

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

VITOR CARVALHO - 00289783

**ANÁLISE DO PROTOCOLO LORA
USANDO GNU RADIO**

Porto Alegre
2024

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

VITOR CARVALHO - 00289783

**ANÁLISE DO PROTOCOLO LORA
USANDO GNU RADIO**

Trabalho de Conclusão de Curso submetido à
COMGRAD/CCA da UFRGS como parte dos requi-
sitos para a obtenção do título de *Bacharel em Enge-
nharia de Controle e Automação*.

Orientadores:
Prof. Dr. Marcelo Götz
Prof. Dr. Ivan Müller

Porto Alegre
2024

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

VITOR CARVALHO - 00289783

**ANÁLISE DO PROTOCOLO LORA
USANDO GNU RADIO**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Atividade de *Trabalho de Conclusão de Curso CCA - II* e aprovado em sua forma final pelos Orientadores e Banca Examinadora abaixo.

Orientadores:

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn – Paderborn, Alemanha

Prof. Dr. Ivan Müller, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul –
Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn – Paderborn, Alemanha

Prof. Dr. Valner Brusamarello, UFRGS

Doutor pela Universidade Federal de Santa Catarina - Florianópolis, Brasil

Me. Max Feldman, UFRGS

Mestre pela Universidade Federal do Rio Grande do Sul - Porto Alegre, Brasil.

Alceu Heinke Frigeri

Coordenador de Curso

Eng. de Controle e Automação

Porto Alegre

Fevereiro - 2024

DEDICATÓRIA

Dedico esse trabalho aos meus pais, Jones e Loreci, que tanto me apoiaram ao longo de toda minha trajetória. Dedico igualmente a minha irmã, Sarah, que sempre me serviu de modelo.

*"Wir müssen wissen, wir werden wissen."
- David Hilbert*

AGRADECIMENTOS

Agradeço a Universidade Federal do Rio Grande do Sul, UFRGS, pela oportunidade de realização de estudos de maneira gratuita e de qualidade. Estendo meu agradecimento ao meu orientador, professor Marcelo, por todo o apoio neste trabalho e em toda a jornada no curso. De igual forma, sou grato ao professor Ivan, coorientador deste trabalho, por fornecer auxílio em dúvidas técnicas que ocorreram ao longo do caminho. Agradeço também à minha banca que investiu seu precioso tempo para ler este trabalho.

Sou grato também a todos os colegas de curso que participaram da minha jornada até esse momento. Aos professores que, exercendo sua profissão com excelência, possibilitaram que eu chegasse até aqui com uma base sólida dos conceitos de engenharia.

Agradeço a todos os meus amigos, em especial ao Lucas Esteves, sem o qual muitos trabalhos teriam sido tediosos.

Agradeço a minha namorada Mariana, pelo auxílio na revisão deste trabalho e pelos gratificantes momentos de descontração.

Agradeço aos meus pais, Jones e Loreci. Se cheguei até aqui, foi muito pelo auxílio e incentivo de vocês. Espero que saibam que tudo que faço sempre terá a contribuição de vocês. A minha querida irmã, por sempre acreditar na minha capacidade e sempre me motivar a ir além.

Por fim, agradeço ao Henrique Jongh, por ter me apresentado o maravilhoso (porém caótico) mundo do *open-source*.

RESUMO

O protocolo LoRa destaca-se por sua aplicabilidade em sensoriamento remoto e na Internet das Coisas. Todavia, esse protocolo é proprietário e implementado diretamente em *chips* desenvolvidos pela *Semtech*, o que pode dificultar análises de desempenho. Nesse contexto, surge a utilização de rádios definidos por *software* (SDR). Esse tipo de dispositivo de comunicação executa em um computador grande parte do processamento que é normalmente efetuado em *hardware*. Assim, este estudo verifica a robustez do protocolo LoRa em relação ao ruído branco gaussiano, usando uma implementação do protocolo em SDR desenvolvida para o GNU Radio por (TAPPAREL et al., 2020). Para tal, o trabalho revisa conceitos fundamentais de comunicação de radiofrequência, especificações do protocolo e ferramentas usadas, desenvolvendo algoritmos supervisórios para automação de simulações e experimentos. Os resultados confirmam a eficácia da modulação LoRa em relações sinais-ruído negativas e a compatibilidade da implementação SDR LoRa com chips comerciais, embora revelem divergências com a implementação proprietária.

Palavras-chave: GNU Radio, LoRa, Protocolos, Rádio Definido por *software*, SDR

ABSTRACT

The LoRa protocol stands out for its applicability in remote sensing and the Internet of Things. However, this protocol is proprietary and implemented directly on chips developed by Semtech, which can hinder performance analysis. In this scenario, the use of Software Defined Radios (SDR) become apparent. This type of communication device performs a significant portion of the processing, which is typically done in hardware, on a computer. Therefore, this study verifies the robustness of the LoRa protocol in a Gaussian noise environment, using an implementation of the protocol in SDR developed for GNU Radio by (TAPPAREL et al., 2020). To achieve this, this paper revisits fundamental concepts of radiofrequency communication, protocol specifications, and tools used, developing supervisory algorithms for the automation of simulations and experiments. The results confirm the effectiveness of LoRa modulation in negative signal-to-noise ratios and the compatibility of the SDR LoRa implementation with commercial chips, although they reveal discrepancies with the proprietary implementation.

Palavras-chave: GNU Radio, LoRa, Protocols, SDR, Software Defined Radio

LISTA DE ILUSTRAÇÕES

1	Espectro eletromagnético no contexto de comunicações.....	16
2	Diagrama de blocos de um SDR genérico.....	18
3	Representação conceitual básica da modulação LoRa	19
4	Estrutura geral dos pacotes LoRa.....	20
5	Representação dos quadros de <i>Up-Link</i> e <i>Down-Link</i>	21
6	Visão sistêmica de uma rede LoRaWAN. Apresentando o seu modelo em camadas, onde é possível observar duas camadas de aplicação possíveis. A primeira, onde a interação é direta entre os <i>devices</i> e <i>gateways</i> é o foco do presente trabalho.	21
7	Fluxograma básico para visualização de uma senoide construído no GRC.	23
8	Visão geral dos pinos da placa WiFi LoRa 32 (V2).	25
9	Diagrama de blocos do RTL-SDR.	26
10	Vistas superiores e inferiores do Lime SDR Mini com destaque a algumas interfaces e componentes	27
11	Combinador/divisor de sinais Wilkinson J2065 da JBM	27
12	Equipamentos utilizados para medir a potência de transmissão.....	28
13	Atenuadores variáveis conectados em série.....	29
14	Fluxograma com o algoritmo proposto para a simulação.....	31
15	Estrutura do arquivo de configurações.	32
16	Fluxograma do GNU Radio utilizado para as simulações.....	33
17	Fluxograma com o algoritmo proposto para prática SDR RX - LoRa <i>chip Tx</i>	34
18	Fluxograma do GNU Radio utilizado para a prática SDR RX - LoRa <i>chip Tx</i>	36
19	Fluxograma do GNU Radio para inserir ruído AWGN no sistema.....	37
20	Aplicação interativa para visualização inicial dos resultados.	38
21	Impacto do número de simulações na suavidade da curva.	40
22	Configuração proposta para o experimento	41
23	Medida da potência percebida na saída do atenuador de 30 dB quando somente o Lime SDR Mini está transmitindo a 10 dBm.....	42
24	Medida da potência percebida na saída do atenuador de 30 dB quando somente o transmissor LoRa está operando a 14 dBm.....	42
25	Gráfico de cascata somente com o sinal LoRa habilitado	43
26	Gráfico de cascata com o sinal LoRa e perturbação habilitados e SNR de 0 dB	43
27	Circuito de RF montado para o experimento prático.	45
28	Resultados da simulação por SF, variando BW e CR	47

29	Comparação da curva média de cada SF com a curva teórica	48
30	Resultados da simulação com e sem a interferência multi caminhos . . .	49
31	Resultados da simulação com <i>hard</i> e <i>soft decoding</i>	50
32	Resultados brutos do experimento prático.	53
33	Curvas ajustadas para os dados do experimento prático	54
34	Curvas $BER \times SNR$ para as aproximações teórica, da simulação e dos dados práticos	55

LISTA DE TABELAS

1	Resumo dos parâmetros do protocolo variados ao longo da simulação e experimentos.....	30
2	Caminhos simulados e valores utilizados para a interferência multicaminho.....	30
3	Mensagens utilizadas na comunicação serial	35
4	Resumo dos parâmetros da simulação.....	39
5	Resumo dos parâmetros da atividade prática	44
6	Resumo dos parâmetros da simulação	46
7	Matriz com o erro quadrático médio entre cada curva de erro de <i>bit</i> para cada fator de espalhamento com largura de banda 250 kHz e fator de codificação variável, considerando efeito de caminhos múltiplos.....	50
8	Matriz com o erro quadrático médio entre cada curva de erro de <i>bit</i> para cada fator de espalhamento com largura de banda 250 kHz e fator de codificação variável, considerando efeito da decisão de <i>bit</i>	51
9	Resumo dos parâmetros da atividade prática	52
10	Parâmetros das curvas sigmóides ajustadas aos dados do experimento.	53

LISTA DE ABREVIATURAS

API	<i>Application Programming Interface</i>
ANATEL	Agência Nacional de Telecomunicações
AM	Amplitude Modulada
AWGN	<i>Additive White Gaussian Noise</i>
BER	<i>Bit Error Rate</i>
bps	Bits por segundo
BW	<i>Bandwidth</i>
CR	<i>Coding Rate</i>
CRC	<i>Cyclic Redundancy Check</i>
CSS	<i>Chirp Spread Spectrum</i>
FIR	<i>Finite Impulse Response</i>
FM	Frequência Modulada
FPGA	<i>Field Programmable Gate Arrays</i>
GRC	<i>GNU Radio Companion</i>
IoT	<i>Internet of Things</i>
ISM	<i>Industrial, Scientific and Medical</i>
ITU	<i>International Telecommunication Union</i>
LACOM	Laboratório de Comunicações
LNA	<i>Low Noise Amplifier</i>
LoRa	<i>Long Range</i>
LoRaWAN	<i>Long Range Wide Area Network</i>
MAC	<i>Medium Access Control</i>
MER	<i>Message Error Rate</i>

OSI	<i>Open Systems Interconnection</i>
PHY	<i>Physical Layer</i>
RF	Rádio Frequência
Rx	Receptor
SDR	<i>Software Defined Radio</i>
SF	<i>Spreading Factor</i>
SNR	<i>Signal-to-noise ratio</i>
SPI	<i>Serial Peripheral Interface</i>
Tx	Transmissor
USRP	<i>Universal Software Radio Peripheral</i>
UFRGS	Universidade Federal do Rio Grande do Sul

