivos tempos de transmissão e de troca de frequência.



**Figura 10** – *Dwell time e hop time.*

## 2.6 Regulamentação

LoRa opera na banda ISM (do inglês *Industrial Scientific and Medical*), na faixa de 433 MHz, 868 MHz ou 915 MHz e largura de banda de 125, 250 ou 500 KHz, a depender do país. Dessa maneira, não é necessário ter uma licença para operar neste espectro, bastando apenas respeitar as regras impostas por agências reguladoras, como a Agência Nacional de Telecomunicações, ANATEL. Por outro lado, com o compartilhamento livre da banda há maior interferência entre os canais, fazendo com que seja necessário criar mecanismos para aumentar a robustez do sistema de comunicação.

De acordo com os parâmetros regionais definidos pela LoRa Alliance ([LORA ALLIANCE, 2021c](#)) e as resoluções aprovadas pela ANATEL ([ANATEL, 2018](#); [ANATEL, 2017](#)), o Brasil utiliza o plano de frequência AU915-928, de 915 MHz a 928 MHz, sem limitação de *duty cycle*, com *dwell time* de 400ms e largura de banda de 125 kHz para *uplink* e 250 kHz para *downlink*. As tabelas 1 e 2 a seguir apresentam o plano de frequência adotado pelo Brasil em maiores detalhes, relacionando cada canal com o *Spreading Factor* (SF), fator relacionado a modulação e explicado na seção 3.2.1, e a respectiva largura de banda.

As tabelas 3 e 4 mostram os parâmetros do plano de frequência EU863-870, adotado no continente europeu.

**Tabela 1** – Sumário dos parâmetros de *uplink* da especificação LoRaWAN para o plano de frequência AU915-928 ([THE THINGS NETWORKS, 2021](#)).

Uplink	
Frequência (MHz)	Parâmetros
916.8	SF7BW125 a SF12BW125
917.0	SF7BW125 a SF12BW125
917.2	SF7BW125 a SF12BW125
917.4	SF7BW125 a SF12BW125
917.6	SF7BW125 a SF12BW125
917.8	SF7BW125 a SF12BW125
918.0	SF7BW125 a SF12BW125
918.2	SF7BW125 a SF12BW125
917.5	SF7BW125 a SF12BW125

**Tabela 2** – Sumário dos parâmetros de *downlink* da especificação LoRaWAN para o plano de frequência AU915-928 ([THE THINGS NETWORKS, 2021](#)).

Downlink	
Frequência (MHz)	Parâmetros
923.3	SF7BW500 a SF12BW500 (RX1)
923.9	SF7BW500 a SF12BW500 (RX1)
924.5	SF7BW500 a SF12BW500 (RX1)
925.1	SF7BW500 a SF12BW500 (RX1)
925.7	SF7BW500 a SF12BW500 (RX1)
926.3	SF7BW500 a SF12BW500 (RX1)
926.9	SF7BW500 a SF12BW500 (RX1)
927.5	SF7BW500 a SF12BW500 (RX1)
923.3	SF12BW500 (RX2)

**Tabela 3** – Sumário dos parâmetros de *uplink* da especificação LoRaWAN para o plano de frequência EU863-870 ([THE THINGS NETWORKS, 2021](#)).

Uplink	
Frequência (MHz)	Parâmetros
868.1 (canal 1)	SF7BW125 a SF12BW125
868.3 (canal 2)	SF7BW125 a SF12BW125 e SF7BW250
868.5 (canal 3)	SF7BW125 a SF12BW125
867.1 (canal 4)	SF7BW125 a SF12BW125
867.3 (canal 5)	SF7BW125 a SF12BW125
867.5 (canal 6)	SF7BW125 a SF12BW125
867.7 (canal 7)	SF7BW125 a SF12BW125
867.9 (canal 8)	SF7BW125 a SF12BW125
868.8 (canal 9)	FSK

**Tabela 4** – Sumário dos parâmetros de *downlink* da especificação LoRaWAN para o plano de frequência EU863-870 ([THE THINGS NETWORKS, 2021](#)).

Downlink	
Frequência (MHz)	Parâmetros
Canais de <i>uplink</i> 1 a 9	RX1
869.525	SF9BW125 (RX2)

## 3 A Camada Física LoRa

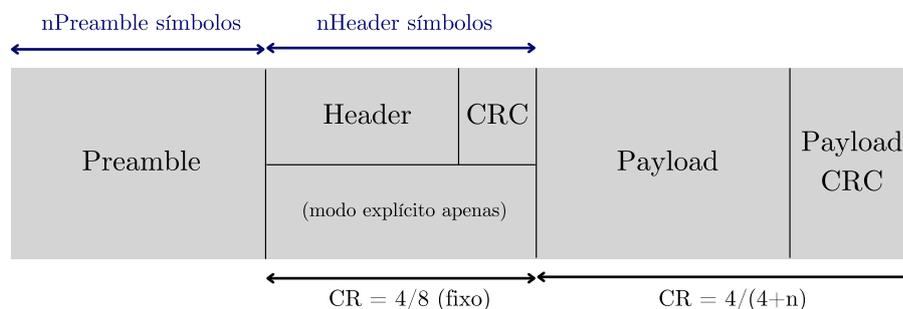
LoRa é uma tecnologia que utiliza técnica de modulação de espalhamento espectral proprietária, derivada da tecnologia *Chirp Spread Spectrum* (CSS) (SEMTECH, 2020b), onde unidades básicas chamadas *chirps* variam linearmente sua frequência ao longo do tempo. Possui baixa taxa de dados, baixa potência e longo alcance. Além da modulação CSS, utiliza codificação Hamming para detectar erros, garantindo robustez. É uma tecnologia resistente a problemas de múltiplos percursos, a fenômenos do tipo Doppler e que possibilita um baixo consumo de energia na transmissão de seus sinais (STANIEC; KOWAL, 2018).

Apesar da especificação LoRaWAN ser *open-source*, a camada física LoRa é fechada e, conseqüentemente, apenas parte de seus detalhes foram publicados abertamente por seus inventores (SELLER, 2014; SEMTECH, 2015). Em conseqüência, o restante das características da tecnologia foram reveladas através de tentativas de engenharia reversa, como em (KNIGHT; SEEGER, 2016; ROBYNS. et al., 2018; MARQUET; MONTAVONT; PAPADOPOULOS, 2020; TAPPAREL et al., 2020).

Neste capítulo, será mostrada a estrutura de um pacote LoRa, os principais parâmetros da camada física, sua modulação CSS e, baseando-se nos trabalhos de engenharia reversa publicados, serão explorados os blocos fundamentais de um transceptor LoRa.

### 3.1 Estrutura de um Quadro LoRa

A estrutura de um quadro LoRa é definida pelo desenvolvedor da tecnologia, Semtech (SEMTECH, 2020a). O quadro é composto de três elementos principais: um preâmbulo, um *header* opcional e um *payload*, como ilustrado na figura 11.



**Figura 11** – Estrutura de um quadro LoRa. Adaptado de (SEMTECH, 2020a).

Nas subseções a seguir serão apresentados os detalhes do quadro LoRa.

### 3.1.1 Preâmbulo

O preâmbulo é composto de símbolos que não representam nenhuma informação, mas são utilizados para fins de sincronização. É constituído de três partes com relação aos *chirps*, descritas na tabela 5.

**Tabela 5** – Descrição do preâmbulo de um quadro LoRa.

Componente	Descrição
<i>preamble</i>	$n$ <i>upchirps</i> de base (não modulados). Em redes LoRaWAN costuma-se empregar $n = 8$ .
<i>frame sync</i>	2 <i>upchirps</i> modulados, utilizados como identificadores de rede.
<i>freq sync</i>	2,25 <i>downchirps</i> de base para sincronização em frequência.

### 3.1.2 Header

O *header* é um componente opcional no quadro, podendo ser transmitido ou não (modo explícito ou implícito). É composto somente de *upchirps*, 3 *bytes* de comprimento no total, com taxa de código CR (do inglês *Coding Rate*) fixa, igual a 4. Uma taxa de codificação fixa garante que o receptor sempre possa decodificar o cabeçalho sem a necessidade de informações preliminares sobre sua taxa de codificação. Uma vez que o cabeçalho é decodificado, a taxa de codificação da carga útil é extraída do segundo byte do cabeçalho, conforme a tabela 6 que descreve sua composição.

**Tabela 6** – Descrição do *header* de um quadro LoRa.

Tamanho	Descrição
1 <i>byte</i>	Tamanho do <i>payload</i>
3 <i>bits</i>	Taxa de codificação do <i>payload</i>
1 <i>bit</i>	Indicação da presença de CRC (do inglês <i>Cyclic Redundancy Check</i> )
4 <i>bits</i>	Não utilizados
1 <i>byte</i>	<i>Checksum</i> indicando a integridade do <i>header</i>

### 3.1.3 Payload

O *payload* de um quadro LoRa contém a carga útil de informação, ou seja, os dados da camada de enlace. É composto somente de *upchirps* modulados e pode ter até 255 bytes de comprimento. Acompanhando o *payload*, há dois *bytes* de CRC indicando a verificação cíclica de redundância do *payload*, também composto somente de *upchirps*.

## 3.2 Parâmetros da Camada Física

Os principais parâmetros configuráveis da camada física LoRa são a largura de banda BW (do inglês *Bandwidth*), o fator de espalhamento SF (do inglês *Spreading Factor*),

taxa de codificação CR, e a potência transmitida (TX\_PWR). A seguir são apresentados na tabela 7 os possíveis valores destes parâmetros.

**Tabela 7** – Principais parâmetros da camada física LoRa e seus possíveis valores.

Parâmetro	Descrição
BW	7,8 a 500 kHz (tipicamente 125, 250 ou 500 kHz)
SF	6 a 12
CR	4/5 a 4/8
TX_PWR	+14 dBm

Nas subseções a seguir serão detalhados os principais parâmetros de um sinal LoRa.

### 3.2.1 Símbolo e *Spreading Factor*

Um símbolo LoRa em banda base é representado por um número decimal, que por sua vez corresponde a um conjunto de *bits* de dados, podendo carregar de 6 a 12 bits de informação por símbolo, onde a quantidade de valores que um símbolo pode assumir será dada pelo fator de espalhamento:

$$M = 2^{SF} \quad (3.1)$$

Na tecnologia LoRa, o fator de espalhamento SF (do inglês *Spreading Factor*) é a razão entre a taxa de símbolo e a taxa de *chip* (unidade básica da sequência de espalhamento utilizada na modulação LoRa).

Por exemplo, o conjunto de dados 1010011 possui 7 *bits* que compõe um símbolo, o que implica em um SF igual a 7. Como  $SF \in \{6, 7, 8, 9, 10, 11, 12\}$  os símbolos LoRa representam números inteiros de 0 a 63 ou até 4095 em decimal, ou seja, carregam de 6 a 12 bits de informação por símbolo, dependendo do SF escolhido.

Um SF maior implica em um maior tempo de símbolo, o que facilita a recepção do sinal mas pode afetar a capacidade da rede quando há muitos nós (CROCE et al., 2018). A tabela 8 a seguir apresenta a quantidade de símbolos possíveis e valores de sensibilidade de recepção para uma banda de 125 kHz utilizando diferentes SFs.

### 3.2.2 Período de Símbolo, Taxas de Símbolo e de Bit

Seja  $T_s$  o período de símbolo e  $BW$  a largura de banda. Então, relacionando o *Spreading Factor* com a largura de banda, temos (SEMTECH, 2015):

$$T_s = \frac{2^{SF}}{BW} \quad (3.2)$$

**Tabela 8** – Quantidade de símbolos e sensibilidade de recepção na banda 125 kHz do transceptor Semtech SX1276 por  $SF$ , adaptado de (SEMTECH, 2020a).

Spreading Factor	Símbolos possíveis	Sensibilidade RX (dBm)
7	128	-123
8	256	-126
9	512	-129
10	1024	-132
11	2048	-133
12	4096	-136

Logo, a taxa de símbolo  $R_s$  é dada por:

$$R_s = \frac{1}{T_s}$$

$$R_s = \frac{1}{\frac{2^{SF}}{BW}}$$

$$\boxed{R_s = \frac{BW}{2^{SF}}} \quad (3.3)$$

Como veremos a frente, LoRa também inclui um código Hamming para detecção e correção de erros provenientes do canal, onde a taxa de codificação  $CR$  é dada por  $4/(4+n_{CR})$  com  $n_{CR} \in \{1, 2, 3, 4\}$ . Então, a taxa de *bit*  $R_b$  nominal será a taxa de símbolo multiplicada pela quantidade de bits em um símbolo ( $SF$ ) e pela taxa de codificação:

$$R_b = R_s \times SF \times CR$$

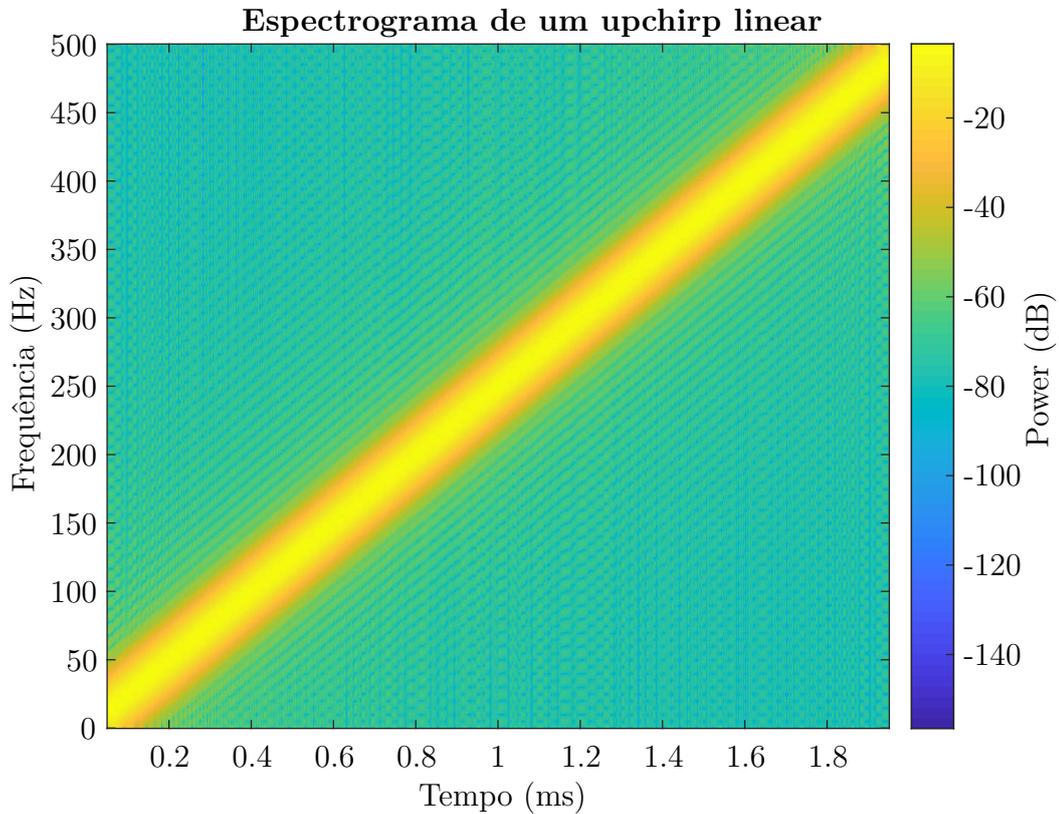
$$\boxed{R_b = \frac{BW}{2^{SF}} \times SF \times \frac{4}{(4+n_{CR})}} \quad (3.4)$$

Então, por exemplo, com boas condições de comunicação para uma largura de banda  $BW = 125$  kHz,  $SF = 7$  e  $CR = 4/5$ , temos uma taxa de *bit* de  $R_b = 5,5$  kbps (AUGUSTIN et al., 2016).