SUMÁRIO

1	INTRODUÇÃO	14
2	REVISÃO BIBLIOGRÁFICA	15
2.1	Compreensão dos princípios fundamentais das comunicações sem fio ..	15
2.2	Análise do conceito e aplicações do rádio definido por <i>software</i> (SDR).	17
2.3	LoRa e LoRaWAN	19
2.3.1	LoRa	19
2.3.2	LoRaWAN	21
2.4	GNU Radio	22
3	MATERIAIS E MÉTODOS	24
3.1	Objetivos e Pré-requisitos	24
3.2	Análises Propostas	29
3.3	Desenvolvimento	30
3.3.1	Algoritmo de simulação	31
3.3.2	Fluxograma GNU Radio da Simulação	32
3.3.3	Algoritmo para a prática com SDR Rx	34
3.3.4	Fluxograma GNU Radio SDR RX	35
3.3.5	Fluxograma GNU Radio Ruído AWGN	36
3.3.6	Visualização de dados	37
3.4	Execução	38
3.4.1	Simulações	38
3.4.2	Prática	40
4	RESULTADOS E DISCUSSÕES	46
4.1	Resultados da Simulação	46
4.1.1	Simulação Caminho direto	46
4.1.2	Simulação Caminhos múltiplos	48
4.1.3	Simulação <i>hard decoding</i>	50
4.1.4	Considerações Gerais dos Resultados da Simulação	51
4.2	Resultados do Experimento	52
4.2.1	Considerações Gerais dos Resultados do Experimento	56
5	CONCLUSÃO	57
	REFERÊNCIAS	59

1 INTRODUÇÃO

A comunicação é essencial na configuração atual da Indústria 4.0 assim como no seu futuro (ACETO; PERSICO; PESCAPÉ, 2019). Nesse contexto, o desenvolvimento de novas tecnologias e estudos nessa área são de grande relevância. Levando em conta esse cenário, surge o protocolo *Long Range* (LoRa). Esse protocolo de comunicação sem fio desenvolvido pela Semtech tem como objetivo a comunicação de qualidade a longa distância com um baixo consumo energético (SEMTECH, 2023d). Por esse motivo, ele tem sido utilizado em aplicações de Internet das Coisas (IoT - *Internet of Things*, em inglês), que variam desde sensoriamento remoto em zonas agrícolas (AHMED et al., 2022) até o uso em ambientes industriais (RIZZI et al., 2017).

Assim, a análise do desempenho desse protocolo em diferentes cenários pode auxiliar a melhor definir as suas aplicações e limitações. Dessa forma, o presente trabalho busca justamente efetuar essa verificação, considerando um ambiente com relações de sinal-ruído variáveis (SNR), com foco principal em configurações de SNR negativo. Essa definição parte do conceito de que o protocolo LoRa apresenta uma grande capacidade de atuar nesse tipo de ambiente (VANGELISTA, 2017). Para isso, será utilizada uma implementação em um *Software Defined Radio* (SDR) para simular diferentes formas de interferência na comunicação sem fio, considerando um modelo de canal com Ruído Branco Aditivo (AWGN — *Additive White Gaussian Noise*, em inglês).

Um SDR é um sistema de comunicação em que componentes tradicionalmente implementados em *hardware*, tais como misturadores, filtros, moduladores e demoduladores, são virtualizados e implementados por meio de *software* em um computador ou processador embarcado (GARG, 2007). Esta abordagem oferece significativa flexibilidade, permitindo que o mesmo *hardware* se adapte a diferentes padrões e faixas de frequência mediante ajustes no *software*, otimizando assim a adaptabilidade e escalabilidade do sistema de comunicação. De igual forma, como grande parte do processamento é feito de maneira digital em um computador, é possível simular diversos cenários de operação que é o objetivo do presente trabalho. Para efetuar essa análise será utilizado o *software* GNU Radio.

O GNU Radio é uma ferramenta de desenvolvimento de código aberto que fornece blocos de processamento de sinais para implementar SDRs e manipulação de sinais em geral (PROJECT, 2023a). Ele permite que os usuários criem, simulem e testem fluxogramas de processamento de sinal de maneira rápida e visual, para aplicações em tempo real ou não.

O presente trabalho está organizado em cinco capítulos sendo o primeiro esta introdução. O Capítulo 2 traz uma revisão bibliográfica de forma a embasar conceitualmente futuras etapas do desenvolvimento, bem como justificar a viabilidade e relevância do estudo proposto. O Capítulo 3 discute a abordagem adotada para o problema em análise bem como os materiais utilizados. Por sua vez, o Capítulo 4 apresenta os resultados obtidos e é seguido pelo Capítulo 5 onde os resultados do trabalho são ponderados assim como a possibilidade de futuras abordagens e melhorias.

2 REVISÃO BIBLIOGRÁFICA

A busca contínua por sistemas de comunicação robustos e eficientes, principalmente no contexto da Indústria 4.0 e da *Internet of Things* (IoT), tem desencadeado uma série de investigações em diversos protocolos e tecnologias emergentes. Em especial, o campo das comunicações sem fio tem passado por evoluções significativas nas últimas décadas, enfrentando desafios que vão desde a otimização do espectro eletromagnético até a garantia de transmissões seguras e estáveis em ambientes adversos, como os industriais. A revisão bibliográfica subsequente busca contextualizar e fornecer um panorama acerca dos fundamentos da comunicação sem fio, com foco especial na análise do protocolo LoRa, da definição de SDRs e da plataforma GNU Radio.

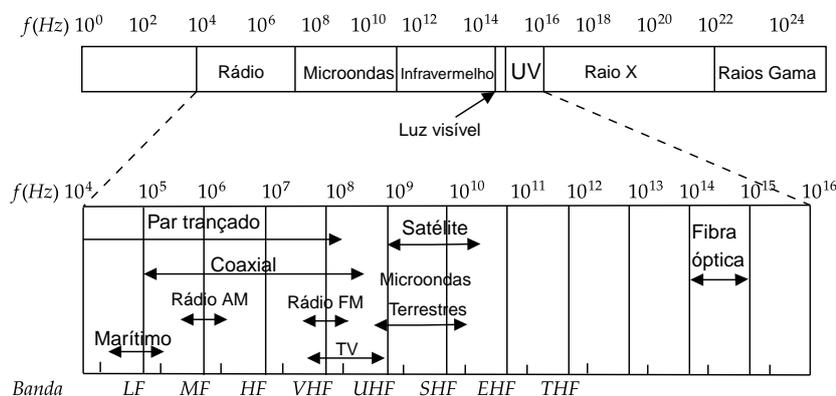
Buscando facilitar a interpretação para o leitor, este capítulo foi separado em seções da revisão bibliográfica proposta.

2.1 COMPREENSÃO DOS PRINCÍPIOS FUNDAMENTAIS DAS COMUNICAÇÕES SEM FIO

A presente seção busca detalhar os principais aspectos da comunicação sem fio. O detalhamento proposto irá abordar desde os conceitos básicos do espectro eletromagnético e modulação até os desafios de propagação e interferência, oferecendo uma visão abrangente do intrincado mundo da transmissão de dados através do ar. Naturalmente, como cada um desses princípios constitui uma área de estudo de profundidade considerável, somente aspectos-chaves, necessários para a construção de uma visão sistêmica consistente das comunicações sem fio, serão definidos. De igual forma, sempre que pertinente, o escopo será devidamente limitado de forma a focar em definições relevantes para o estudo proposto.

Um elemento fundamental das comunicações por Rádio Frequência (RF) é a definição do espectro eletromagnético. Do ponto de vista teórico, a contribuição fundamental para isso foram as equações de Maxwell, propostas pelo físico escocês James Clerk Maxwell (1831-1879) e observadas pelo físico alemão Heinrich Hertz (1857-1894). Sobre a ótica de comunicação em RF, o espectro eletromagnético é especificamente relevante para definir a faixa de operação de determinada aplicação. A Figura 1, apresenta o espectro eletromagnético e como ele é utilizado na comunicação:

Figura 1: Espectro eletromagnético no contexto de comunicações.



Fonte: Tanenbaum e Feamster (2021)

Nota: Na parte superior da imagem, observa-se a faixa eletromagnética como um todo. Na parte inferior é dado o destaque à faixa de frequências utilizadas em meios de comunicação.

Os nomes de cada faixa são definidos pela *International Telecommunication Union* (ITU). A ITU (mais especificadamente seu setor de comunicações de RF, a ITU-R) é responsável pela alocação de espectro, padronização, desenvolvimento, apoio a políticas e regulamentação do espectro (ITU, 2023). A regulamentação e controle do espectro eletromagnético é fundamental, uma vez que é um recurso compartilhado por todos. No âmbito nacional, a Agência Nacional de Telecomunicações (ANATEL) executa essa tarefa (ANATEL, 2023a).

O protocolo LoRa, opera numa faixa ISM, acrônimo de *Industrial, Scientific and Medical* (Industrial, Científica e Médica). A faixa ISM é essencialmente de livre uso, de acordo com a ITU. No país, de acordo com a Resolução nº 680/2017 a faixa ISM é reservada de 902-928 MHz (ANATEL, 2023b), sendo a frequência de 915 MHz a mais comum atribuída a redes LoRa no país.

Outro aspecto fundamental da comunicação sem fio é a modulação. Esse processo consiste em variar uma ou mais propriedades de uma onda portadora, geralmente uma onda senoidal, de acordo com a informação ou sinal que se deseja transmitir (PROAKIS; SALEHI, 2001). Existem diversos tipos de modulação dentre os quais os mais conhecidos são a modulação em amplitude (AM) e a modulação em frequência (FM) devido ao seu uso pelas comunicações via rádio. O protocolo LoRa utiliza uma modulação específica, baseada no esquema de modulação denominado *Chirp Spread Spectrum* (CSS) (SEMTECH, 2023c), a qual será explicada em detalhes posteriormente.

Um elemento fundamental que será variado de forma a possibilitar a análise da robustez do protocolo LoRa, é o canal. O canal em comunicações sem fio é frequentemente definido como o meio físico pelo qual as ondas eletromagnéticas se propagam (PROAKIS; SALEHI, 2001), normalmente a atmosfera. Por esse motivo a comunicação sem fio apresenta grande adesão, mas também é tão suscetível ao ruído e a corrupção de sinais (MCHENRY; ROBERTSON; MATHESON, 2015).

Dois principais modelos de canal na comunicação por rádio frequência são normalmente utilizadas em simulação. São eles: o modelo AWGN (*Additive White Gaussian Noise*, Ruído Branco Aditivo, em português) e o modelo Rayleigh (PROAKIS; SALEHI, 2001).

No primeiro modelo de canal, a potência ao longo de toda a faixa de frequências é limitada seguindo uma distribuição gaussiana. Essa aproximação é a mais simples de todas e é a mais utilizada para análises iniciais de relação sinal-ruído (SNR).