### 3.3 Modulação

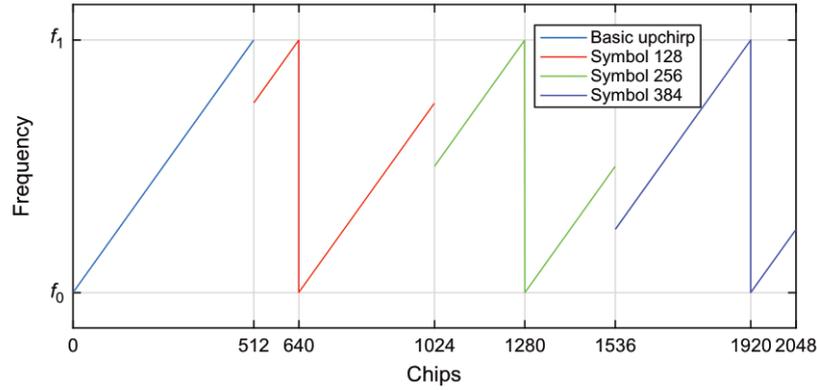
LoRa baseia-se na modulação *Chirp Spread Spectrum* (CSS) para aumentar o alcance e a robustez dos links de comunicação de rádio em comparação com técnicas de modulação tradicionais baseadas em FSK ou OOK, por exemplo (SEMTECH, 2020a). CSS é uma técnica de espalhamento espectral que utiliza *chirps* para modular a informação. *Chirps*

são sinais de frequência variável que incrementam ou decrementam em um determinado período de tempo, indo de uma frequência mínima ( $f_{min}$ ) a uma máxima ( $f_{max}$ ) ou na direção oposta, chamados *upchirps* ou *downchirps* (FABER et al., 2020), como exemplificado na figura 12.



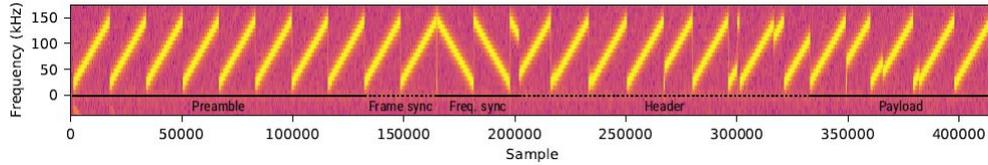
**Figura 12** – Espectrograma de um *upchirp* linear.

Também conhecida como *Frequency Shift Chirp Modulation*, a modulação LoRa tem como elemento portador de informação a mudança de frequência no início do símbolo (VANGELISTA, 2017). Desta forma, para modular um símbolo LoRa em um *chirp*, o modulador realiza um deslocamento inicial em frequência ( $f_{off}$ ) para cada *chirp*, dado pelo valor do símbolo que será transmitido. No caso de um *upchirp*, a frequência instantânea do *chirp* incrementa até  $f_{max}$  a partir do qual seu valor é ciclicamente deslocado para  $f_{min}$ . Então, a frequência instantânea incrementa novamente até atingir a frequência inicial  $f_{off}$ , completando um ciclo dentro do período de símbolo  $T_s$  (FABER et al., 2020). De forma análoga, aconteceria para um *downchirp*. Na figura 13 pode-se observar um exemplo de como LoRa modula os chirps por meio de um deslocamento inicial em frequência (CROCE et al., 2020).



**Figura 13** – Diferentes *chirps* LoRa e seus deslocamentos iniciais em frequência (CROCE et al., 2020).

A figura 14 exemplifica parte de um sinal LoRa real por meio de um espectrograma, onde é possível notar o preâmbulo, o *header* e o *payload* variável nos *chirps*.



**Figura 14** – Espectrograma de um pacote LoRa (ROBYNS. et al., 2018).

### 3.3.1 Descrição Matemática da Modulação LoRa

Seja  $s(nT_s)$  um símbolo LoRa e sendo a taxa de amostragem  $T_{sa}$  dada por  $T_{sa} = 1/BW$ , podemos reescrever o período de símbolo  $T_s$  como  $T_s = 2^{SF} \times T_{sa}$ . Então, um símbolo  $s(nT_s)$  é composto de  $2^{SF}$  amostras, expresso por (VANGELISTA, 2017):

$$s(nT_s) = \sum_{h=0}^{SF-1} \vec{b}_h(nT_s) \times 2^h, \quad (3.5)$$

Onde  $\vec{b}(nT_s)$  é um vetor de SF dígitos binários contendo os *bits* a serem transmitidos. Ou seja,  $s(nT_s)$  basicamente realiza uma conversão de binário para decimal, obtendo um número inteiro entre  $\{0, 1, 2, \dots, 2^{SF} - 1\}$  que definirá o deslocamento inicial em frequência  $f_{off}$ . Então, a forma de onda tem a seguinte expressão:

$$w_s(nT_{sa}) = \sqrt{E_s} \times c_s(nT_{sa}), \quad (3.6)$$

Sendo  $E_s$  a energia do sinal e  $c_s(nT)$  a função base de um *upchirp*, dada por:

$$c_s(nT_{sa}) = \frac{1}{\sqrt{2^{SF}}} e^{j2\pi[(s(nT_s)+n) \bmod 2^{SF}] \frac{n}{2^{SF}}}, \quad (3.7)$$

com índice de amostragem  $n \in \{0, 1, \dots, 2^{SF} - 1\}$ .

Por meio da equação 3.7, podemos observar que  $s(nT_s)$  define a frequência inicial do *upchirp*,  $n$  incrementa a frequência instantânea a cada  $T_{sa}$  segundos e a operação módulo limita a banda do sinal, realizando o retorno cíclico para a frequência inicial (FABER et al., 2020).

### 3.3.2 Demodulador Ótimo

Considerando que a condição necessária para ortogonalidade entre dois sinais LoRa distintos seja satisfeita, ou seja, que o produto interno entre os sinais seja nulo e supondo sincronização perfeita em tempo e frequência, podemos definir o demodulador ótimo (ELSHABRAWY; ROBERT, 2018) pelo seguinte processo de correlação:

$$\sum_{n=0}^{2^{SF}-1} r_k(nT_{sa}) \times c_i^*(nT_{sa}) = \begin{cases} \sqrt{E_s} + \eta_i, & i = k \\ \eta_i, & i \neq k \end{cases} \quad (3.8)$$

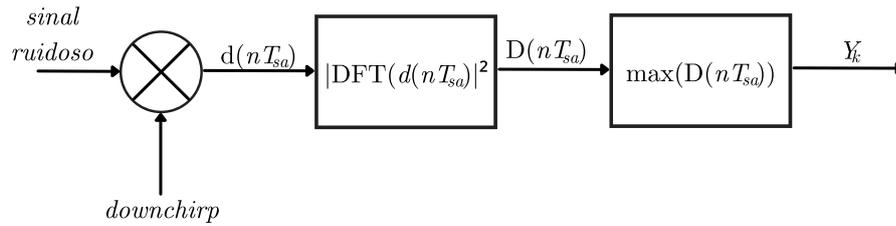
Onde  $c_i^*(nT_{sa})$  é o complexo conjugado de  $c_s(nT_{sa})$  (um *downchirp*),  $r_k(nT_{sa}) = w_s(nT_{sa}) + \eta(nT_{sa})$  é a onda recebida, composta pela onda transmitida  $w_s(nT_{sa})$  somada a um ruído branco aditivo Gaussiano de média zero  $\eta(nT_{sa})$ .

E, de acordo com (ELSHABRAWY; ROBERT, 2018), a detecção do símbolo com maior correlação com a forma de onda transmitida (obtido no domínio da frequência com uma transformada rápida de Fourier) pode ser detectado como:

$$Y_k = \arg_i \max \left( \left| \delta_{k,i} \sqrt{E_s} + \eta_i \right| \right), \quad (3.9)$$

Onde  $\delta_{k,i} = 1$  para  $i = k$  e 0 caso contrário.

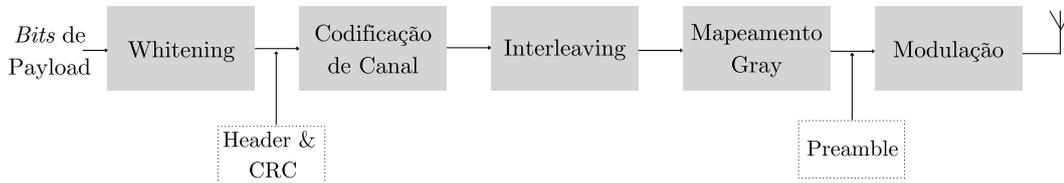
Em resumo, o processo de demodulação consiste basicamente em multiplicar o sinal recebido, amostra por amostra por um *downchirp*, aplicar uma transformada rápida de Fourier no resultado da multiplicação e selecionar o índice de maior correlação (VANGELISTA, 2017). O diagrama em blocos da figura 15 demonstra este processo em tempo discreto.



**Figura 15** – Diagrama em blocos do processo de demodulação. Adaptado de (FIALHO; AZEVEDO, 2018).

### 3.4 Cadeia de Transmissão Completa de um Transceptor LoRa

Além da modulação apresentada anteriormente, o transceptor LoRa executa processamentos adicionais no sinal para aumentar a robustez na comunicação. A cadeia de transmissão LoRa completa é composta por um processo de *whitening*, codificação Hamming, *interleaving* e mapeamento *Gray* antes de modular o sinal (AFISIADIS, 2020) e enviá-lo ao canal de comunicação, como mostra a figura 16. A recepção implementa o processo inverso, ou seja, demodula o sinal, realiza processo de *dewhitening*, decodifica, realiza *deinterleaving* e, por fim, faz o desmapeamento Gray, recuperando assim os *bits* que foram transmitidos.



**Figura 16** – Diagrama em blocos da transmissão de um sinal LoRa. Adaptado de (AFISIADIS, 2020).

A seguir serão descritas as funcionalidades de cada processo dentro do bloco de transmissão completo.

#### 3.4.1 *Whitening*

O *payload* de um sinal LoRa pode conter longas sequências de 0's e 1's, o que introduz um componente DC (do inglês *Direct Current*) no sinal transmitido (SEMTECH, 2020a). Como consequência, essas sequências possuem forte correlação e podem atrapalhar outros algoritmos na cadeia de transmissão que necessitam de sequências fracamente correlacionadas (MARQUET; MONTAVONT; PAPADOPOULOS, 2020), além de fazer com que o sinal possua uma densidade espectral de potência não uniforme (SEMTECH, 2020a). Portanto, o objetivo do processo de *whitening* é randomizar os *bits* antes de serem transmitidos por meio de uma operação XOR (do inglês *exclusive or*) de seus *bits* com

uma sequência aleatória, sendo a mesma tanto na transmissão quanto na recepção do sinal (SEMTECH, 2020a).

### 3.4.2 Codificação Hamming

LoRa utiliza um código linear de bloco Hamming como código corretor de erro, acrescentando uma redundância controlada à mensagem. O tamanho da mensagem  $m$  é fixa em 4 *bits* e as palavras-código  $p$  variam de 5 a 8 (MROUE et al., 2018). Desta forma, sendo a taxa de codificação dada por  $CR = m/p$ , LoRa pode assumir taxas de código  $CR \in \{\frac{4}{5}, \frac{4}{6}, \frac{4}{7}, \frac{4}{8}\}$ .

A detecção de erros só é iniciada a partir de  $CR = 4/6$  e a correção a partir de  $CR = 4/7$ . Portanto, para garantir correção de erros um  $CR \geq 4/7$  deve ser utilizado.

### 3.4.3 Interleaving

*Interleaving* é um processo que consiste em embaralhar as palavras-código geradas pelo codificador de modo a tornar a transmissão mais resistente a *bursts* de interferência.

Ao utilizar *interleaving*, um erro por *burst* é dispersado em várias palavras-código. Como os códigos corretores de erro são projetados para corrigir erros esporádicos em palavras-código, a aplicação de *interleaving* aumentará a robustez do sistema.

O *interleaver* implementado em LoRa é do tipo diagonal com os dois *bits* mais significativos invertidos (MROUE et al., 2018).

### 3.4.4 Mapeamento Gray

O mapeamento Gray utilizado por LoRa serve para garantir que símbolos adjacentes irão diferir 1 *bit*, aumentando as chances de correção de erros devido aos ruídos presentes no canal. Segundo (KNIGHT; SEEBER, 2016), LoRa utiliza um mapeamento Gray inverso na transmissão e, conseqüentemente, um mapeamento Gray convencional aplicado na recepção dos símbolos.