Por sua vez, o modelo Rayleigh insere maior grau de complexidade ao meio físico, uma vez que considera a presença de múltiplas trajetórias de uma maneira probabilística (LUO; ZHANG, 2007). Tal fenômeno é frequentemente observado em contextos urbanos ou edificações, onde a onda radiada é sujeita a reflexões diversas em estruturas como edifícios, vegetações ou veículos, antes de ser captada pelo receptor. Estas diversas reflexões, ao confluírem no ponto de recepção, podem combinar-se de maneira construtiva ou destrutiva, culminando em flutuações rápidas e imprevisíveis na amplitude do sinal recebido. Esse fenômeno de superposição de reflexões pode acarretar em episódios de atenuação súbita do sinal.

Neste trabalho, será utilizado o modelo de canal AWGN. Além disso, vale notar que sua implementação no GNU Radio apresenta a possibilidade de efetuar simulação de interferência de caminhos múltiplos por meio de múltiplas entradas em um filtro de resposta ao impulso finita (filtro FIR).

2.2 ANÁLISE DO CONCEITO E APLICAÇÕES DO RÁDIO DEFINIDO POR SOFTWARE (SDR)

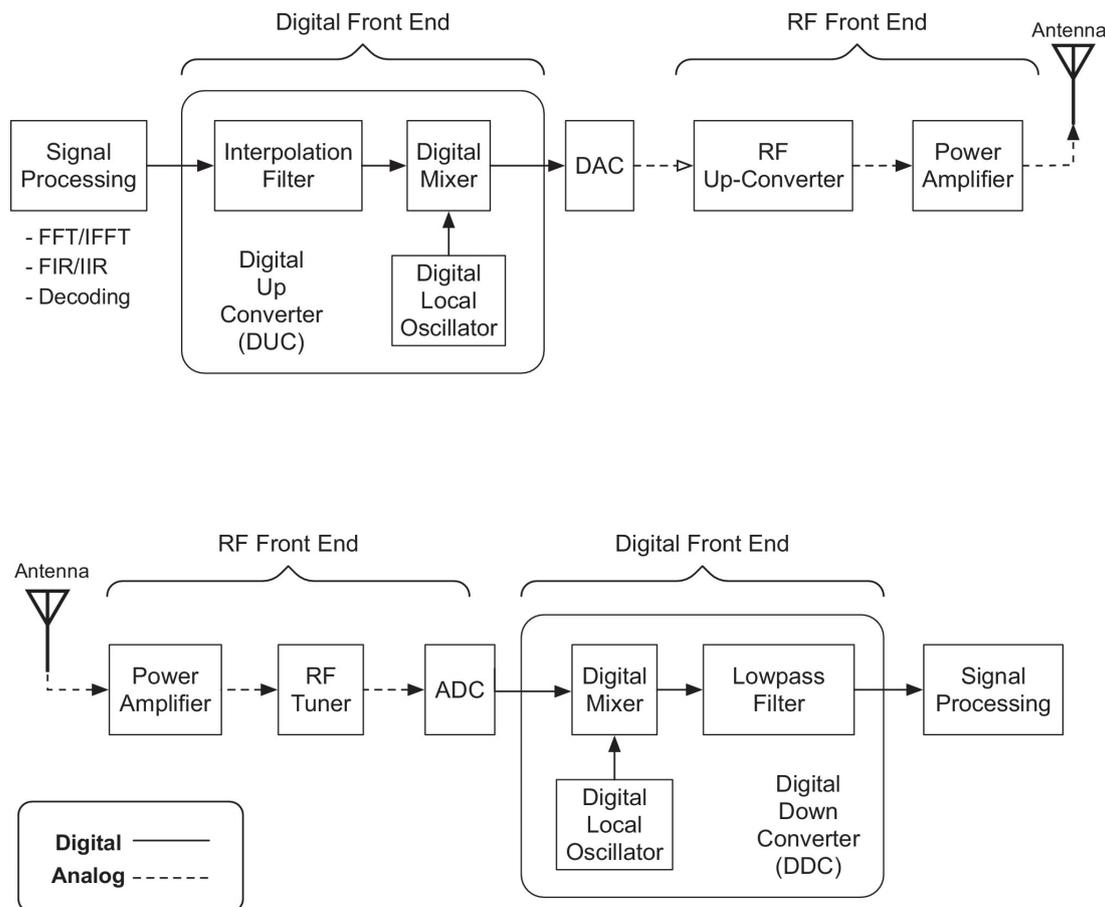
Assim como a maioria das inovações tecnológicas na área de comunicação, os rádios definidos por *software* surgiram em um contexto militar (PAWLAN, 2015). A primeira versão de um SDR foi o SPEAKeasy das forças armadas dos Estados Unidos no final do século passado (COOK; BONSER, 1999). O objetivo era desenvolver um dispositivo capaz de atuar em uma grande faixa de frequências (2 MHz aos 2 GHz), podendo ser facilmente reprogramado para operar em outra faixa (LACKEY; UPMAL, 1995). Atualmente, o Exército Brasileiro (desde 2012) elabora sua própria versão de SDR militar no projeto RDS Defesa (FILHO; GALDINO; MOURA, 2017).

Um SDR pode ser definido como um transmissor e/ou receptor de rádio que utiliza uma tecnologia que permite que os parâmetros operacionais de RF, incluindo, mas não limitado à faixa de frequência, tipo de modulação ou potência de saída, sejam definidos ou alterados por *software* (ITU-R, 2009). Além disso, são excluídas dessa definição rádios onde mudanças operacionais ocorram de acordo com uma especificação ou padrão de sistema. Uma definição mais informal descreve um SDR como um dispositivo de comunicação sem fio onde parte do processamento, que é naturalmente efetuado em *hardware*, é efetuado em *software* de forma a possibilitar maior flexibilidade (MICRO, 2023).

Ainda assim, um SDR mantém uma interface puramente analógica para transmitir e receber dados, de acordo com a sua configuração. A Figura 2 apresenta o diagrama básico de um transmissor SDR e de um receptor SDR (na parte superior e inferior da imagem, respectivamente).

Considerando o diagrama de blocos da Figura 2, percebe-se que alguns elementos-chaves do *hardware* de RF se mantém, como os amplificadores, filtros, osciladores locais, entre outros. Todavia, os componentes fundamentais para um SDR são os conversores analógico digital (A/D) e digital para analógico (D/A), os quais realizam a interface com o computador dedicado que executa uma rotina em *software*. A resolução em *bits* desses conversores é um dos elementos significativos que faz com que um *hardware* equivalente com um conversor de resolução superior tenha um custo elevado. Essa rotina efetua a

Figura 2: Diagrama de blocos de um SDR genérico.



Fonte: Akeela e Dezfouli (2018)

demodulação ou modulação do sinal de forma a obter a informação contida nos dados recebidos. Como o processo de decodificação é feito em *software* isso possibilita que o mesmo *hardware* seja capaz de capturar a informação de diferentes formas de transmissão. É essa flexibilidade que torna os SDRs tão versáteis (AKEELA; DEZFOULI, 2018).

Um exemplo de *hardware* SDR disponível comercialmente é HackRFOne. Esse dispositivo é capaz de operar entre 1 MHz e 6 GHz como receptor e transmissor (GADGETS, 2023) e apresenta uma das maiores faixas de atuação entre os *hardwares* SDR disponíveis no mercado. Um outro exemplo é o RTL-SDR que utiliza o chip RTL2832U da Realtek. O RTL-SDR, entretanto, atua somente como receptor numa faixa de 22 MHz à 2,2 GHz.

Atualmente, pesquisas são conduzidas para analisar a viabilidade de rádios definidos por *software* no contexto de IoT (CHEN et al., 2016) e rádios cognitivos (TALUKDAR et al., 2017). É interessante notar que o uso de rádios cognitivos nas mais diversas aplicações é uma tendência (AL-SUDANI; THABIT; DALVEREN, 2022). Afinal, uma rede de rádios cognitivos, conforme definido pela ITU (ITU-R, 2009), é capaz de analisar o espectro ao seu redor e alterar seus parâmetros de transmissão e recepção de forma a otimizar o uso do espectro eletromagnético. Essa variação e análise de parâmetros durante a operação exige o uso de processamento computacional e conseqüentemente está frequentemente atrelada a tecnologia de SDRs (NEWMAN; EVANS; WYGLINSKI, 2010).

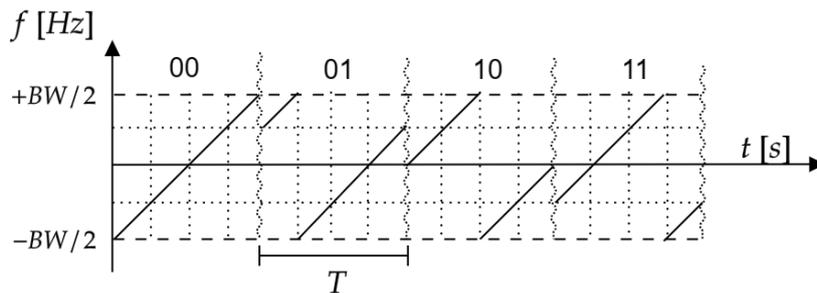
2.3 LORA E LORAWAN

Quando se fala em LoRa é possível dividir o tópico em duas partes: a parte LoRa e a parte LoRaWAN. A parte LoRa implementa a camada física de um protocolo dentro do modelo *Open Systems Interconnection* (OSI) (ORACLE, 2023), ou seja, é responsável pela codificação e transmissão dos *bits*. Essa parte do protocolo é fechada, desenvolvida pela empresa norte-americana Semtech (SEMTECH, 2023d). Por sua vez, a parte LoRaWAN é focada nos níveis superiores do modelo OSI, atuando nas camadas de enlace e rede. Nas subseções a seguir serão apresentados alguns conceitos chaves de cada subdivisão.

2.3.1 LoRa

A Semtech desenvolve os *chips* LoRa que implementam o meio físico (PHY) deste protocolo e que efetuam uma modulação baseada em *Chirp Spread Spectrum* (CSS) proprietária (SFORZA, 2013). Essa abordagem é considerada uma técnica de modulação digital onde o salto entre um *chirp* e outro, bem como sua duração, caracteriza o símbolo transmitido. A Figura 3 apresenta a ideia fundamental dessa modulação.

Figura 3: Representação conceitual básica da modulação LoRa



Fonte: adaptado de Sforza (2013)

Considerando a Figura 3, é possível observar que dois parâmetros são fundamentais na modulação CSS. São eles: o fator de espalhamento (*Spreading Factor* em inglês) e a largura de banda (*Bandwidth*). O *Spreading Factor* (SF) está relacionado ao número de *bits* por símbolo. Matematicamente, essa relação é representada por 2^{SF} (SEMTECH, 2023a). No caso do exemplo da Figura 3, o valor de SF é 2. Na prática, quanto maior o valor de SF, maior a duração de um *chirp* no tempo, mas também maior é número de símbolos possíveis. Os valores mais comuns de fator de espalhamento utilizados mais utilizado na modulação LoRa variam entre 7 e 12.