## 3.5 Taxa de Erro de Bit

Existem diversas propostas de expressão da taxa de erro de *bit* (BER, *Bit Error Rate*) de LoRa publicadas, as quais consideram diferentes técnicas para encontrar a BER, como em (AFISIADIS et al., 2020) e (FABER et al., 2020), por exemplo. Dentre os vários artigos publicados, pode-se considerar os trabalhos de (ELSHABRAWY; ROBERT, 2018) como mais promissor ao definir equações aproximadas para a BER de LoRa, pois

consideram apenas canais AWGN e com efeito de desvanecimento plano Rayleigh em um sistema LoRa sem codificação de canal e sem interferência.

Então, (ELSHABRAWY; ROBERT, 2018) denotam  $\Gamma$  como a SNR para a comunicação LoRa:

$$\Gamma = \frac{E_s/T_s}{N_0 \cdot B} = \frac{E_s}{N_0 \cdot 2^{SF}}, \quad (3.10)$$

Dessa forma, a eq. (23) de (ELSHABRAWY; ROBERT, 2018) define a probabilidade de erro de *bit* para um canal AWGN:

$$P_{b_{AWGN}} \approx 0.5 \times Q\left(\sqrt{\Gamma \cdot 2^{SF+1}} - \sqrt{1,386 \cdot SF + 1,154}\right) \quad (3.11)$$

E, sendo:

$$H_m = \sum_{i=1}^m \frac{1}{i}, \quad (3.12)$$

A aproximação da probabilidade de erro de *bit* do LoRa para um canal com desvanecimento plano Rayleigh é dada pela eq. (33) em (ELSHABRAWY; ROBERT, 2018):

$$P_{b_{RAY}} \approx 0.5 \times \left[ Q\left(-\sqrt{2H_{2^{SF-1}}}\right) - \sqrt{\frac{2^{SF} \cdot \Gamma}{2^{SF} \cdot \Gamma + 1}} \cdot \exp\left(-\frac{2H_{2^{SF-1}}}{2 \cdot (2^{SF}\Gamma + 1)}\right) \right. \\ \left. \times Q\left(\frac{\sqrt{2^{SF}\Gamma + 1}}{\sqrt{2^{SF}\Gamma}} \cdot \left[-\sqrt{2H_{2^{SF-1}}} + \frac{\sqrt{2H_{2^{SF-1}}}}{2^{SF}\Gamma + 1}\right]\right) \right] \quad (3.13)$$

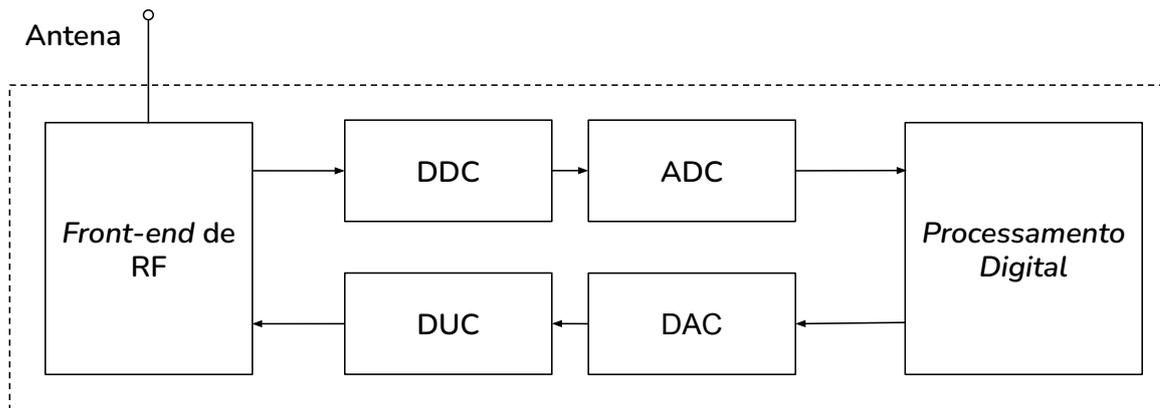
As curvas resultantes das equações aproximadas derivadas pelos autores serão utilizadas posteriormente como referência de acurácia nas simulações desenvolvidas neste trabalho.

## 4 Rádio Definido por *Software*

O termo Rádio Definido por *Software* (do inglês *Software-Defined Radio*, SDR) define uma classe de rádios reconfiguráveis, cuja camada física pode ser significativamente modificada por meio de alterações em *software*. Um sistema SDR é capaz de implementar diferentes funções em diferentes instantes na mesma plataforma, definindo em *software* técnicas de modulação, códigos de correção de erro, frequências de portadoras e padrões diversos (WYGLINSKI, 2013).

Um transceptor pode ser classificado como SDR se suas funcionalidades de comunicação forem implementadas como programas em um processador digital (KRISHNAN et al., 2017). Então, como definido acima, o *hardware* torna-se flexível para utilizar diferentes algoritmos e padrões de recepção ou transmissão definidos via *software*.

A figura 17 representa um sistema de SDR de forma genérica, contendo cinco partes principais (WYGLINSKI MAZIAR NEKOVEE, 2009):



**Figura 17** – Diagrama em blocos de um sistema de SDR genérico (WYGLINSKI MAZIAR NEKOVEE, 2009).

- *Antena*: recebe ou transmite informação contida em ondas de rádio;
- *Front-end de Radiofrequência (RF)*: responsável por receber ou transmitir sinais da antena e converter em uma frequência intermediária;
- *Seção de ADC/DAC (Analog to Digital Conversion, Digital to Analog Conversion)*: responsável por realizar as conversões analógico-digitais e vice-versa;
- *Seção DDC/DUC (Digital Down-Conversion, Digital Up-Conversion)*: responsável pelo deslocamento de frequências intermediárias (IF, *Intermediate Frequency*) para banda base e da banda base para IF, respectivamente.

- *Processamento digital em banda base*: realiza operações de configuração de conexão, equalização, *frequency hopping*, codificação e decodificação, correlação e implementação do protocolo de enlace.

Podemos citar algumas das principais características de um sistema SDR, de acordo com (REED, 2002):

- **Operação multimodo**: Possui a capacidade de suportar vários tipos de funções de rádio usando a mesma plataforma do sistema de comunicação digital;
- **Mobilidade**: Opera de forma transparente com diferentes redes de comunicação, isto é, não confinado a apenas um padrão;
- **Facilidade de desenvolvimento**: As funções de banda-base são um problema de *software*, não um problema de *hardware*.
- **Facilidade de atualização**: Atualizações em *firmware* podem ser realizadas na plataforma SDR para habilitar funcionalidades dos mais recentes padrões de comunicação, trazendo agilidade e facilidade no *upgrade* do sistema.

## 4.1 Principais Aplicações de SDR

Existem diversas aplicações utilizando o conceito de SDR. Serão abordadas, a seguir, algumas das principais aplicações.

### 4.1.1 Rádio Cognitivo

O Rádio Cognitivo é um paradigma de sistemas de comunicação que se concentra em empregar plataformas sem fio inteligentes, altamente ágeis e sensíveis ao ambiente, para escolher e ajustar de forma autônoma os parâmetros operacionais do dispositivo com base nas condições do ambiente de rádio predominantes (WYGLINSKI, 2013). O conceito de Rádio Cognitivo pode ser empregado, por exemplo, para otimizar o acesso ao espectro, permitindo que usuários secundários (não licenciados) compartilhem o recurso espectral disponível dos usuários primários (licenciados) sem causar interferência prejudicial aos usuários licenciados legítimos (EZE et al., 2017). Em muitas situações, a tecnologia SDR é a principal responsável por tornar o Rádio Cognitivo uma realidade, devido à sua agilidade operacional resultante de uma banda de base digital que é totalmente implementada em *software* ou em lógica programável (WYGLINSKI, 2013).

### 4.1.2 Redes de Comunicação Veiculares Dinâmicas

As Redes de Comunicação Veiculares Dinâmicas (do inglês *Vehicular Ad Hoc Networks*, VANETs) são, basicamente, comunicações de veículo para veículo e de veículo para a infraestrutura de rede com base em tecnologias de rede local sem fio. Os principais objetivos de uma rede desta natureza são aumentar a segurança rodoviária, a eficiência do transporte e reduzir o impacto do transporte no meio ambiente por meio da identificação de acidentes, coleta de informações do tráfego local e a utilização inteligente dos recursos de espectro (HARTENSTEIN; LABERTEAUX, 2008). A utilização eficiente dos recursos do espectro, auto-adaptação e acesso dinâmico fazem do Rádio Cognitivo um candidato primário para a comunicação veicular.

### 4.1.3 Comunicações por Satélite

Uma das aplicações de SDR são as comunicações por satélite. Em particular, pode-se citar os CubeSats, que são nano satélites de tamanho e peso padronizados utilizados em missões especiais, como monitoramento de desastres, observação Terrestre, sensoriamento remoto, comunicações e navegação (MAHESHWARAPPA; BOWYER; BRIDGES, 2018).

### 4.1.4 Internet das Coisas

Com a popularidade e o desenvolvimento de diversas verticais no ecossistema de IoT, o conceito de SDR encontra espaço para estudos de novas formas de comunicação sem fio, protocolos e melhorias em tecnologias existentes, como já revisado e exemplificado anteriormente nos trabalhos de (KNIGHT; SEEBER, 2016; ROBYNS. et al., 2018; MARQUET; MONTAVONT; PAPADOPOULOS, 2020; TAPPAREL et al., 2020), entre outros.

## 5 Simulações da Camada Física LoRa

Nesta seção, são apresentados os resultados das simulações desenvolvidas em MATLAB da camada física LoRa em canais sujeitos a desvanecimento plano. Primeiramente, é descrito o modelo do sistema simulado, seu fluxograma, os parâmetros adotados e uma breve introdução teórica dos canais considerados. Então, o desempenho do sistema simulado, ou seja, as curvas BER x SNR (dB) obtidas para SFs iguais a 7 e 12 em canais AWGN, Rice e Rayleigh são discutidas.

### 5.1 Modelo do Sistema

Um modelo simplificado de camada física LoRa foi implementado em MATLAB, o qual não considerou as etapas de *whitening*, codificação de canal e *interleaving* da cadeia de transmissão completa de um sinal LoRa apresentadas na seção 3.4.

De modo aleatório um fluxo de *bits* é gerado e mapeado através de codificação *Gray*, o que garante que símbolos vizinhos difiram de um *bit*. Os *bits* são convertidos para decimal, tendo assim um símbolo LoRa que é enviado ao modulador. O símbolo LoRa modula a frequência inicial de um *upchirp*. O modulador é implementado de acordo com a descrição matemática feita na seção 3.3.1. Então, o sinal modulado é transmitido em um canal Rice com fator K ajustável, induzindo ao comportamento AWGN e Rayleigh nos extremos. Após a propagação do sinal por meio do canal, o sinal recebido é multiplicado por um *downchirp*. O resultado é processado por meio de uma FFT e seu máximo indicará o símbolo de maior correlação, como indica o demodulador da seção 3.3.2. Então, inicia-se o processo reverso de conversão e mapeamento e, ao final, a BER numérica é obtida e os gráficos traçados. O fluxograma da figura 18 ilustra o processo descrito acima.

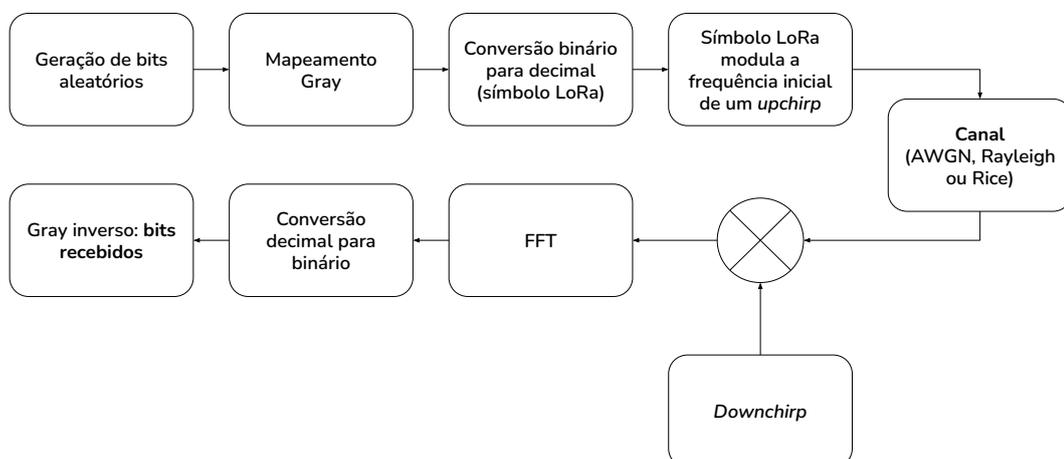


Figura 18 – Fluxograma do sistema LoRa simulado.

Com relação aos valores utilizados nas simulações, a tabela 9 apresenta os parâmetros do sistema.

**Tabela 9** – Parâmetros utilizados nas simulações do sistema LoRa.

Parâmetro	Valor
Largura de Banda	125 kHz
<i>Spreading Factor</i>	7 e 12
Frequência de Amostragem	125 kHz
Tamanho Preâmbulo	8
Tamanho <i>Sync</i>	2
<i>Bits por Frame</i>	980 para SF7; 1680 para SF12
$E_b/N_0$	-10 a 40, passo de 2
Iterações	5000

A largura de banda escolhida é a utilizada comercialmente em redes LoRa. Os *Spreading Factors* de 7 e 12 adotados são o menor e maior empregados em redes LoRaWAN, respectivamente. A frequência de amostragem foi adotada de modo a ser igual a largura de banda do sinal. O preâmbulo de tamanho 8 e *sync* 2 são os convencionalmente utilizados em dispositivos comerciais. Os *bits* por *frame* foram adotados de modo a serem menores que 256 *bytes* (1680 *bits*/8=210 *bytes*), que é o tamanho máximo do *payload* do LoRa. A energia por bit sobre densidade espectral de potência foi escolhida de modo a ser possível obter uma SNR na faixa de -30 dB a 25 dB, que é a faixa de SNR exibida no artigo de referência para a BER teórica (ELSHABRAWY; ROBERT, 2018).

Nas subseções a seguir serão descritos os modelos de canal empregados nas simulações do sistema LoRa.

### 5.1.1 Canal AWGN

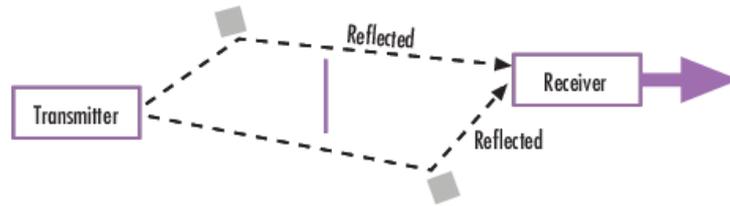
Um canal com Ruído Gaussiano Branco Aditivo (AWGN) é o principal modelo usado para representar o efeito do ruído térmico. O sinal de entrada  $x(t)$  é corrompido apenas por um processo aleatório  $n(t)$  aditivo, que é a única fonte de degradação, onde o ruído composto por componentes distribuídas igualmente em todas as frequências do espectro (CASELLA, 2020).

### 5.1.2 Canal Rayleigh

Além do ruído térmico, outras fontes de erro podem atuar no sinal enquanto ele se propaga no canal, como em canais com múltiplos percursos que espalham o sinal no tempo.

No caso de modelos de canal Rayleigh, devido aos diversos obstáculos presentes entre o transmissor e receptor, com frequência não há linha de visada (*Line of Sight*, LOS)

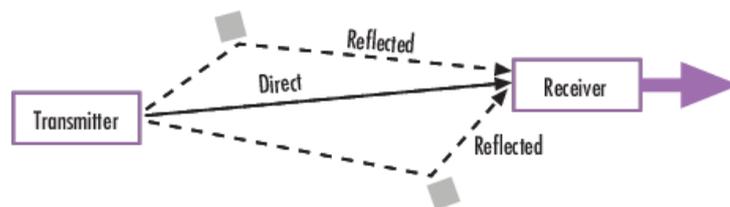
na transmissão, como ilustra a figura 19. Os obstáculos presentes na vizinhança do receptor espalham o sinal enviado, fazendo com que cheguem diversas cópias do sinal transmitido com amplitudes, fases e ângulo diferentes, aleatórios (SILVA, 2004).



**Figura 19** – Representação de um modelo de canal Rayleigh. Adaptado de (MATLAB, 2022).

### 5.1.3 Canal Rice

Se houver uma componente incidindo sobre o receptor com uma potência significativamente maior que as demais, então o sinal no receptor terá uma envoltória com distribuição Rice (SILVA, 2004). Essa componente com potência predominante, como ilustra a figura 20, recebe o nome de componente direta. Para mensurar a potência da componente dominante, é definido um fator chamado "K", ou coeficiente Rice, que é definido como a razão da potência do sinal no componente dominante sobre a potência espalhada (média local).



**Figura 20** – Representação de um modelo de canal Rice. Adaptado de (MathWorks, 2022).

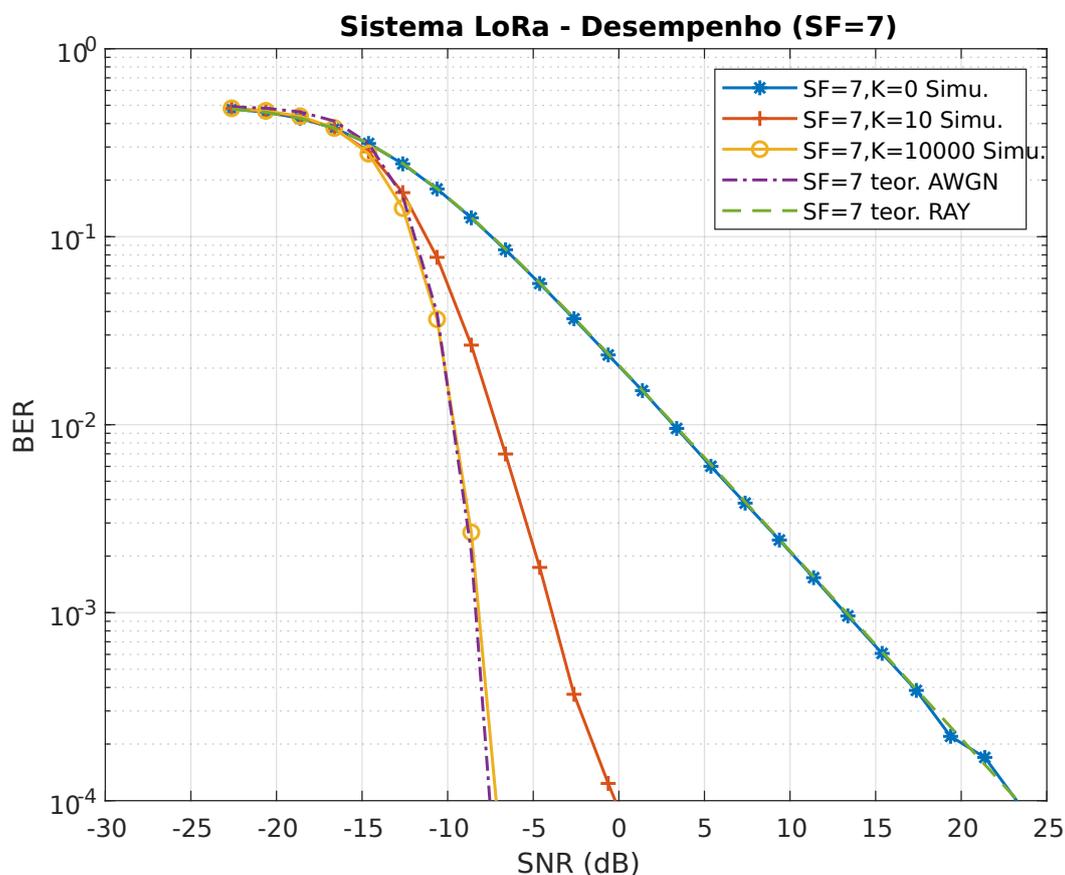
## 5.2 Desempenho do Sistema Simulado

Como apresentado na seção 3.5, o trabalho de (ELSHABRAWY; ROBERT, 2018) serviu como base teórica para as simulações do sistema. Por meio do equacionamento apresentado no artigo foi possível determinar os limites teóricos de desempenho da tecnologia LoRa em canais AWGN e Rayleigh. Dessa forma, espera-se que o desempenho do sistema fique próximo destes casos limites.

Então, foram feitas simulações considerando que o canal adotado muda a cada símbolo LoRa (desvanecimento plano lento) e, por meio da alteração do fator K, foi possível reproduzir os seguintes comportamentos no canal:

- Canal com múltiplos percursos e sem componente direta (Rayleigh,  $K=0$ ).
- Canal com múltiplos percursos, com linha de visada e componente direta e potência predominante (Rice,  $K=10$ ).
- Canal com componente direta muito maior do que os múltiplos percursos, ou seja, um coeficiente Rice  $K \rightarrow \infty$ , correspondendo assintoticamente ao canal AWGN.

O desempenho foi analisado para  $SF = 7$  e  $SF = 12$ , mantendo todos os outros parâmetros apresentados na tabela 9 fixos. As figuras 21 e 22 apresentam as curvas obtidas.

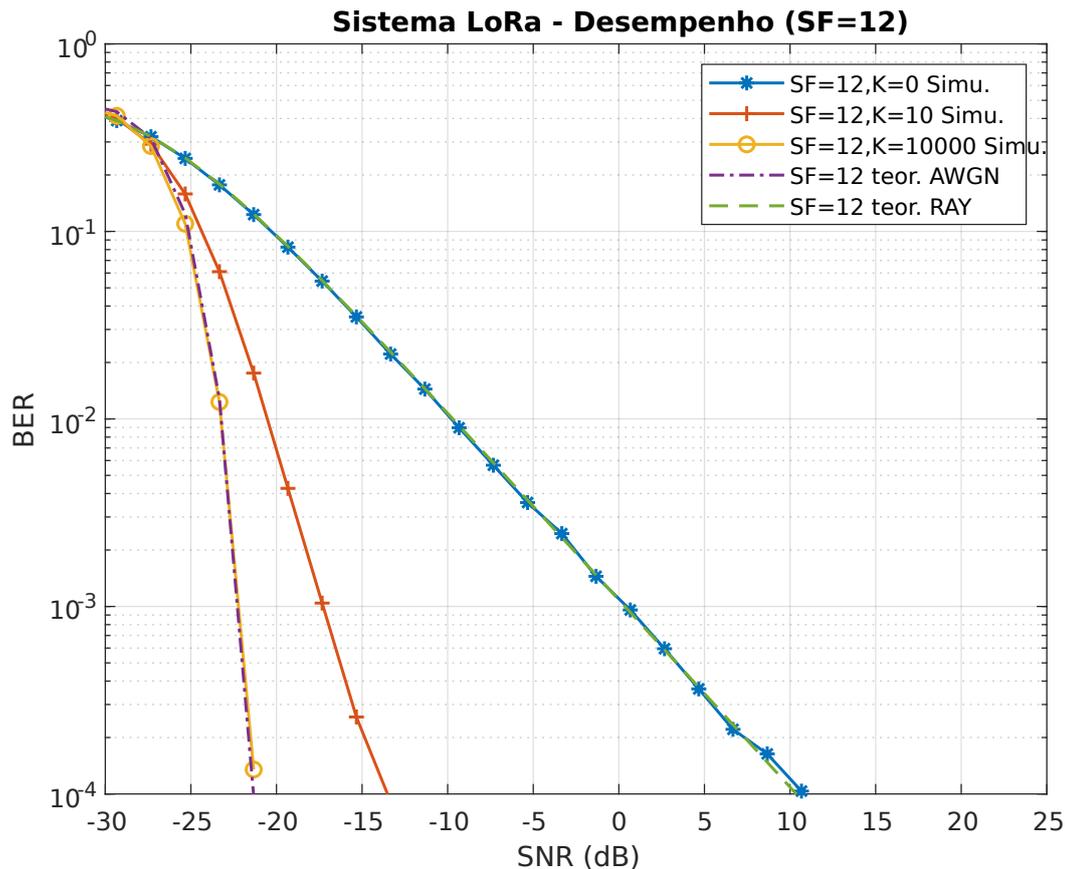


**Figura 21** – Curva BER  $\times$  SNR (dB) do sistema LoRa com  $SF=7$  em um canal com desvanecimento plano lento do tipo Rice, com coeficiente  $K$  variante, sem codificação de canal.

Analisando as curvas da figura 21 e começando pelo pior caso possível, ou seja, um canal sem linha de visada e com múltiplos percursos (Rayleigh), esperava-se que o sistema obtivesse uma BER de  $10^{-3}$  e  $10^{-4}$  com uma SNR de aproximadamente 13 dB e 23 dB, respectivamente, ao utilizar  $SF=7$ , dados pela curva "SF=7 teor. RAY". Além disso, no domínio log esta curva deveria apresentar-se como linear. Com o sistema simulado foram obtidos resultados coerentes com o teórico, tanto em valores quanto em forma.

Ao adicionar uma componente direta com potência predominante ( $K > 0$ ), o sistema simulado tendeu a se aproximar do caso AWGN, como esperado e apresentado na curva "SF=7,K=10 Simu."

Ao aumentar significativamente o valor do coeficiente Rice (curva "SF=7, K=10000 Simu."), as componentes de múltiplo percurso foram suprimidas pela componente dominante, fazendo com que o canal apresentasse comportamento AWGN.



**Figura 22** – Curva BER  $\times$  SNR (dB) do sistema LoRa com SF=12 em um canal com desvanecimento plano lento do tipo Rice, com coeficiente K variante, sem codificação de canal.

Já ao analisar a figura 22, observa-se que os resultados simulados se mostraram coerentes com os teóricos e as curvas para SF=7 e SF=12 possuem o mesmo formato, diferindo de um deslocamento para a esquerda da figura 21 para a figura 22.

Isso representa que ao aumentar o *Spreading Factor* do sistema de 7 para 12 é possível obter um ganho de desempenho de aproximadamente 13 dB, independentemente do modelo de canal analisado (AWGN, Rice e Rayleigh). Além disso, com este resultado nota-se que a menor SNR onde é possível o sistema operar com baixa BER, dadas as condições pré-estabelecidas nas simulações, é de cerca de -20 dB em um canal AWGN com SF=12.

Portanto, uma maneira de obter taxas de erro de *bit* pequenas, em canais com baixa relação sinal-ruído, é utilizar um SF maior. Em contrapartida, incrementar o SF implica em tempo de símbolo maior (conforme a equação 3.2), o que pode ser negativo em implementações com grande número de dispositivos, pois aumentarão as chances de colisão entre símbolos.

Outra maneira de melhorar o desempenho do sistema seria por meio da implementação do restante dos processos da cadeia de transmissão apresentados na seção 3.4, principalmente a codificação Hamming adotada em LoRa, sendo uma proposta de investigação para trabalhos futuros.

## 6 Avaliação Experimental da Camada Física LoRa

Nesta seção, são apresentados os dispositivos, *softwares*, a identificação dos sinais LoRa utilizando SDR, assim como os resultados obtidos com o nó transmissor LoRa e o receptor SDR *open source* em ambientes interno e externo.

### 6.1 Hardware Utilizado

#### 6.1.1 Heltec WiFi LoRa 32

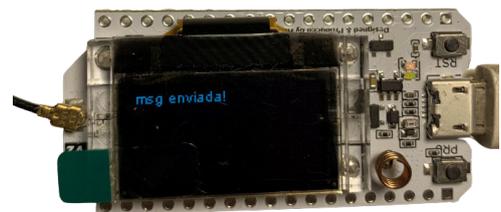
Como nó transmissor foi utilizada a placa de desenvolvimento WiFi LoRa 32 projetada e produzida pela Heltec Automation, baseada no SoC (*System on Chip*) ESP32, transceptor SX1276 da Semtech, sistema de gerenciamento de bateria Li-Po e tela OLED de 0,96”.