Por sua vez, a largura de banda (BW) define o quanto os *chirps* LoRa estão espalhados ao redor da frequência central. Os valores típicos para a BW são: 125 kHz, 250 kHz e 500 kHz. A combinação dos valores de SF e BW resulta na taxa de transmissão (R_b) em *bits* por segundo (bps) apresentada na Equação 1. Por sua vez, o período de um símbolo (T) pode ser calculado pela Equação 2.

$$R_b = SF \cdot \frac{BW}{2^{SF}} \text{ [bps]} \quad (1)$$

$$T = \frac{2^{SF}}{BW} [s] \quad (2)$$

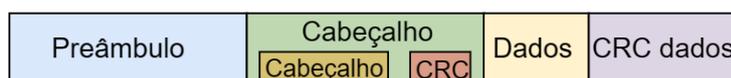
Analisando as equações 1 e 2, percebe-se claramente que o valor de SF influencia significativamente na taxa de *bits* por segundo. Por atuar como expoente do denominador da equação, o aumento do SF implica na diminuição significativa dessa taxa. Vale notar que o valor de R_b não pode ser considerado o valor de *bits* úteis enviados no processo de comunicação. Isso porque, a modulação LoRa, assim como muitos protocolos, além de todos os cabeçalhos envolvidos, utiliza uma Taxa de Codificação (*Coding Rate* em inglês) (LIBELIUM, 2023). Os valores de *Coding Rate* (CR) assumem os valores de 1 a 4. Um CR de valor 1 denota que codificação utilizada é de 4/5. Ou seja, 4 *bits* de dados são convertidos em 5 *bits* que então são enviados para a transmissão (TANENBAUM; FEAMSTER, 2021). A mesma lógica vale para os demais valores de CR, por exemplo, $CR = 4 \implies 4/8$. O CR de 4/5 é o mais comum, todavia aumentar esse valor aumenta a robustez do protocolo. Com isso é possível reescrever a equação 1 levando em conta essa possibilidade conforme a Equação 3. Nessa Equação, o valor CR é um valor entre 1 e 4.

$$R_{bCR} = SF \cdot \frac{4}{4 + CR} \cdot \frac{BW}{2^{SF}} [bps] \quad (3)$$

Embora a camada física do protocolo seja patenteada, inúmeras tentativas de engenharia reversa foram feitas. Dessas, vale a pena destacar uma implementação em SDR utilizando o GNU Radio desenvolvida por (TAPPAREL et al., 2020). O trabalho desenvolvido por (TAPPAREL et al., 2020) mostrou que essa implementação é consistente. Dessa forma, com este trabalho, busca-se também validar os resultados de (TAPPAREL et al., 2020) assim como analisar em maior detalhe o impacto da variação de parâmetros nesse protocolo e como isso interfere na sua robustez frente a um canal AWGN.

A estrutura de um pacote LoRa inicia com um preâmbulo que tem um comprimento variando entre 10 a $2^{16} - 1$ símbolos (ELSHABRAWY; ROBERT, 2018). Após o preâmbulo, um cabeçalho opcional pode ser inserido. Ele contém informações sobre o tamanho da carga útil e informações sobre configuração do LoRa utilizada, como o CR em uso. É interessante destacar que essa parte é sempre codificada com CR de 4. No final do quadro, um Código de Redundância Cíclica (CRC) opcional também pode ser incluído. A Figura 4 apresenta a estrutura geral dos pacotes LoRa.

Figura 4: Estrutura geral dos pacotes LoRa

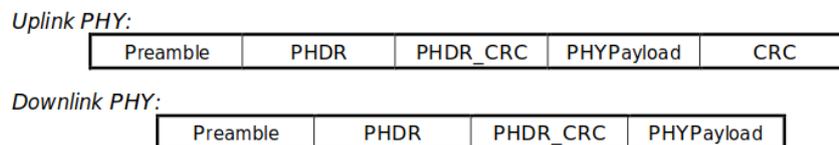


Fonte: do Autor

Ainda sobre os pacotes LoRa, eles se dividem em dois modelos, são eles os *Up-Links* e *Down-Links* (GTA-UFRJ, 2023). Um *Up-Link* refere-se à transmissão de dados do dispositivo final (como um sensor ou atuador) para o servidor. Essa informação pode ser capturada por qualquer elemento que realize a interface com o servidor (*gateway*). A camada superior do protocolo deve tratar essa redundância. Um exemplo desse tipo de transmissão é um sensor se comunicando com um dispositivo supervisor. Por sua vez, um *Down-Link* é exatamente o contrário. Consiste em um dispositivo *gateway* enviando

dados para um atuador, por exemplo. A Figura 5 apresenta a estrutura dos quadros de *Up-Link* e *Down-Link*.

Figura 5: Representação dos quadros de *Up-Link* e *Down-Link*.



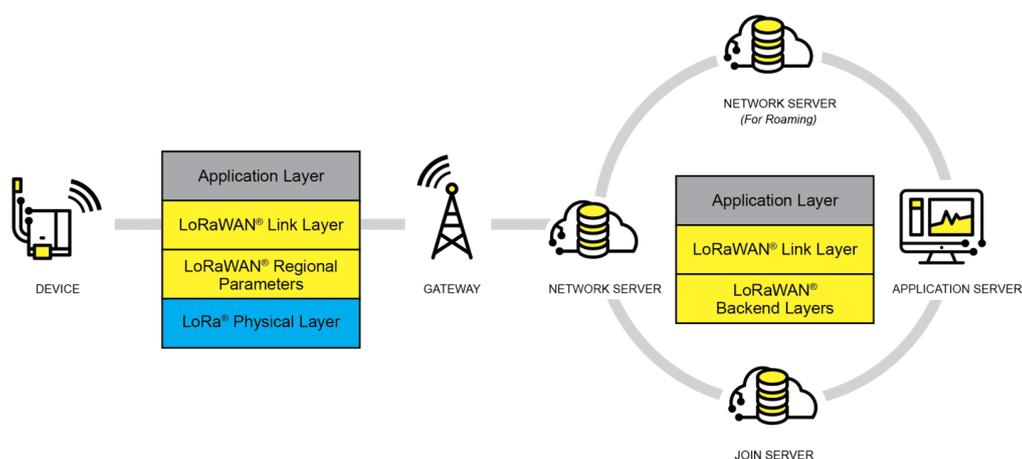
Fonte: Adaptado de GTA-UFRJ (2023)

Como é possível observar na Figura 5, esses quadros são similares, exceto pela presença do campo de verificação por *Cyclic Redundancy Check* (CRC). Vale notar que o polinômio de CRC utilizado é de normalmente 16 *bits* (ALLIANCE, 2020). Mais detalhes sobre cada campo podem ser obtidos na documentação oficial da LoRaWAN (ALLIANCE, 2020).

2.3.2 LoRaWAN

O foco do presente trabalho será na camada física do padrão LoRaWAN, ou seja, no protocolo LoRa. Todavia, uma noção geral de aplicação desse protocolo no contexto de redes ajuda a validar sua relevância. A LoRaWAN (*Long Range Wide Area Network*) não é patenteada e é mantida pela LoRa Alliance (ALLIANCE, 2023). Essa organização define, essencialmente, as outras camadas - exceto a camada de aplicação - do protocolo. Uma rede LoRaWAN utiliza as especificações definidas pela LoRa Alliance para definir um tipo de *Low Power Wide Area Network* (LPWAN). As características mais importantes de uma rede LoRaWAN são: o baixo consumo, longo alcance e bidirecionalidade da comunicação. Suas duas primeiras características são justamente provenientes do uso da modulação LoRa. A Figura 6 apresenta uma visão sistêmica do protocolo LoRaWAN.

Figura 6: Visão sistêmica de uma rede LoRaWAN. Apresentando o seu modelo em camadas, onde é possível observar duas camadas de aplicação possíveis. A primeira, onde é a interação é direta entre os *devices* e *gateways* é o foco do presente trabalho.



Fonte: Alliance (2023)

Os dois tipos de equipamentos em uma rede LoRaWAN que atuam diretamente com a camada física, são os *gateways* e os dispositivos finais (*End-Nodes*) (ALLIANCE, 2020). Tais dispositivos finais são, essencialmente, os sensores ou atuadores que se encontram espalhados em uma grande área de forma a coletar e executar determinadas tarefas de acordo com a característica desses instrumentos. Esses elementos devem possuir um *chip* LoRa ou outro mecanismo capaz de se comunicar com o(s) *gateway(s)*.

Por sua vez, os *gateways* possibilitam a comunicação entre os sensores e os servidores. Eles essencialmente recebem instruções dos servidores aos quais são conectados e são responsáveis por transmiti-las para os dispositivos finais ou então retransmitir os dados de sensores para o servidor para que sejam devidamente processados. Esses dispositivos não necessitam efetuar nenhum processamento do ponto de vista de validação. Tudo isso é feito pelos servidores (ALLIANCE, 2023). Assim, os *gateways* são essencialmente transceptores da modulação LoRa que possuem comunicação direta com um dispositivo de maior capacidade de processamento.

2.4 GNU RADIO

O GNU Radio é uma ferramenta de código aberto voltada para o desenvolvimento de rádios definidos por *software*, atuando primordialmente como uma biblioteca de processamento de sinais digitais. Esse *toolkit* foi criado para abstrair e facilitar a criação, teste e execução de algoritmos de processamento de sinal em tempo real (PROJECT, 2023a).

O projeto teve seu desenvolvimento iniciado em 2001, por Eric Blossom, que queria desenvolver um *software* para capturar sinais de televisão em alta definição (NORTON, 2023). Por sua vez, Matt Ettus foi responsável por desenvolver o *Universal Software Radio Peripheral* (USRP) um dos primeiros *hardwares* SDR desenvolvidos para o público geral. Esse foi o *hardware* utilizado por Blossom para capturar as ondas eletromagnéticas irradiadas e então tratá-las em seu *software*. Atualmente, o USRP é um entre tantos produtos da National Instruments mostrando o quanto o seu projeto de *hardware* evoluiu (ETTUS, 2023).

De igual forma, o próprio GNU Radio cresceu inclusive sendo utilizado como ferramenta no mundo acadêmico. Atualmente, encontra-se disponível para ambientes Windows, Mac e várias distribuições Linux (GNU RADIO WIKI, s.d.). Nos últimos dez anos, mais de uma centena artigos foram publicados pela IEEE, entre conferências e revistas científicas, com o GNU Radio como uma das palavras-chave (XPLORE, 2023). Por sua vez, incontáveis outros trabalhos foram desenvolvidos no campo de rádios definidos por *software* utilizando a ferramenta.

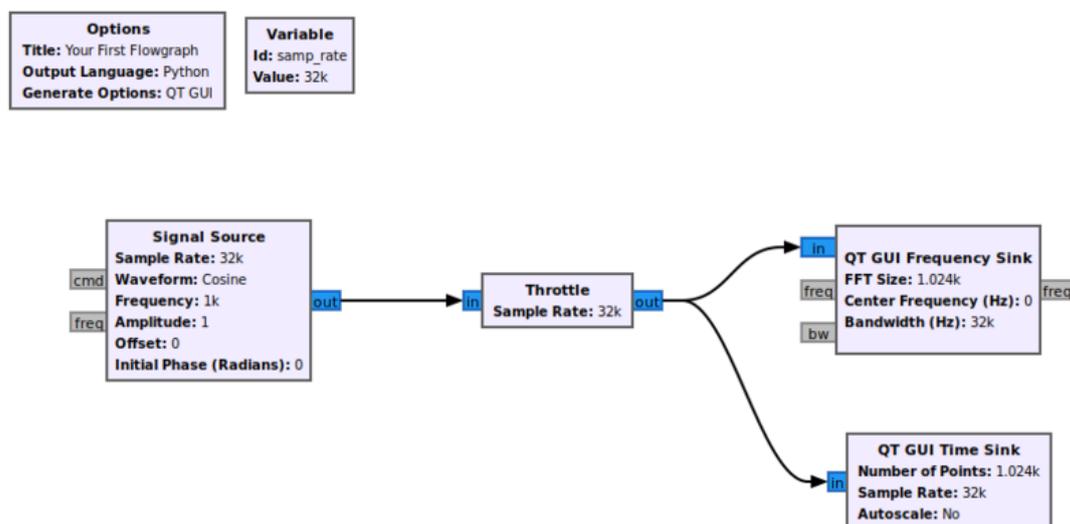
Atualmente, o GNU Radio conta com uma interface gráfica para o desenvolvimento de fluxogramas: o *GNU Radio Companion* (GRC). A Figura 7 apresenta um fluxograma de exemplo criado no *GNU Radio Companion*.

Os fluxogramas criados facilitam o desenvolvimento de aplicações utilizando o GNU Radio. Os diagramas gerados são salvos no formato *.grc*, que é essencialmente um arquivo de configuração que contém a posição e as relações entre os blocos, bem como as variáveis criadas. O GRC é responsável por, quando solicitado, exportar os fluxogramas desenvolvidos para códigos em *Python* ou C++ (PROJECT, 2023d).

De maneira geral, todo o processamento de baixo nível é efetuado pela biblioteca principal do GNU Radio escrita em C++. Todavia, o programa apresenta API para usuários realizarem a integração dessa biblioteca principal com outras bibliotecas e ferramentas da

linguagem *Python*. Além disso, o GNU Radio suporta uma ampla gama de dispositivos de *hardware* SDR, como USRP, HackRF One e RTL-SDR, tornando-o uma das ferramentas mais flexíveis para experimentos práticos.

Figura 7: Fluxograma básico para visualização de uma senoide construído no GRC.



Fonte: (PROJECT, 2023a)

Atualmente, o GNU Radio bem como o *GNU Radio Companion* são mantidos pela GNU Radio Project (PROJECT, 2023c). Essa organização sem fins lucrativos assegura que o *software* continue sendo atualizado, além de proporcionar um ambiente para a troca de informações entre a comunidade de usuários. De igual forma, também é responsável por conservar a documentação oficial do projeto. Essa referência é gerada utilizando o *Doxygen* — ferramenta que interpreta comentários especiais diretamente do próprio código fonte implementado — e pode ser consultada no *GNU Radio Manual and C++ API Reference* (PROJECT, 2023b).

3 MATERIAIS E MÉTODOS

Neste capítulo apresenta-se o desenvolvimento da análise da implementação do protocolo LoRa utilizando o GNU Radio.

3.1 OBJETIVOS E PRÉ-REQUISITOS

A primeira etapa do projeto consistiu em definir quais as metas e requisitos devem ser atingidos pela análise proposta. A maior motivação do presente trabalho é verificar se uma implementação com uso de SDR do protocolo LoRa, desenvolvida por (TAPPAREL et al., 2020) apresenta um comportamento consistente com a especificação do protocolo e como essa implementação reage a ruídos gaussianos. Com isso, busca-se verificar também a viabilidade do uso do protocolo LoRa em uma rede de comunicação onde a relação sinal-ruído seja baixa. Como objetivo secundário, busca-se verificar a similaridade de desempenho da implementação de (TAPPAREL et al., 2020) em comparação com a versão oficial desenvolvida pela Semtech.

Para isso, dados provenientes de simulações efetuadas no GNU Radio, dados de equipamentos que utilizem os chips LoRa da Semtech e uma relação numérica propostas para o comportamento da modulação LoRa (ELSHABRAWY; ROBERT, 2018) serão comparados. A comparação efetuada utilizará a métrica de taxa de erro de *bits* (BER).

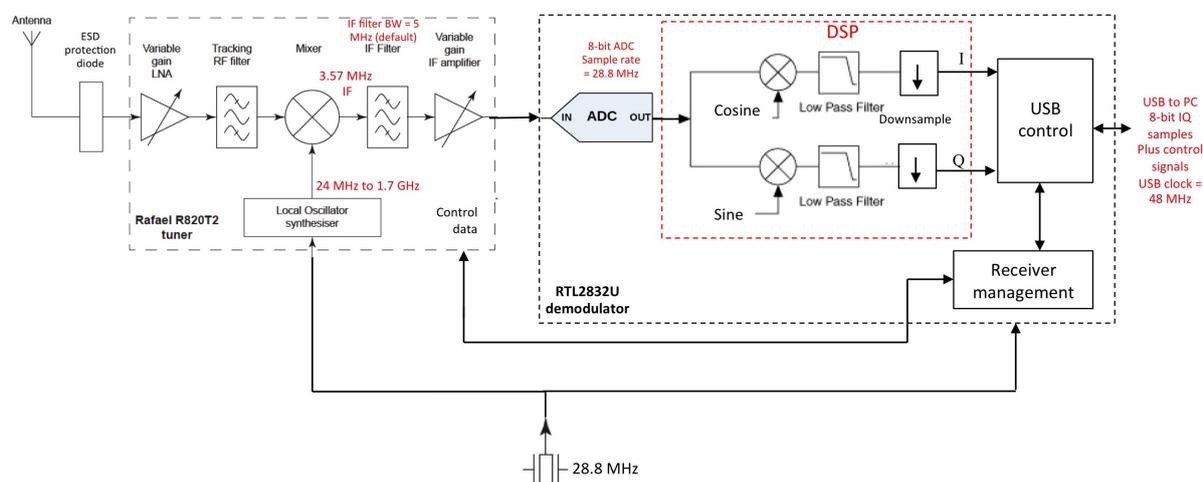
O primeiro pré-requisito que deve ser elencado é a necessidade de um ambiente de simulação estável. Com esse propósito é necessário definir as versões de *software* que devem ser utilizadas. Considerando o período compreendido entre o início do projeto e o registro deste relatório é possível que novas versões estejam disponíveis. Todavia, as principais dependências de *software* utilizadas e suas versões foram mantidas desde o princípio, sendo listadas a seguir.

- GNU Radio Companion v3.10.1.1 (PROJECT, 2023c);
- Python v3.10.12 (PYTHON SOFTWARE FOUNDATION, 2024);
- CubicSDR v0.2.5 (CLIFFORD, C. J., 2024);
- gr-lora_sdr v0.5.5 (TAPPAREL et al., 2020);
- rtl-sdr v2.0.1 (OSMOCOM, 2024);
- LimeSuite v23.10.0 (MYRIADRF, 2024);
- Arduino IDE 2.2.1 (ARDUINO, 2024);
- Heltec ESP-32 Dev-Boards v1.12.0 (HELTEC AUTOMATION, 2024);

Uma especificação muito relevante para os experimentos é a potência de transmissão deste dispositivo. A placa WiFi LoRa 32 (V2) possibilita a transmissão mínima de 5 ± 1 dBm e máxima de 19 ± 1 dBm.

Um dos *hardwares* SDR escolhidos foi o RTL-SDR V3.0 (RTL-SDR, 2023). Esse dispositivo utiliza o *chip* RTL2832U da Realtek e atua somente como receptor numa faixa de 22 MHz à 2,2 GHz. Os principais motivos para a sua escolha foram o seu baixo custo, a sua alta disponibilidade e capacidade de atender a faixa de operação dos protocolo LoRa. A Figura 9 apresenta o diagrama de blocos do *hardware* desse equipamento.

Figura 9: Diagrama de blocos do RTL-SDR.

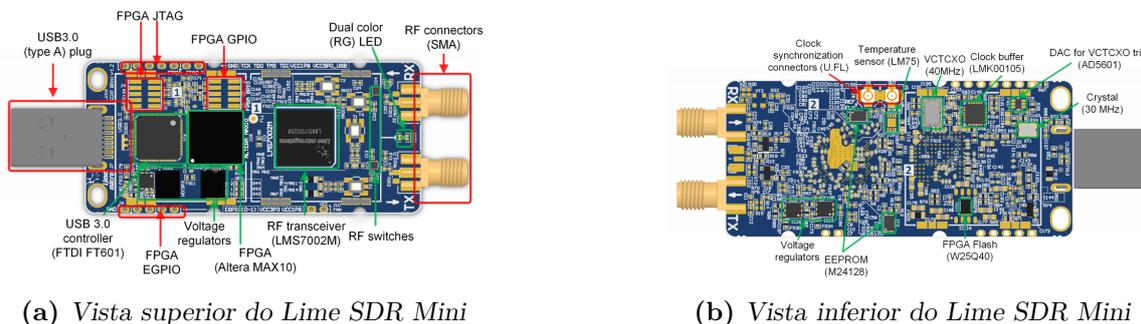


Fonte: Scher (2023)

Analisando a Figura 9 percebe-se a presença do *frontend* de RF padrão com seu LNA e filtros, seguidos do conversor analógico digital. O sinal digitalizado é então processado de forma a extrair os dados em quadratura (amostras I e Q) e então finalmente transmiti-las via USB 2.0 para o computador que executa a demodulação do sinal capturado. Sua simplicidade o torna ideal para os testes de recepção.

Para a transmissão de RF de fluxogramas implementados no GNU Radio, será utilizado o Lime SDR Mini (LIME MICROSYSTEMS, 2024). Esse *hardware* SDR utiliza o transceptor Lime Microsystems LMS7002M para lidar com os sinais de RF, dois conectores SMA (um para a transmissão e outro para a recepção), uma FPGA Altera MAX 10, conexão USB 3.0 Tipo A e possibilidade de sinal de relógio externo. Sua faixa de operação vai de 10 MHz até 3,5 GHz. Além disso, o Lime SDR Mini encontra-se disponível no Laboratório de Comunicações da UFRGS (LACOM-UFRGS). A Figura 10 apresenta uma visão geral desse *hardware*, com destaque a seus principais elementos.