Esta placa pode operar tanto em 868MHz quanto em 915MHz, possui suporte para desenvolvimento em Arduino IDE, com uma biblioteca disponibilizada pelo fabricante com diversos exemplos (HELTEC, 2019), onde basendo-se nas APIs (*Application Programming Interface*) (HELTEC, 2020) é possível ajustar os principais parâmetros da camada física LoRa, tais como a largura de banda, SF, CR, tamanho do preâmbulo, entre outros.

Desta forma, foi desenvolvido um *firmware* em Arduino IDE, o qual define os parâmetros mostrados na tabela 10, inicia o rádio LoRa e o display OLED e, em seguida, entra em *loop* constante enviando uma mensagem "count 0", "count 1", "count 2", assim por diante, até "count 999", a cada segundo, mostrando o conteúdo da mensagem na tela OLED e na porta de comunicação serial do dispositivo. A figura a seguir mostra um exemplo do conteúdo exibido no dispositivo.



(a) Mensagem sendo enviada.



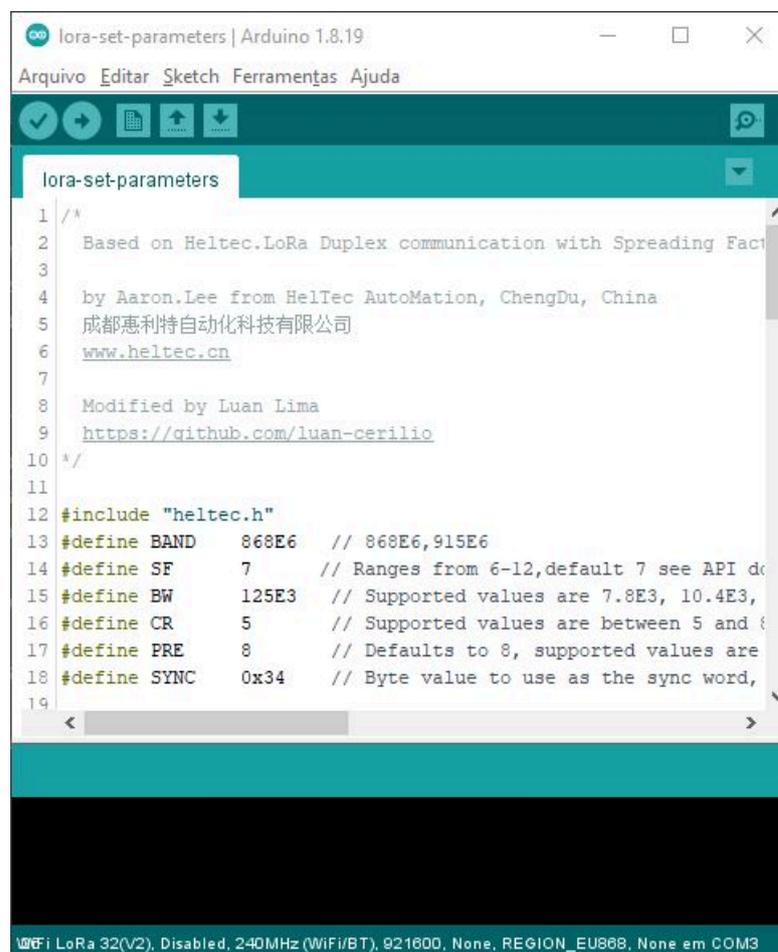
(b) Mensagem enviada.

**Figura 23** – Conteúdo exibido no display OLED da placa de desenvolvimento Heltec WiFi LoRa 32 ao enviar um pacote LoRa.

Tabela 10 – Parâmetros ajustáveis no *firmware* do dispositivo.

Parâmetro	Descrição	Possíveis valores
BAND	Frequência de Operação	868 e 915 MHz
SF	<i>Spreading Factor</i>	6-12, 7 quando não declarado
BW	Largura de banda	7.8, 10.4, 15.6, 20.8, 31.25, 41.7, 62.5, 125 e 250 kHz
CR	<i>Coding Rate</i>	Entre 5 e 8
PRE	Tamanho do preâmbulo	6-65535, 8 quando não declarado
SYNC	<i>Byte</i> de sincronização	Por padrão 0x34

Estes parâmetros são definidos no cabeçalho do código, assim como mostra a figura 24.



```

1 /*
2  Based on Heltec.LoRa Duplex communication with Spreading Factor
3
4  by Aaron.Lee from HelTec AutoMation, ChengDu, China
5  成都惠利特自动化科技有限公司
6  www.heltec.cn
7
8  Modified by Luan Lima
9  https://github.com/luan-cerilio
10 */
11
12 #include "heltec.h"
13 #define BAND      868E6 // 868E6,915E6
14 #define SF        7 // Ranges from 6-12,default 7 see API doc
15 #define BW        125E3 // Supported values are 7.8E3, 10.4E3,
16 #define CR        5 // Supported values are between 5 and 8
17 #define PRE       8 // Defaults to 8, supported values are
18 #define SYNC      0x34 // Byte value to use as the sync word,
19

```

Figura 24 – Ambiente de desenvolvimento Arduino IDE mostrando o cabeçalho do código.

### 6.1.2 RTL-SDR

O dispositivo SDR utilizado foi o *dongle* RTL-SDR, desenvolvido pela Realtek, a qual tinha como objetivo inicial receber sinais televisivos DVB-T e hoje é comumente utilizado como SDR após descobrirem que as amostras I e Q no *chipset* RTL2832U podem ser acessadas diretamente (RTL-SDR, 2013). É baseado nos *chipsets* R802T, o qual

converte o sinal para banda base, e no *chipset* RTL2832U, o qual transforma em amostras que são enviadas para a interface USB. A tabela 11 resume as principais características do *dongle*.

**Tabela 11** – Principais características do *dongle* RLT-SDR (RTL-SDR, 2013)

RTL-SDR	
Frequência mín.	24 MHz
Frequência máx.	1766 MHz
Largura de banda	3,2 MHz
Resolução ADC	8 bits
Transmissor?	Não

Por ser popular, de baixo custo se comparado a dispositivos mais complexos como LimeSDR (Lime Microsystems, 2020) ou HackRF One (Great Scott Gadgets, 2014) e trabalhar na faixa de frequência em que LoRa opera, foi utilizado como receptor SDR neste trabalho. A seguir é mostrado o dispositivo adquirido.



**Figura 25** – RTL-SDR utilizado na avaliação experimental.

## 6.2 Identificação de sinais LoRa utilizando SDR#

Por meio da utilização do *software* SDR#, instalado no sistema operacional Windows 10, foram feitas as primeiras análises dos sinais LoRa. O *setup* em ambiente interno foi montado conforme mostra a figura 26, conectando o RTL-SDR a uma porta USB do notebook e o nó transmissor posto a cerca de 12 cm. Para poder locomover-se com o

transmissor posteriormente e mantê-lo com a antena fixa, foi utilizado um *power bank* de 5V/1A como fonte de alimentação do nó transmissor e a antena foi fixada em uma caixa plástica, conforme mostra a figura 27.

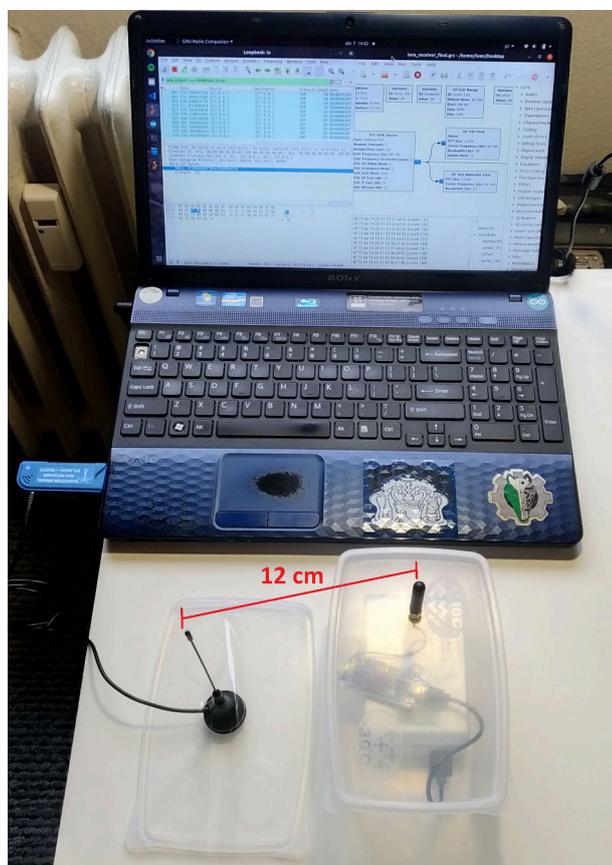


Figura 26 – *Setup* experimental em ambiente interno.



(a) Parte interna do transmissor com caixa plástica.



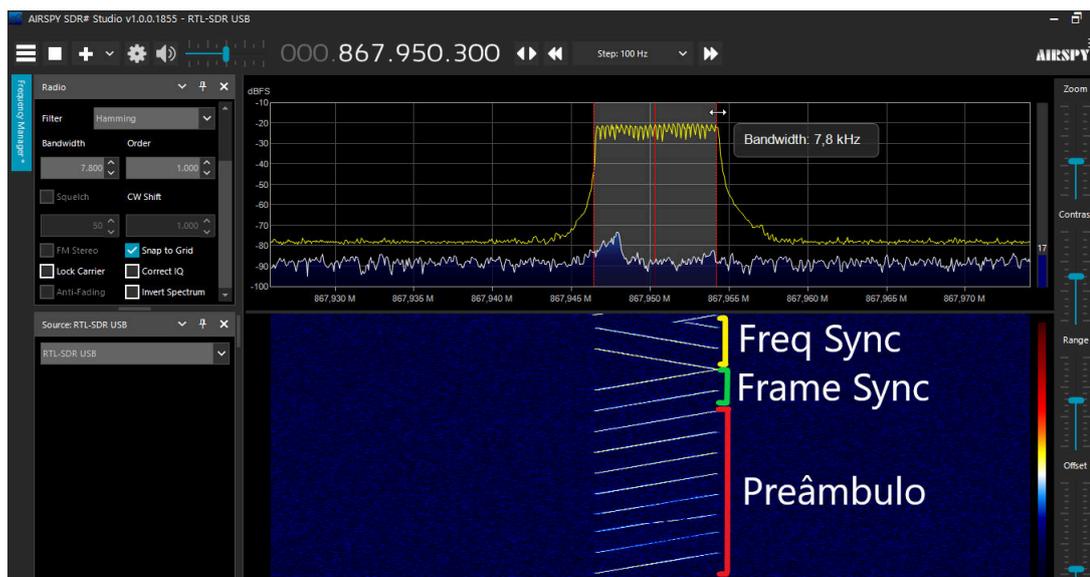
(b) Transmissor com a antena fixa pela caixa.

**Figura 27** – *Setup* experimental do transmissor.

Considerando o período de símbolo dado pela equação 3.2, é possível visualizar o sinal ao utilizar um SF alto e a menor largura de banda possível. Dessa forma, alterando o *firmware* do dispositivo para SF=12 e BW=7,8 kHz, o período de símbolo passa a ser grande o suficiente para identificar os *chirps* no SDR#. A tabela 12 apresenta as configurações do dispositivo.

**Tabela 12** – Parâmetros utilizados no transmissor ao capturar os pacotes LoRa com SDR#.

Config	Parâmetros			
	SF	BW (kHz)	$P_{TX}$ (dBm)	CR
1	7	7,8	14	4/5



**Figura 28** – Captura de um símbolo LoRa com SF=12 e BW=7.8 kHz utilizando o *software* SDR#.

Na figura 28 é possível observar os 8 *upchirps* de preâmbulo, 2 *upchirps* de sincronização de *frame*, 2,25 *downchirps* de sincronização de frequência e a largura de banda do sinal (7,8 kHz). Iniciando com uma frequência central de 868 MHz, foi possível identificar exatamente em qual frequência o sinal é transmitido (867,950 MHz) e perceber que há um *offset* de 50 kHz com relação ao canal de *uplink* 8 (conforme tabela 3) ao ajustar a captura do sinal dentro do SDR#. Esta frequência será importante posteriormente, no bloco de recepção e decodificação da mensagem no GNU Radio.

### 6.3 Identificação de sinais LoRa utilizando GNU Radio

O GNU Radio é um kit de ferramentas de desenvolvimento de *software* gratuito e de código aberto que fornece blocos de processamento de sinais para implementar sistemas de SDR em Linux, Windows e macOS (GNU Radio, 2022). Ele pode ser usado com hardware de RF externo ou sem hardware, como um ambiente de simulação. Seu ambiente de desenvolvimento tem uma estrutura modular e diversas bibliotecas nativas, prontas para implementação.

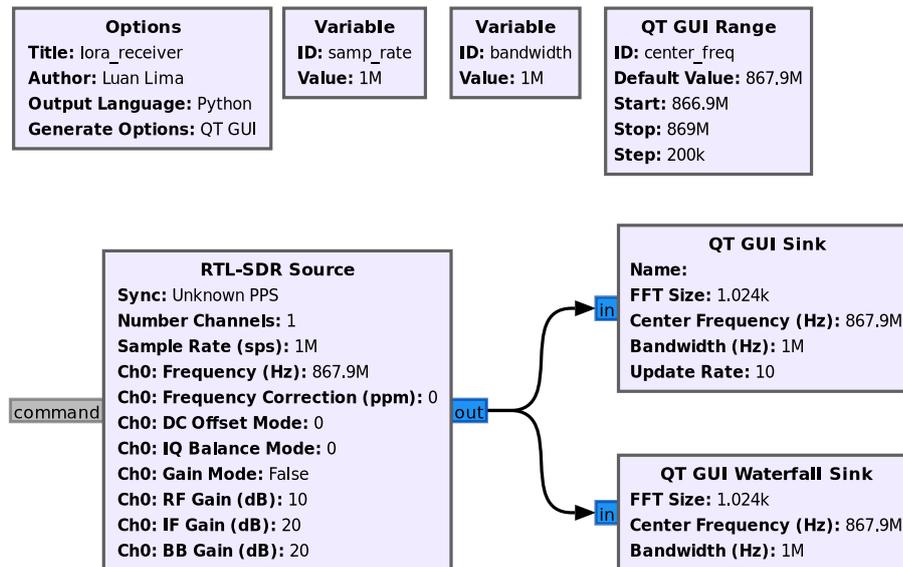
Como o *software* é de código livre, é possível modificar os blocos existentes e criar novos para atender objetivos específicos, caso necessário, como no caso do projeto gr-osmosdr para utilizar o *dongle* RTL-SDR (STOLNIKOV, 2021).