Figura 10: Vistas superiores e inferiores do Lime SDR Mini com destaque a algumas interfaces e componentes



Fonte: Myriad-RF Community (2024)

A potência de transmissão do Lime SDR Mini também é relevante para os experimentos. Seu valor máximo é de 10 dBm, todavia, esse valor está condicionado em parte à frequência de uso do equipamento. Assim, para uma operação coerente, o sinal de saída na frequência desejada deve ser verificado antes dos experimentos. Mais detalhes sobre a implementação do circuito e do *hardware* como um todo estão disponíveis em (MYRIAD-RF COMMUNITY, 2024). Esse equipamento será, num primeiro momento, utilizado para injetar um ruído gaussiano no caminho fechado do experimento prático, de forma a elevar o nível de ruído.

Para combinar os sinais de RF gerados pelo Lime SDR Mini e pela placa WiFi LoRa 32 (V2) foi utilizado um combinador/divisor de sinais Wilkinson J2065 da JBM (JBM INSTRUMENTOS, 2024). Este dispositivo atua na faixa de 700 MHz a 2,7 GHz, possui uma impedância de 50 Ohms, duas portas de entrada e uma porta de saída. Um diagrama exibindo as conexões efetuadas para os testes práticos será apresentado posteriormente. A Figura 11 apresenta o combinador de sinais utilizado.

Figura 11: Combinador/divisor de sinais Wilkinson J2065 da JBM



Fonte: JBM Instrumentos (2024)

Devido a necessidade de validar a saída de potência dos transmissores torna-se necessário o uso de um equipamento capaz de realizar essa tarefa. Isso foi possível utilizando

a ponteira de medição de potência U2001A da Agilent Technologies também disponível no LACOM. Esse equipamento tem uma faixa dinâmica de 80 dB, sendo capaz de ler potências de -60 dBm a +20 dBm (AGILENT TECHNOLOGIES, s.d.[c]). Para a exibição dos dados, essa ponteira é ligada ao analisador de espectro FieldFox N9912A da Agilent Technologies via USB (AGILENT TECHNOLOGIES, s.d.[b]). No analisador de espectro, é possível verificar os valores lidos utilizando uma interface gráfica. Ambos os equipamentos encontram-se disponíveis para o uso no LACOM. A Figura 12 apresenta esses instrumentos.

Figura 12: Equipamentos utilizados para medir a potência de transmissão



(a) Ponteira de potência U2001A da Agilent



(b) Analisador de RF N9912A da Agilent

Fonte: do Autor

Nos experimentos práticos, buscando preservar o *hardware* do RTL-SDR, um atenuador de 30 dB VAT-30+ da Mini-Circuits foi utilizado e conectado diretamente a esse equipamento. Isso foi feito pois a potência máxima de entrada do RTL-SDR é de 10 dBm. Assim, caso o sinal combinado do Lime SDR Mini e do transmissor LoRa fossem injetados nesse equipamento em potências próximas as suas máximas, esse equipamento poderia ser danificado. Vale notar que o VAT-30+ da Mini-Circuits não apresenta uma atenuação constante em toda a faixa de frequências (MINI-CIRCUITS, s.d.). Todavia, como o foco do seu uso é a proteção do RTL-SDR, o impacto dessa variação não é significativo.

Para alterar a relação sinal-ruído na prática, optou-se por atenuar a potência de saída do transmissor LoRa. Essa potência é atenuada utilizando uma combinação dos atenuadores variáveis da Agilent (AGILENT TECHNOLOGIES, s.d.[a]). Um dos atenuadores variáveis utilizado foi o modelo 8496B. Ele possui uma atenuação máxima de 110 dB e passos de 10 dB. Esse equipamento é ligado em série com o atenuador variável 8494B cuja atenuação máxima é de 11 dB e apresenta passos de 1 dB. Quando conectados em série, fornecem uma faixa de atenuação de até 121 dB e uma resolução de 1 dB. Os atenuadores ligados em série encontram-se expostos na Figura 13.

Figura 13: Atenuadores variáveis conectados em série.



Fonte: do Autor

3.2 ANÁLISES PROPOSTAS

A principal curva que será gerada pelo estudo é a curva $BER \times SNR$. A curva de $BER \times SNR$ é uma das curvas mais comuns quando se analisa a qualidade de protocolos (COLLINS; WYGLINSKI; GETZ, 2018).

As medições obtidas serão comparadas considerando propriedades do protocolo e análises teóricas já efetuadas provenientes de uma aproximação teórica disponível (FERREIRA DIAS; RODRIGUES DE LIMA; FRAIDENRAICH, 2019) (ELSHABRAWY; ROBERT, 2018). De igual forma, as curvas de simulações e dos experimentos serão comparadas entre si, de forma a verificar se a simulação é uma representação significativa dos resultados práticos.

Cada experimento variará alguns parâmetros do protocolo. Os parâmetros que serão alterados são o CR, a BW e o SF. Os valores definidos para o CR foram: 1 e 4. Vale notar que é possível utilizar um CR de 0, porém isso compromete a robustez do protocolo, além de não ser muito utilizado na prática. Dessa forma, os valores de CR 1 e 4 são os valores extremos de codificação e conseqüentemente sua análise é significativa.

Para as larguras de banda de transmissão foram avaliados os valores de 125 kHz, 250 kHz e 500 kHz. Conforme pode ser verificado consultando a Equação 1, uma largura de banda de 500 kHz resulta em maior taxa de dados. Todavia, essa largura elevada tende poluir mais o espectro além de tornar a transmissão menos robusta. Assim, verificado que nas aplicações mais comuns de redes LoRaWAN, a velocidade de comunicação é frequentemente sacrificada em função de um ganho da distância de transmissão (KHUTSOANE; ISONG; ABU-MAHFOUZ, 2017) foram selecionadas as larguras de banda de 125 kHz e 250 kHz para a simulação e experimentos práticos.

Por sua vez, os valores de SF foram definidos em 7, 9 e 10. O protocolo LoRa, possibilita valores de SF entre 5 e 12 (SEMTECH, 2023c), embora os valores de 7 à 12 sejam os mais comuns. Dessa forma, o valor 7 foi escolhido como limite inferior, e por sua vez, o valor 9 e 10 foram escolhidos por se encontrarem no meio dessa faixa. O valor de SF igual a 12 foi desconsiderado pois, embora torne o protocolo significativamente mais robusto, apresenta a menor taxa de dados e tende a ser utilizado em aplicações onde a distância de comunicação é o principal fator de interesse (sensoriamento agrícola, por exemplo). Todavia, notou-se em algumas experimentações iniciais com a implementação SDR do protocolo LoRa (TAPPAREL et al., 2020) que, para os valores de SF 11 e 12,

principalmente com larguras de banda pequenas, alguns erros de execução ocorriam. Esses problemas ocorreram também na execução dos fluxogramas de exemplo que acompanham esse pacote no GNU Radio. Esses problemas estavam vinculados principalmente a sincronia de *frames*, onde a exceção gerada era proveniente do código fonte "*frame_sync.cc*". Portanto, restringiu-se os valores de SF de interesse a 7, 9 e 10. A Tabela 1 apresenta um compilado dos parâmetros do protocolo que serão variados tanto na prática quanto na simulação.

Tabela 1: *Resumo dos parâmetros do protocolo variados ao longo da simulação e experimentos.*

Parâmetro	Valores
SF	7 ou 9 ou 10
BW	125 kHz ou 250 kHz
CR	1 ou 4

Fonte: do Autor

Além disso, optou-se por verificar o efeito da perturbação de caminhos múltiplos na implementação SDR LoRa proposta por (TAPPAREL et al., 2020), somente em simulação, uma vez que esse tipo de experimento prático demanda uma circuito de RF mais complexo. Os demais parâmetros variados foram os mesmos da Tabela 1.

No modelo de canal do GNU Radio, cada caminho é correspondente a uma amostra (*tap*) de um filtro FIR, o qual é utilizado para modelar essa interferência. Essa prática é comum na modelagem de canais de comunicação (COLLINS; WYGLINSKI; GETZ, 2018). Os caminhos e os valores escolhidos estão definidos na Tabela 2. Esses valores foram escolhidos considerando exemplos de fluxogramas encontrados nos fóruns da comunidade do GNU Radio.

Tabela 2: *Caminhos simulados e valores utilizados para a interferência multicaminho.*

Caminho	Valor Utilizado
1º Caminho	$0,5+j0,0$
2º Caminho	$0,3+j0,1$
3º Caminho	$0,2-j0,4$

Fonte: do Autor

Os valores presentes na Tabela 2 buscam simular o sinal sendo recebido por meio de três caminhos onde o sinal de maior intensidade é recebido sem nenhuma defasagem e os demais sinais apresentam alguma defasagem e menor intensidade (a maior intensidade possível em um caminho direto é representada pelo número complexo $1,0 + j0,0$). A implementação do meio físico LoRa é dita resistente a esse tipo de perturbação (SEMTECH, 2023d), logo é esperado que poucas variações sejam observadas.

3.3 DESENVOLVIMENTO

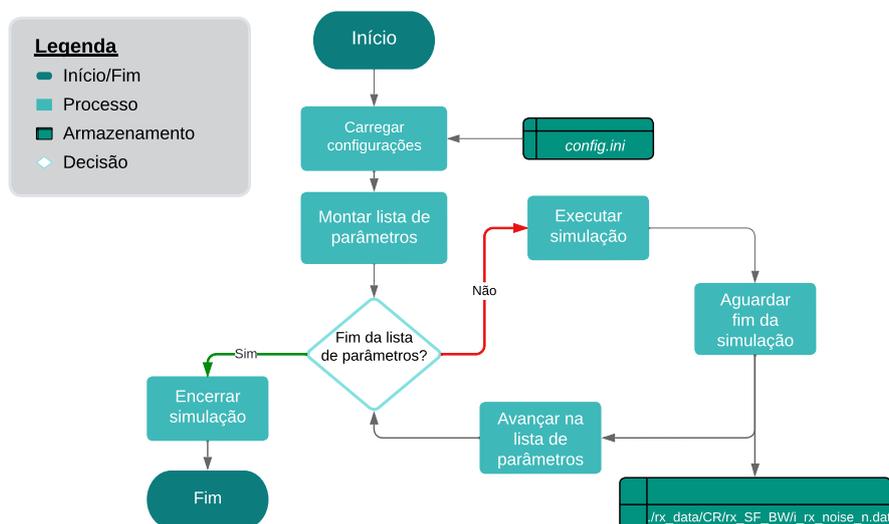
Buscando facilitar a leitura e a organização geral deste trabalho, o desenvolvimento é dividido em subseções. Cada subseção foca em uma parte da implementação necessária para o desenvolvimento do trabalho.