Portanto, para iniciar os testes com o GNU Radio, primeiramente foi feito um ajuste nos parâmetros do *firmware* do dispositivo, conforme a tabela 13 a seguir.

**Tabela 13** – Parâmetros utilizados no transmissor ao capturar os pacotes LoRa com GNU Radio.

Config	Parâmetros			
	SF	BW (kHz)	$P_{TX}$ (dBm)	CR
1	7	125	14	4/5

Ao iniciar um novo projeto e construir o fluxo gráfico da figura 29, é possível reproduzir as funcionalidades encontradas no SDR# e identificar a frequência e a largura de banda do sinal. A taxa de amostragem escolhida de 1 Msps está dentro da faixa de operação do RTL-SDR. A largura de banda de 1 MHz foi definida de maneira a ser suficiente para visualizar cinco canais: por exemplo, do canal 4 ao canal 8 do plano de frequência EU863-870 (tabela 3), ou seja, de 867,1 MHz a 867,9 MHz. A frequência central foi definida como sendo 867,9 MHz (canal 8). A frequência inicial da interface gráfica (QT GUI Range) foi definida como sendo 867 MHz, que é menor que o canal 4 descontado de metade da largura de banda ( $867,1 - 0,125/2 = 867,0375$  MHz), a final como 869 MHz, que é maior que o canal 9 somado a metade da largura de banda ( $868,8 + 0,125/2 = 868,8625$  MHz) e o passo em 200 kHz, que é o valor do espaçamento entre canais.

**Figura 29** – Fluxo gráfico inicial de identificação do sinal LoRa em GNU Radio.

Então o sinal pôde ser visualizado na interface gráfica, conforme a figura 30. O gráfico em cascata “*Time* (s) x *Frequency* (MHz)” não possui resolução tão boa quanto o SDR#, de maneira que a visualização dos *chirps* é um pouco pior.

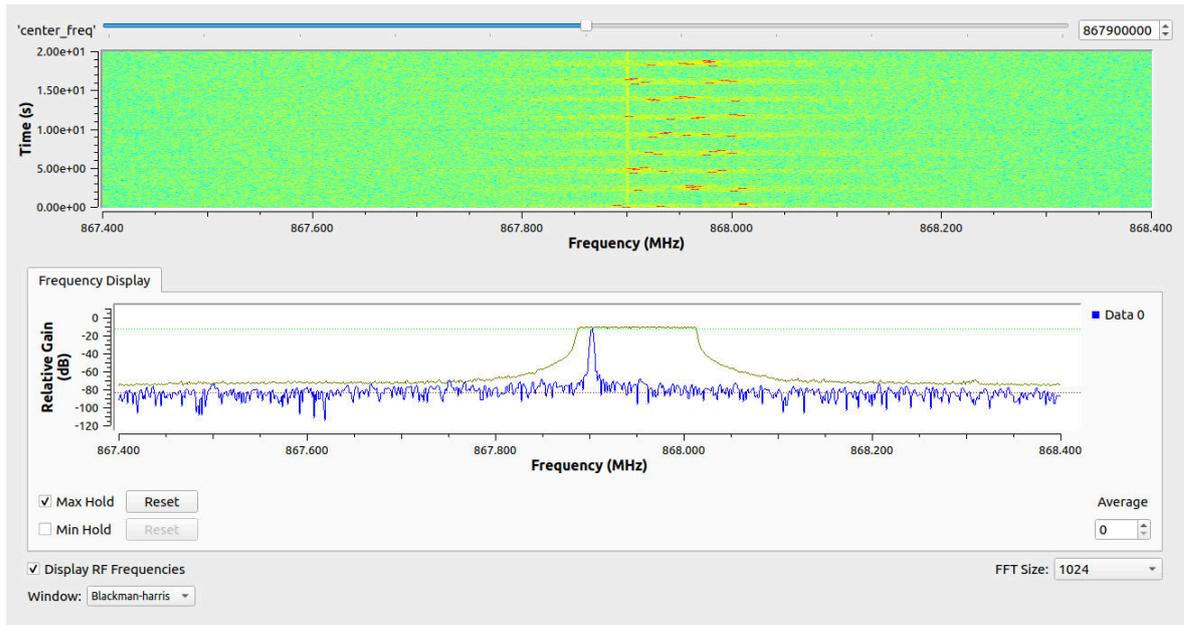


Figura 30 – Visualização do espectro do sinal utilizando GNU Radio.

Ao identificar a frequência central da transmissão na figura 30, é possível perceber um *offset* de cerca de 50 kHz com relação ao que deveria ser a frequência do canal (867,900 MHz). Então, foi adicionado o bloco "LoRa Receiver" (ROBYNS, 2017a) e o fluxo da figura 31 gerado, onde o parâmetro *channel list* deste bloco é definido como a soma da frequência central e este *offset* (variáveis  $center\_freq + offset$ ). Sem este *offset*, o bloco não consegue identificar o sinal e decodificar as mensagens.

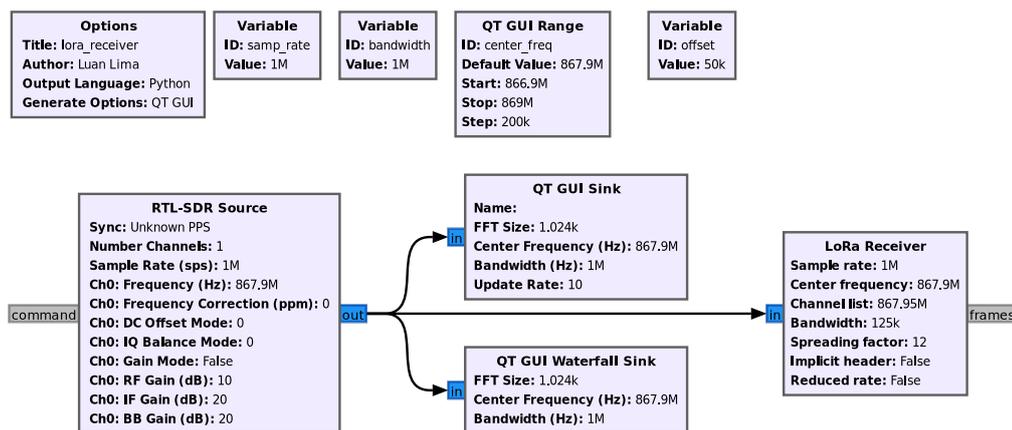


Figura 31 – Fluxo gráfico com o bloco LoRa Receiver adicionado.

Com um SF 12, é possível ver na console as mensagens sendo recebidas, ainda que não decodificadas perfeitamente, conforme indica a figura 32a. Ao alterar o SF para 7, o bloco consegue decodificar as mensagens perfeitamente. Os caracteres adicionais ao final da mensagem referem-se aos *bytes* de CRC habilitados na transmissão.



Figura 32 – Decodificação de um sinal LoRa na console do *software* GNU Radio.

Para processar os dados LoRa recebidos posteriormente, um bloco “*Message Socket Sink*” pode ser usado para transmitir mensagens UDP para o *localhost* na porta 40868 e capturá-las com o Wireshark, salvando o arquivo pcap resultante da captura, por exemplo, resultando no fluxo gráfico final da figura 33. As capturas na interface de *loopback* do computador são observadas na figura 34.

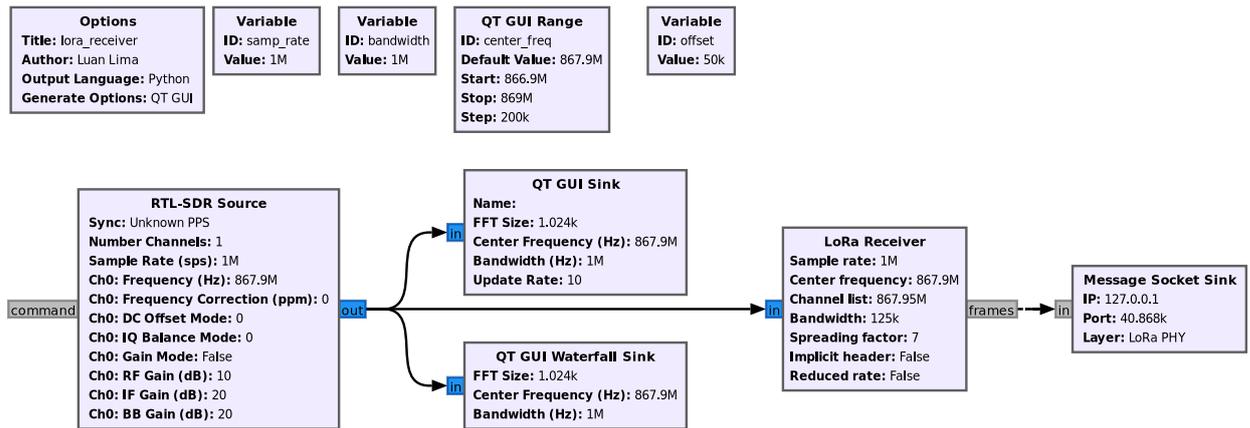


Figura 33 – Fluxo gráfico definitivo de recepção e decodificação de mensagens LoRa.

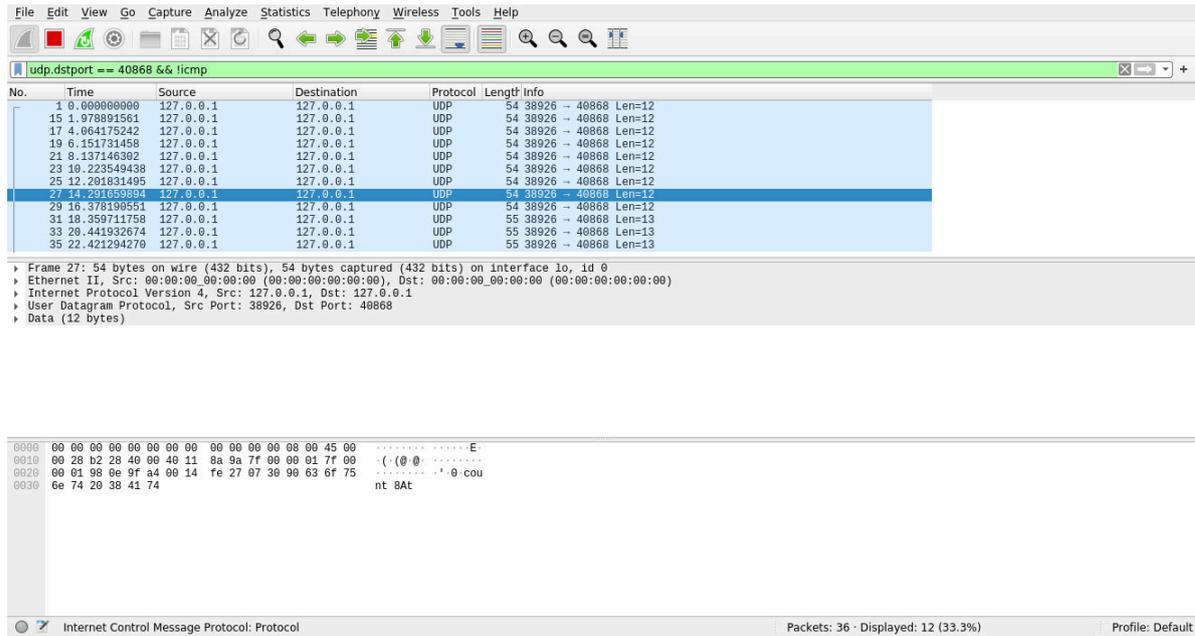


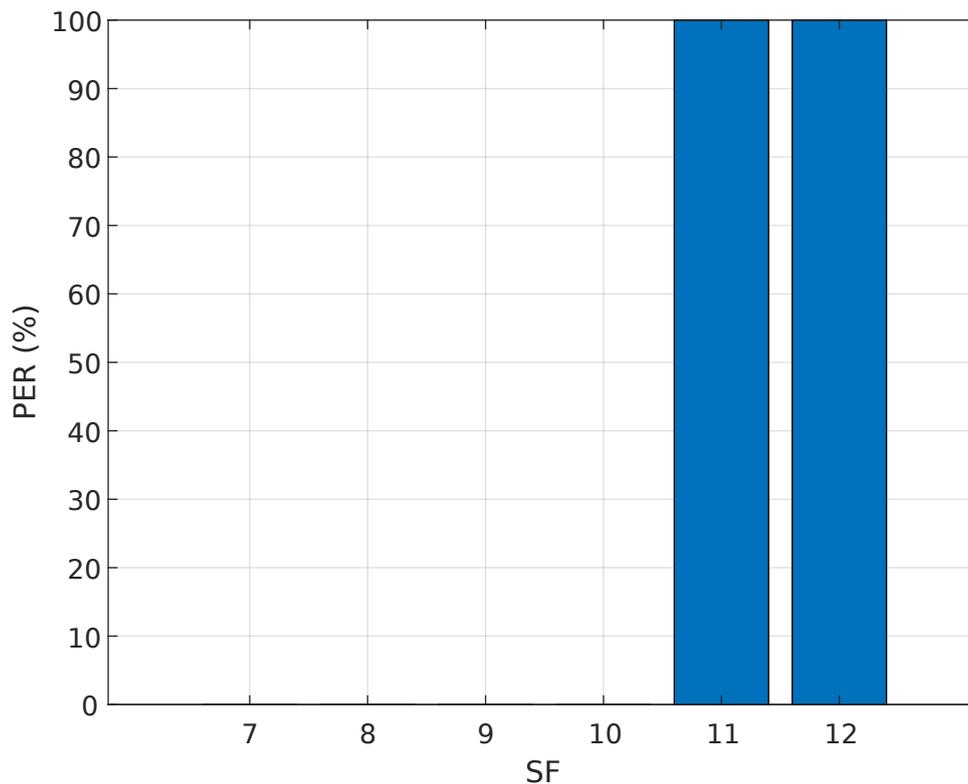
Figura 34 – Pacotes LoRa sendo visualizados no *software* Wireshark.