3.3.1 Algoritmo de simulação

A primeira etapa na execução do projeto consistiu em desenvolver o algoritmo utilizado no *script* supervisor da simulação. Como o GNU Radio possibilita que os fluxogramas desenvolvidos possam ser executados por um código externo via chamadas em Python essa foi parte da abordagem proposta. Vale notar que o GNU Radio apresenta algumas funcionalidades de manipulação de arquivos em seus fluxogramas. Essa funcionalidade foi explorada, quando possível, buscando facilitar o desenvolvimento do *script* supervisor.

O *script* supervisor tem a função de carregar os parâmetros, executar a simulação e organizar a estrutura de arquivos necessários para a posterior análise de dados. O algoritmo proposto para executar essas tarefas é apresentado na Figura 14

Figura 14: Fluxograma com o algoritmo proposto para a simulação.

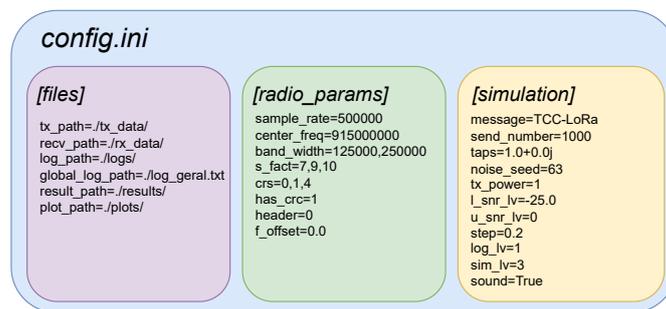


Fonte: do Autor

Conforme é possível visualizar na Figura 14, o algoritmo é direto. Vale notar que foi inserido um arquivo de configurações separado (*config.ini*) que contém os parâmetros da simulação desejada bem como algumas outras informações relevantes. Nesse arquivo de configurações é possível definir quais os valores de CR, SF e BW que serão utilizados na simulação, o número de mensagens que serão transmitidas, a localização dos arquivos de saída entre outros. Esse arquivo é independente do código fonte de forma a possibilitar mudanças da simulação sem alterar o *script* propriamente dito. A Figura 15 apresenta as entradas existentes nesse arquivo de configuração com exemplos de parâmetros definidos.

Considerando a Figura 15, o cabeçalho *files* contém a estrutura de arquivos proposta para o armazenamento de dados necessários com dados de exemplo. O cabeçalho *radio_params* contém as propriedades específicas do protocolo, os quais serão variados ao longo da simulação. Além disso, define alguns parâmetros específicos da implementação SDR como a taxa de amostragem (*sample_rate*). Por sua vez, o cabeçalho *simulation* contém as propriedades específicas para a simulação. Desses, vale destacar as propriedades *l_snr_lw* e *u_snr_lw* as quais definem o limite inferior e superior para os valores de SNR, respectivamente.

Figura 15: Estrutura do arquivo de configurações.



Fonte: do Autor

Outras propriedades, que por sua relevância merecem ser explicadas, são a `send_number` e a `step`. A primeira é responsável por definir quantas mensagens serão enviadas em cada configuração. A segunda define em quantos pontos o intervalo fechado de SNR definido anteriormente por `l_snr_lv` e `u_snr_lv` será dividido. No intervalo das informações expostas, a simulação executada com os dados da Figura 15 transmite um total de 126000 mensagens por configuração do rádio (valor de CR, BW e SF). A propriedade `noise_seed` deve ser variada ou fixada caso os resultados de uma simulação devam ser reproduzidos ou uma nova simulação com os mesmos parâmetros deva ser executada.

3.3.2 Fluxograma GNU Radio da Simulação

Conforme mencionado na subseção 3.3.1, um fluxograma do GNU Radio pode ser executado por um *script* Python. De forma a facilitar a supervisão desse código, cada execução de um fluxograma pelo *script* supervisor utiliza somente uma configuração de parâmetros específicos do protocolo LoRa e um ponto de relação sinal-ruído.

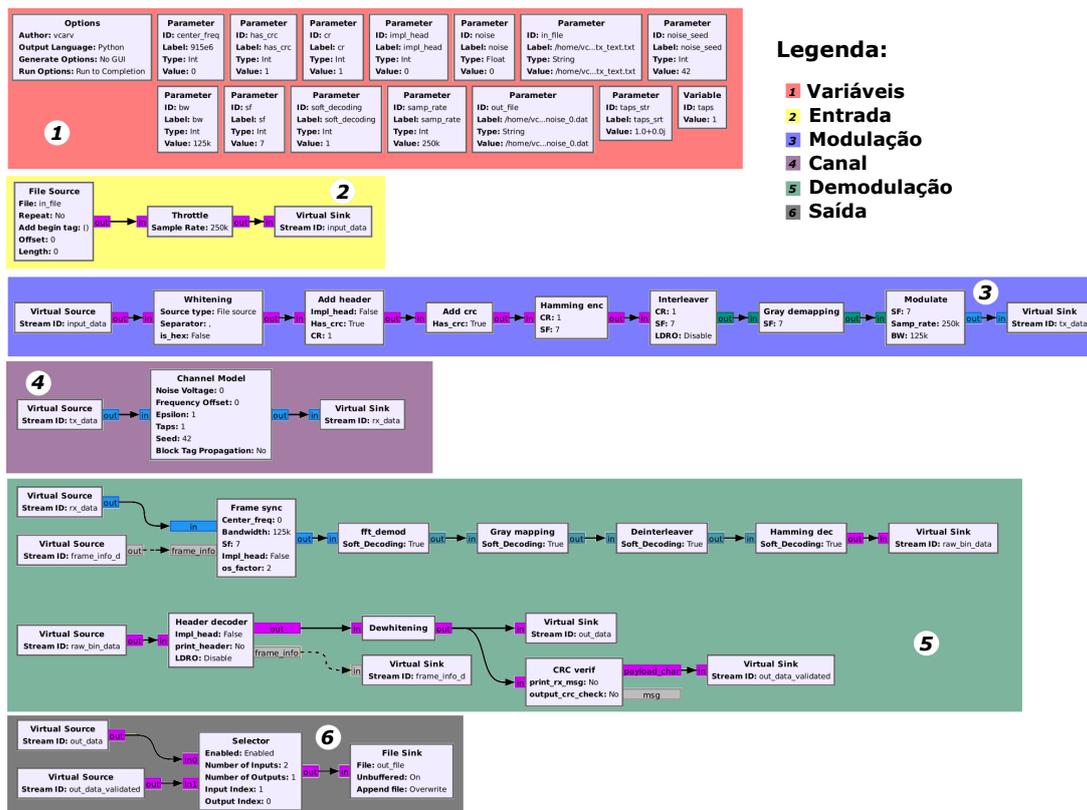
Essa abordagem, embora de simples implementação, torna-se mais custosa computacionalmente devido a necessidade de inicializar o fluxograma em cada combinação de parâmetros do protocolo e ponto de SNR. Porém, proporciona uma vantagem na manipulação de arquivos. Da forma como o fluxograma foi elaborado, cada simulação gera seu próprio arquivo com as mensagens recebidas sem a necessidade de nenhuma interferência por parte do supervisor, exceto nos nomes dos arquivos. A Figura 16 apresenta o fluxograma desenvolvido.

Conforme é possível observar na Figura 16, esse fluxograma pode ser dividido em seis áreas. A primeira área, destacada em vermelho, contém as opções do fluxograma, bem como os parâmetros que são recebidos quando o fluxograma é inicializado pelo *script* supervisor. Essa área também contém algumas variáveis locais responsáveis por efetuar controles de fluxo e conversões de tipos necessários.

A segunda subdivisão é responsável por carregar o arquivo que contém as mensagens que devem ser transmitidas, uma linha por vez, processado como *bytes* para que possa ser transmitido. O bloco *Throttle* foi inserido para limitar o uso de CPU. A saída dessa subdivisão é conectada à terceira subdivisão azul.

Nessa subdivisão são utilizados os blocos do *gr-lora_sdr* para efetuar a modulação LoRa proposta pelos autores desse pacote (TAPPAREL et al., 2020). Inicia-se pelo processo de branqueamento (*Whitening*). Esse processo consiste em transformar os dados de forma que sua densidade de energia espectral fique igualmente distribuída (de maneira

Figura 16: Fluxograma do GNU Radio utilizado para as simulações.



Fonte: do Autor

análoga ao ruído branco, o que justifica seu nome) (COLLINS; WYGLINSKI; GETZ, 2018). A seguir, é efetuada a adição do cabeçalho (*Add header*) com o CR desejado, adição de CRC (*Add crc*) e codificação Hamming (*Hamming enc*) para possibilitar a correção de erros (COLLINS; WYGLINSKI; GETZ, 2018). Após, utiliza-se um bloco entrelaçador (*Interleaver*) para otimizar a forma como os símbolos são transmitidos, seguido de um desmapeamento de código Gray (*Gray demapping*) para então chegar, finalmente, na etapa onde os dados são modulados em quadratura (amostras I e Q) utilizando o SF e BW definidos (bloco *Modulate*).

Na quarta subdivisão, é efetuada a perturbação no sinal transmitido ao passar pelo modelo de canal AWGN. Nesse bloco vale a pena notar que existe o campo *Noise Voltage*. É nesse campo que, com base no valor de potência da transmissão informado no *config.ini*, o supervisor irá inserir um novo valor de forma a gerar uma relação sinal-ruído desejada. Esse valor representa o desvio padrão que a curva gaussiana utilizada para gerar os sinais de ruído terá. Outro campo de interesse é o campo *Taps* onde é possível inserir uma lista de números complexos de forma a simular a intensidade do sinal recebido e eventuais distorções de fase, assim como o efeito de interferência de multicaminhos.

Na penúltima subdivisão é efetuada a demodulação do sinal recebido da saída do modelo de canal. O primeiro processo consiste em buscar a sincronia com um novo quadro (*Frame sync*), seguido de uma demodulação utilizando uma Transformada Rápida de Fourier de maneira específica (bloco *fft_demod*). Após essa etapa, o processo é essencialmente o inverso das etapas de modulação, exceto pela etapa de verificação de CRC (*CRC verif*) que é executada após o processo de desbranqueamento (*Dewhitening*).