## 6.4 Desempenho em Ambiente Interno

Para avaliar a eficácia do receptor SDR LoRa *open source* em ambiente interno, foram enviados e capturados no Wireshark 1000 pacotes LoRa, com os parâmetros da tabela 14 e distância de 12cm entre transmissor e receptor, conforme a figura 26, variando apenas o SF de 7 a 12. A carga útil da mensagem, ou *payload*, é composta por uma mensagem "count 0", "count 1" até "count 999", quando o nó para a transmissão. No GNU Radio, foi utilizado o fluxo da figura 33, alterando apenas o SF de acordo com o valor em questão. Os resultados obtidos são indicados no gráfico da figura 35.

Tabela 14 – Parâmetros definidos no *firmware* do dispositivo na análise em ambiente *indoor*.

Config	Parâmetros			
	SF	BW (kHz)	$P_{TX}$ (dBm)	CR
1	7	125	14	4/5
2	8	125	14	4/5
3	9	125	14	4/5
4	10	125	14	4/5
5	11	125	14	4/5
6	12	125	14	4/5



**Figura 35** – Taxa de erros de pacotes recebidos pelo receptor SDR em função da configuração em ambiente interno a 12 cm entre transmissor e receptor e parâmetros de *firmware* conforme tabela 14.

Nota-se que ao utilizar SF=7, 8, 9 e 10 (configurações 1 a 4) foi possível decodificar perfeitamente todos os pacotes. Com SF=11 e 12 (configurações 5 e 6) não foi possível obter nenhum pacote perfeito, ainda que seu conteúdo estivesse parcialmente legível. Por exemplo, por vezes era recebida a mensagem "coe 19" ao invés de "count 19", o que indica que nem todos os *bits* sofreram erros durante a transmissão. Além disso, também foi observado que, ao utilizar SF=12, alguns pacotes eram recebidos "fora de ordem", ou seja, uma mensagem "coe- 24", por exemplo, foi recebida depois das mensagens "coe- 26" e "coe- 27".

Como o bloco de recepção em GNU Radio desenvolvido pelos autores propicia pouca flexibilidade para testar os parâmetros, não é possível dizer certamente o porquê as mensagens não são decodificadas perfeitamente ao utilizar os SFs mais altos, mas é sabido por parte dos autores de que são necessários aperfeiçoamentos no projeto para decodificar perfeitamente ao utilizar SF=12, já que os mesmos realizaram testes similares e chegaram nos mesmos resultados para este SF (ROBYNS, 2017b).

## 6.5 Desempenho em Ambiente Externo

Em seguida foram feitas medidas de alcance de recepção dos sinais em ambiente externo utilizando o fluxo da figura 33. A figura 36 mostra o *setup* com a antena receptora fixada na janela do apartamento.



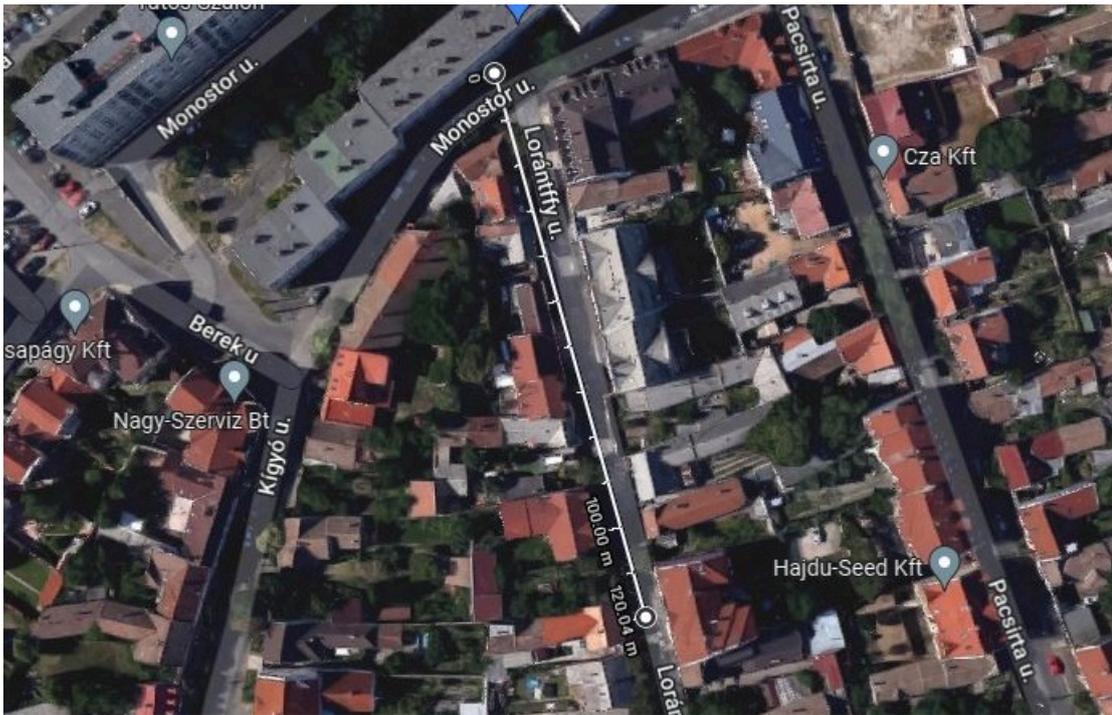
**Figura 36** – *Setup* experimental do receptor para ambiente externo.

Então, foram definidos cinco pontos onde 1000 pacotes LoRa foram enviados pelo nó transmissor utilizando SF=7 e SF=10, os quais foram o menor e maior SF em que se pôde decodificar as mensagens perfeitamente nos testes iniciais, respectivamente. Ao receber os pacotes no GNU Radio, os dados foram capturados pelo Wireshark e salvos para processamento posterior. A figura 37 ilustra a disposição da antena receptora com relação aos pontos onde foi posto o nó transmissor, buscando a melhor linha de visada direta possível.



**Figura 37** – Pontos de transmissão do sinal.

Em cada ponto foram feitos os envios dos pacotes, mantendo o transmissor estático em uma altura do chão de aproximadamente 170 cm em todos os cinco pontos. O ponto mais distante (ponto 5), que é o último ponto onde ainda foi possível receber pacotes legíveis, ficou a uma distância de cerca de 120 metros da base do prédio, conforme indica a visualização em 2D obtida pelo Google Maps na figura 38.



**Figura 38** – Visualização em 2D obtida pelo Google Maps do ponto mais distante.

Para definir as distâncias em linha de visada, ou seja, a distância em linha reta entre o nó transmissor e a antena receptora, foi feito o seguinte cálculo:

$$d_{vis} = \sqrt{d_{base}^2 + h_{pred}^2}, \quad (6.1)$$

Onde  $d_{vis}$  é a distância de linha de visada em metros,  $d_{base}$  é a distância entre o ponto (1 a 5) e a base do prédio e  $h_{pred}$  é a altura do prédio (da janela).

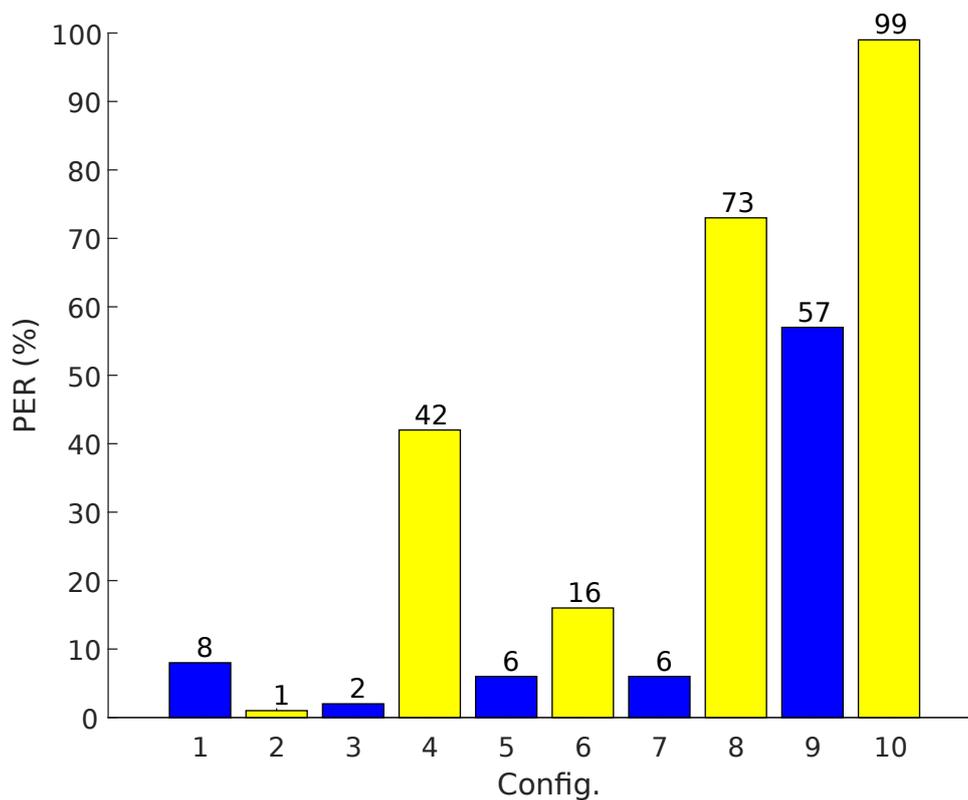
**Tabela 15** – Distâncias de linha de visada de cada ponto medido.

Altura prédio	Ponto	Distância até a base (aprox.)	Distância de visada (aprox.)
15 m	1	10 m	<b>18 m</b>
	2	35 m	<b>38 m</b>
	3	50 m	<b>52 m</b>
	4	95 m	<b>96 m</b>
	5	120 m	<b>120 m</b>

Foram feitas medições com 10 configurações distintas, fixando a largura de banda, potência, CR e variando apenas o *Spreading Factor* e a distância entre receptor e transmissor, conforme apresenta a tabela 16.

**Tabela 16** – Parâmetros utilizados no transmissor ao executar o experimento externamente.

Config	Parâmetros				Distância de visada (aprox.) (m)
	SF	BW (kHz)	$P_{TX}$ (dBm)	CR	
1	7	125	14	4/5	18
2	10	125	14	4/5	18
3	7	125	14	4/5	38
4	10	125	14	4/5	38
5	7	125	14	4/5	52
6	10	125	14	4/5	52
7	7	125	14	4/5	96
8	10	125	14	4/5	96
9	7	125	14	4/5	120
10	10	125	14	4/5	120

**Figura 39** – Taxa de erro de pacotes recebidos pelo receptor SDR em função da configuração em ambiente externo conforme tabela 16.

Como mostra o gráfico da figura 39, ao utilizar SF=10 (barras amarelas, configurações pares) em pequenas distâncias (<20m) obteve-se um ótimo desempenho, chegando a apenas 1 pacote perdido dentre 1000 transmitidos. Porém, ao colocar o transmissor mais distante, a partir de 38m em diante, o desempenho foi significativamente inferior ao SF=7 (barras azuis, configurações ímpares). Considerando o último ponto com SF=10, a 120m,

somente 1 pacote foi decodificado perfeitamente. Para SF=7 (barras azuis, configurações ímpares), foi possível decodificar as mensagens perfeitamente em todos os pontos, ainda que, no último ponto, mais da metade (57%) dos pacotes foram perdidos, sendo boa parte no início da transmissão. Entre os pontos 1 e 4 (configurações 1, 3, 5 e 7) para SF=7, a PER variou entre 2% e 8%, indicando bom desempenho com este *Spreading Factor*.

Portanto, conclui-se que é possível utilizar o receptor LoRa *open-source* em SDR para capturar sinais próximos, de dispositivos que estejam transmitindo em um raio de aproximadamente 100m com visada direta. Realizando uma varredura no espectro em que LoRa opera, basta identificar os *chirps* sendo transmitidos e testar diferentes SFs e larguras de banda, criando possibilidades de analisar o tráfego de redes LoRa ou verificar vulnerabilidades de segurança, por exemplo.

## 7 Conclusão

Ao final deste trabalho, foi possível concluir com os objetivos esperados: compreender os blocos constituintes da camada física de uma tecnologia de IoT já utilizada em larga escala, em particular o esquema de modulação *Chirp Spread Spectrum* característico dela, simular o sistema LoRa em face de canais AWGN e com desvanecimento plano e avaliar um receptor SDR LoRa *open source*, possibilitando novos estudos, tanto da camada física quanto da camada de enlace.

As simulações do sistema reforçaram o que se espera da tecnologia LoRa, um bom desempenho, mesmo em ambientes bastante ruidosos e capacidade de funcionamento em SNRs menores que -20 dB em canais AWGN com SF=12. Além disso, o desempenho nos canais AWGN, Rayleigh e Rice do sistema simulado proposto apresentaram boa coerência com a literatura já publicada e usada como forma de validação dos resultados obtidos.

O receptor SDR LoRa *open source* em GNU Radio mostrou-se funcional ao utilizar um *hardware* de baixo custo (RTL-SDR), o qual foi utilizado para interceptar transmissões LoRa que foram demoduladas e decodificadas com sucesso para *Spreading Factors* de 7 a 10, decodificando perfeitamente todos os pacotes quando o transmissor e receptor se encontram próximos (12 cm) e tendo alcance significativamente bom em ambiente externo, apresentando 6% de perda de pacotes a uma distância de cerca de 90m com linha de visada direta. Para SF=11 e 12, não foi possível decodificar pacotes perfeitamente, ainda que parcialmente legíveis, necessitando melhorias no projeto. Em suma, pode-se dizer que é uma ferramenta útil para análise da camada física e decodificação dos sinais, abrindo novas oportunidades de exploração da tecnologia LoRa em ambientes controlados ou até mesmo em campo aberto em redes LoRa já implantadas.

Como sugestões de melhoria nas simulações, poderiam ser feitas investigações de desempenho do sistema em canais seletivos em frequência, já que apesar de LoRa possuir baixa taxa de dados, sua modulação espalha o sinal em frequência, o que poderia causar seletividade em frequência no canal. Além disso, implementações de esquemas de codificação poderiam ser feitas para analisar a robustez se comparado a um sistema sem codificação, pois em sistemas LoRa reais sempre haverá ao menos um CR=4/5 na carga útil e cabeçalho do pacote.

Já no que diz respeito aos experimentos práticos, a utilização de antenas de maior qualidade na transmissão e recepção possibilitaria uma melhora significativa de alcance. Também seria interessante servir-se de um SDR com capacidade de transmissão, como o LimeSDR ou HackRFOne, viabilizando o caminho contrário do proposto neste trabalho, ou seja, ao invés de transmitir com um nó comercial e receber com um SDR, transmitir com

SDR e receber com nó LoRa comercial. Outra possibilidade seria realizar a transmissão e recepção da camada física LoRa completamente utilizando SDRs.

Tratando-se do receptor SDR LoRa *open source* em GNU Radio, uma sugestão de trabalho futuro é criar novos blocos separados para cada função da cadeia de transmissão baseando-se neste projeto, de modo a ter mais flexibilidade para alterar parâmetros e possibilitar novos experimentos. Por fim, com este receptor em GNU Radio a camada MAC LoRaWAN pode ser amplamente explorada, pois é possível implementar *sniffers* e realizar testes de segurança na rede (força bruta nas chaves de ativação AppSkey e NwkSkey por exemplo), verificar o comportamento do sinal e conteúdo da mensagem quando há falha de comunicação entre o nó transmissor e *gateway*, entre outros.

## Referências

- AFISIADIS, O. Physical layer aspects of lora and full-duplex wireless transceivers. EPFL, Lausanne, p. 190, 2020. Disponível em: <<http://infoscience.epfl.ch/record/278039>>. Citado 2 vezes nas páginas 5 e 33.
- AFISIADIS, O. et al. On the error rate of the lora modulation with interference. *IEEE Transactions on Wireless Communications*, v. 19, n. 2, p. 1292–1304, 2020. Citado na página 34.
- AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, v. 17, n. 4, p. 2347–2376, 2015. Citado na página 12.
- ANATEL. *ATO nº 14448, de 04 de dezembro de 2017*. 2017. Disponível em: <<https://informacoes.anatel.gov.br/legislacao/atos-de-certificacao-de-produtos/2017/1139-ato-14448>>. Citado na página 23.
- ANATEL. *Resolução nº 705, de 21 de dezembro de 2018*. 2018. Disponível em: <<https://www.anatel.gov.br/legislacao/resolucoes/2018/1220-resolucao-705>>. Citado na página 23.
- ASHTON, K. That 'internet of things' thing. *RFID Journal*, 2009. Citado na página 12.
- AUGUSTIN, A. et al. A study of lora: Long range & low power networks for the internet of things. *Sensors*, v. 16, n. 9, 2016. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/16/9/1466>>. Citado na página 29.
- CASELLA, I. R. S. Comunicações móveis. *ESTI015-17 (EN2620) - Notas de aula*, UFABC, 2020. Citado na página 40.
- CHEN, M. et al. Narrow band internet of things. *IEEE Access*, v. 5, p. 20557–20577, 2017. Citado 2 vezes nas páginas 5 e 12.
- CROCE, D. et al. Impact of lora imperfect orthogonality: Analysis of link-level performance. *IEEE Communications Letters*, v. 22, n. 4, p. 796–799, 2018. Citado na página 28.
- CROCE, D. et al. 9 - performance of lora technology: link-level and cell-level performance. In: CHAUDHARI, B. S.; ZENNARO, M. (Ed.). *LPWAN Technologies for IoT and M2M Applications*. Academic Press, 2020. p. 181–197. ISBN 978-0-12-818880-4. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780128188804000107>>. Citado 3 vezes nas páginas 5, 30 e 31.
- DINO. *Mercado de IoT movimentará mais de 30 bilhões de dólares na América Latina até 2023*. 2021. Disponível em: <<https://www.terra.com.br/noticias/dino/mercado-de-iot-movimentara-mais-de-30-bilhoes-de-dolares-na-america-latina-ate-2023,8be0917d3e48fedd08180dae6859cd396i1iizxg.html>>. Citado na página 13.

- ELSHABRAWY, T.; ROBERT, J. Closed-form approximation of lora modulation ber performance. *IEEE Communications Letters*, v. 22, n. 9, p. 1778–1781, 2018. Citado 5 vezes nas páginas 32, 34, 35, 40 e 41.
- EZE, J. et al. Cognitive radio technology assisted vehicular ad-hoc networks (vanets): Current status, challenges, and research trends. In: *2017 23rd International Conference on Automation and Computing (ICAC)*. [S.l.: s.n.], 2017. p. 1–6. Citado na página 37.
- FABER, M. J. et al. A theoretical and experimental evaluation on the performance of lora technology. *IEEE Sensors Journal*, v. 20, n. 16, p. 9480–9489, 2020. Citado 3 vezes nas páginas 30, 32 e 34.
- FIALHO, V.; AZEVEDO, F. Wireless communication based on chirp signals for lora iot devices. v. 4, p. 5, 12 2018. Citado 2 vezes nas páginas 5 e 33.
- GLOBAL DATA. *Global IoT market will surpass the \$1 trillion mark by 2024, says GlobalData*. 2021. Disponível em: <<https://www.globaldata.com/global-iot-market-will-surpass-1-trillion-mark-2024-says-globaldata/>>. Citado na página 13.
- GNU Radio. *About GNU Radio*. 2022. Disponível em: <<https://www.gnuradio.org/about/>>. Citado na página 50.
- Great Scott Gadgets. *HackRF One*. 2014. Disponível em: <<https://greatscottgadgets.com/hackrf/one/>>. Citado na página 47.
- HARTENSTEIN, H.; LABERTEAUX, L. P. A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, v. 46, n. 6, p. 164–171, 2008. Citado na página 38.
- HAXHIBEQIRI, J. et al. A survey of lorawan for iot: From technology to application. *Sensors*, v. 18, n. 11, 2018. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/18/11/3995>>. Citado na página 21.
- HELTEC. *Heltec ESP32+LoRa Series Quick Start*. 2019. Disponível em: <[https://heltec-automation-docs.readthedocs.io/en/latest/esp32/quick\\_start.html](https://heltec-automation-docs.readthedocs.io/en/latest/esp32/quick_start.html)>. Citado 2 vezes nas páginas 15 e 45.
- HELTEC. *Heltec LoRa basic library API*. 2020. Disponível em: <[https://github.com/HelTecAutomation/Heltec\\_ESP32/blob/master/src/lora/API.md](https://github.com/HelTecAutomation/Heltec_ESP32/blob/master/src/lora/API.md)>. Citado na página 45.
- KNIGHT, M.; SEEBER, B. Decoding lora: Realizing a modern lpwan with sdr. *Proceedings of the GNU Radio Conference*, v. 1, n. 1, 2016. Disponível em: <<https://pubs.gnuradio.org/index.php/grcon/article/view/8>>. Citado 5 vezes nas páginas 15, 16, 26, 34 e 38.
- KRISHNAN, R. et al. Software defined radio (sdr) foundations, technology tradeoffs: A survey. In: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. [S.l.: s.n.], 2017. p. 2677–2682. Citado na página 36.
- Lime Microsystems. *LimeSDR*. 2020. Disponível em: <<https://limemicro.com/products/boards/limesdr/>>. Citado na página 47.

- LORA ALLIANCE. *LoRaWAN Link Layer Specification v1.0.4 (TS1-1.0.4)*. 2020. Disponível em: <[https://lora-alliance.org/resource\\_hub/lorawan-104-specification-package/](https://lora-alliance.org/resource_hub/lorawan-104-specification-package/)>. Citado 4 vezes nas páginas 18, 19, 20 e 22.
- LORA ALLIANCE. *About LoRa Alliance*. 2021. Disponível em: <<https://lora-alliance.org/about-lora-alliance/>>. Citado na página 17.
- LORA ALLIANCE. *LoRa shoots for 75% of IoT market, versus 25% for 5G; aims for stars with satellite constellation*. 2021. Disponível em: <<https://lora-alliance.org/lorawan-news/lora-shoots-for-75-of-iot-market-versus-25-for-5g-aims-for-stars-with-satellite-constellation/>>. Citado na página 13.
- LORA ALLIANCE. *RP2-1.0.3 LoRaWAN Regional Parameters*. 2021. Disponível em: <[https://lora-alliance.org/resource\\_hub/rp2-1-0-3-lorawan-regional-parameters/](https://lora-alliance.org/resource_hub/rp2-1-0-3-lorawan-regional-parameters/)>. Citado 2 vezes nas páginas 19 e 23.
- MAHESHWARAPPA, M. R.; BOWYER, M. D. J.; BRIDGES, C. P. A reconfigurable sdr architecture for parallel satellite reception. *IEEE Aerospace and Electronic Systems Magazine*, v. 33, n. 11, p. 40–53, 2018. Citado na página 38.
- MARQUET, A.; MONTAVONT, N.; PAPADOPOULOS, G. Towards an SDR implementation of LoRa: Reverse-engineering, demodulation strategies and assessment over Rayleigh channel. *Computer Communications*, Elsevier, 2020. Disponível em: <<https://hal.archives-ouvertes.fr/hal-02485052>>. Citado 4 vezes nas páginas 15, 26, 33 e 38.
- MathWorks. *Fading Channels*. 2022. Disponível em: <<https://www.mathworks.com/help/comm/ug/fading-channels.html>>. Citado 2 vezes nas páginas 5 e 41.
- MATLAB. *Fading Channels*. 2022. Disponível em: <<https://www.mathworks.com/help/comm/ug/fading-channels.html>>. Citado 2 vezes nas páginas 5 e 41.
- MEKKI, K. et al. Overview of cellular lpwan technologies for iot deployment: Sigfox, lorawan, and nb-iot. In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. [S.l.: s.n.], 2018. p. 197–202. Citado na página 13.
- MROUE, H. et al. Analytical and simulation study for lora modulation. In: *2018 25th International Conference on Telecommunications (ICT)*. [S.l.: s.n.], 2018. p. 655–659. Citado na página 34.
- RAZA, U.; KULKARNI, P.; SOORIYABANDARA, M. Low power wide area networks: An overview. *IEEE Communications Surveys Tutorials*, v. 19, n. 2, p. 855–873, 2017. Citado na página 12.
- REED, J. H. *Software Radio: A Modern Approach to Radio Engineering*. [S.l.]: Prentice Hall, 2002. ISBN 0130811580,9780130811585. Citado na página 37.
- ROBYNS, P. *gr-lora: An efficient LoRa decoder for GNU Radio*. Zenodo, 2017. Disponível em: <<https://doi.org/10.5281/zenodo.892174>>. Citado na página 52.
- ROBYNS, P. *gr-lora: An efficient LoRa decoder for GNU Radio - Test suite: decode\_long\_rtl-sdr*. 2017. Disponível em: <[https://github.com/rpp0/gr-lora/blob/master/docs/test-results/decode\\_long\\_rtl-sdr.md](https://github.com/rpp0/gr-lora/blob/master/docs/test-results/decode_long_rtl-sdr.md)>. Citado na página 55.

- ROBYNS., P. et al. A multi-channel software decoder for the lora modulation scheme. In: INSTICC. *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security - Volume 1: IoTBDS.*, [S.l.]: SciTePress, 2018. p. 41–51. ISBN 978-989-758-296-7. ISSN 2184-4976. Citado 6 vezes nas páginas 5, 15, 16, 26, 31 e 38.
- RTL-SDR. *About RTL-SDR*. 2013. Disponível em: <<https://www.rtl-sdr.com/about-rtl-sdr/>>. Citado 4 vezes nas páginas 7, 15, 46 e 47.
- SELLER, N. S. O. B. A. Low power long range transmitter. *EP2763321A1*, European Patent Application, 2014. Citado na página 26.
- SEMTECH. *AN1200.22 LoRa Modulation Basics*, Application Note, Rev 2, 2015. Citado 2 vezes nas páginas 26 e 28.
- SEMTECH. *SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver*, Datasheet, Rev 7, 2020. Citado 6 vezes nas páginas 5, 7, 26, 29, 33 e 34.
- SEMTECH. *What are LoRa and LoRaWAN?* 2020. Disponível em: <<https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>>. Citado 4 vezes nas páginas 5, 17, 19 e 26.
- SILVA, V. A. D. Modelagem computacional de canais de comunicação móvel. *Dissertação (Mestrado)*, Escola Politécnica da Universidade de São Paulo, 2004. Citado na página 41.
- STANIEC, K.; KOWAL, M. Lora performance under variable interference and heavy-multipath conditions. *Wireless Communications and Mobile Computing*, Hindawi, v. 2018, p. 6931083, Apr 2018. ISSN 1530-8669. Disponível em: <<https://doi.org/10.1155/2018/6931083>>. Citado na página 26.
- STOLNIKOV, D. *gr-osmosdr GNU Radio Blocks*. 2021. Disponível em: <<https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr>>. Citado na página 50.
- TAPPAREL, J. et al. *An Open-Source LoRa Physical Layer Prototype on GNU Radio*. 2020. Citado 3 vezes nas páginas 16, 26 e 38.
- THE THINGS NETWORKS. *Frequency Plans*. 2021. Disponível em: <<https://www.thethingsnetwork.org/docs/lorawan/frequency-plans/>>. Citado 3 vezes nas páginas 7, 24 e 25.
- VANGELISTA, L. Frequency shift chirp modulation: the lora tm modulation. *IEEE Signal Processing Letters*, PP, p. 1–1, 10 2017. Citado 3 vezes nas páginas 30, 31 e 32.
- WYGLINSKI, A. *Digital Communication Systems Engineering with Software-Defined Radio*. [S.l.]: Artech House, 2013. (Mobile Communications). ISBN 1608075257,9781608075256. Citado 2 vezes nas páginas 36 e 37.
- WYGLINSKI MAZIAR NEKOVEE, T. H. A. M. *Cognitive Radio Communications and Networks: Principles and Practice*. [S.l.: s.n.], 2009. ISBN 0123747155,9780080879321,9780123747150. Citado 2 vezes nas páginas 5 e 36.
- ZAYAS, A. D.; MERINO, P. The 3gpp nb-iot system architecture for the internet of things. In: *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. [S.l.: s.n.], 2017. p. 277–282. Citado na página 13.