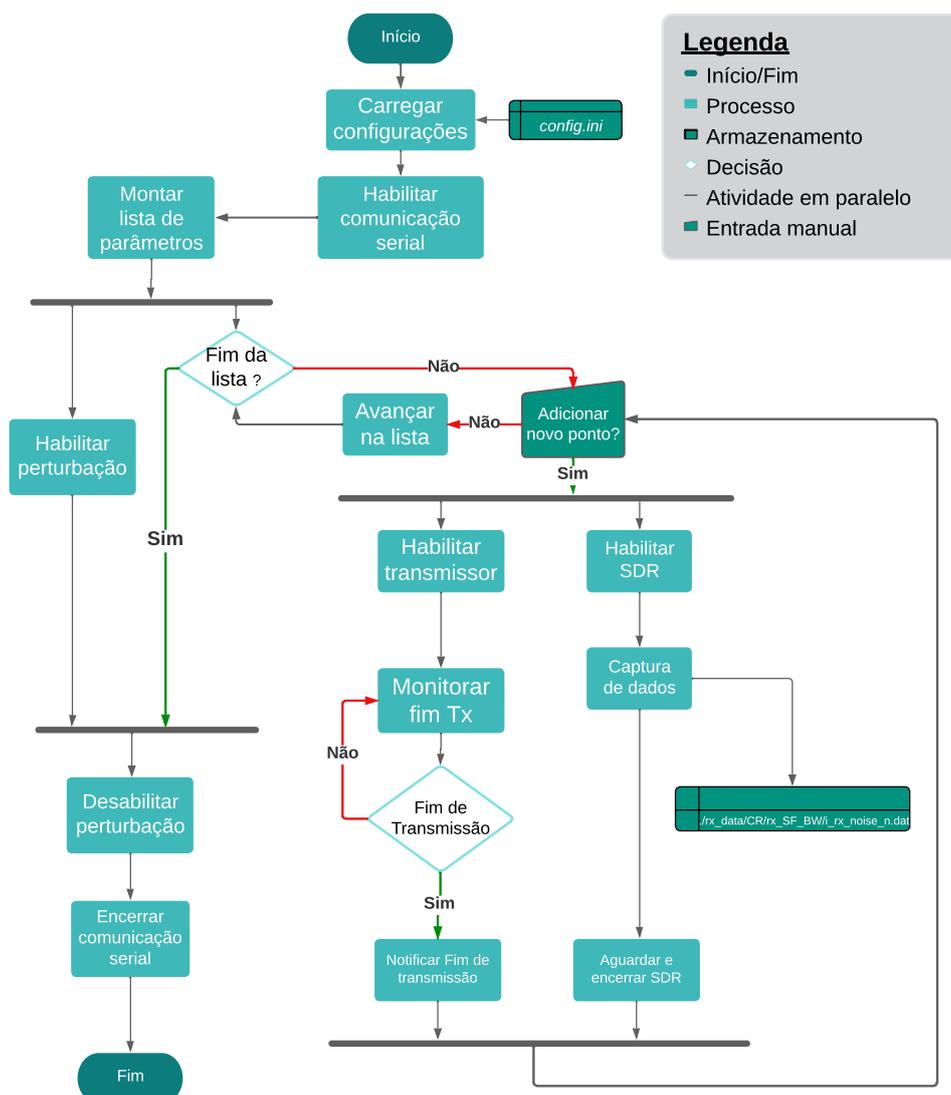
Finalmente, temos a subdivisão de saída. Nessa área existe um seletor que possibilita

a escolha entre salvar em um arquivo os dados brutos ou somente mensagens onde o CRC foi verificado corretamente.

3.3.3 Algoritmo para a prática com SDR Rx

De maneira análoga a simulação, foi desenvolvido um algoritmo para o experimento onde um SDR atua como receptor e um transceptor LoRa da Semtech atua como transmissor. Esse algoritmo também foi desenvolvido para, posteriormente, ser implementado na linguagem Python. Buscou-se, durante o desenvolvimento do código desse supervisor, implementá-lo de maneira genérica de forma que grande parte pudesse ser aproveitada para outras variações do experimento prático. A Figura 17 apresenta o algoritmo proposto.

Figura 17: Fluxograma com o algoritmo proposto para prática SDR RX - LoRa chip Tx.



Assim como na parte prática, o *script* supervisor é responsável por carregar os parâmetros de transmissão e recepção do arquivo *config.ini*. Vale notar que algumas propriedades do cabeçalho "*simulation*" (Figura 15 subseção 3.3.1) são utilizadas como parâmetros do transmissor. Posteriormente, o *script* deve habilitar a comunicação serial com o transmissor LoRa. A comunicação serial com a placa WiFi LoRa 32 (V2) tem o objetivo de possibilitar a configuração deste equipamento durante o experimento bem como monitorar quando uma sequência de transmissão proposta foi encerrada.

Ainda sobre a comunicação serial, um padrão de mensagens foi definido para garantir que a configuração correta do transmissor foi efetuada. A Tabela 3 apresenta o formato de mensagens utilizado, onde o *host* é o computador ao qual o SDR está conectado e executa o *script* supervisor e o cliente é a placa WiFi LoRa 32 (V2).

Tabela 3: Mensagens utilizadas na comunicação serial

Nº	Mensagem Host	Mensagem Cliente	Objetivo
1	Setup#CR#SF#BW#N_MSG#PWR#MSG!	SetupOk!	Configurar o transmissor
2	Start!	StartOk!	Iniciar a transmissão
3	EndSimOk!	EndSim!	Notificar fim do experimento
4	Qualquer outra mensagem	Erro!	Notificar problema na comunicação serial

Fonte: do Autor

A mensagem mais complexa que é transmitida no esquema proposto é a mensagem número 1 da Tabela 3. O conjunto de dados enviados contém um total de 7 campos separados pelo caractere "#". O primeiro campo indica que é uma mensagem de configuração, o segundo campo contém o valor de *coding rate*, o próximo valor do fator de espalhamento e o quarto campo indica qual a largura de banda que será utilizada. O quinto campo contém o número de mensagens que são enviadas pelo transmissor. A seguir existe um campo onde é definida a potência da transmissão e, finalmente, um último campo onde é definido o conteúdo da mensagem que será transmitida. Além disso, o caractere "!" é utilizado para indicar o fim de qualquer mensagem.

A principal diferença entre o algoritmo de simulação e o algoritmo de recepção por parte do SDR está no fato que foi utilizada programação concorrente para possibilitar que a serial possa ser monitorada enquanto o receptor SDR está executando seu código de captura. Além disso, existe a necessidade de uma entrada manual enquanto se executa o experimento. No momento dessa entrada manual, deve ser inserido o valor de atenuação que está sendo aplicado ao sinal transmissor, para posteriormente possibilitar o cálculo da relação sinal-ruído. Caso se deseje ir para o próximo valor da lista de parâmetros, deve ser inserido um comando para sair dessa etapa do experimento.

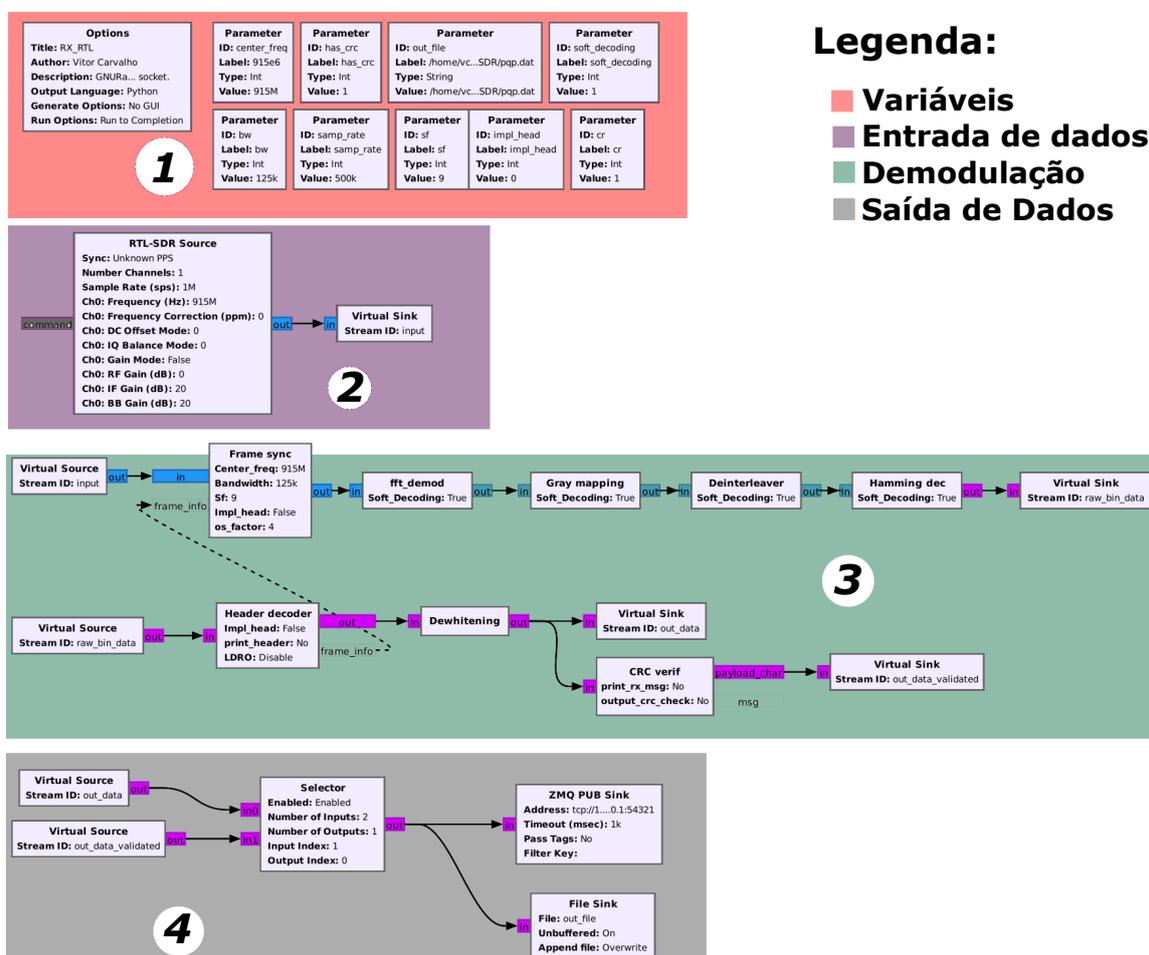
Uma *thread* específica também é habilitada para a execução do bloco responsável por gerar a perturbação. Dessa forma, o *script* supervisor baseado nesse fluxograma deve executar duas *threads* que executam os fluxogramas do GNU Radio.

3.3.4 Fluxograma GNU Radio SDR RX

O desenvolvimento do fluxograma para a recepção de pacotes LoRa foi significativamente baseado no fluxograma de simulação. Essencialmente, foram removidos os elementos de transmissão e modelo de canal da Figura 16 (subseção 3.3.2) e, como entrada do sistema, inserido o bloco *RTL-SDR Source*. Esse bloco possibilita a aquisição das amostras em quadratura capturadas pelo *hardware* RTL-SDR v3.0. A Figura 18 apresenta o fluxograma

desenvolvido no GNU Radio para o experimento de recepção com SDR e transmissão com um *chip* LoRa oficial.

Figura 18: Fluxograma do GNU Radio utilizado para a prática SDR RX - LoRa *chip* Tx.



Fonte: do Autor

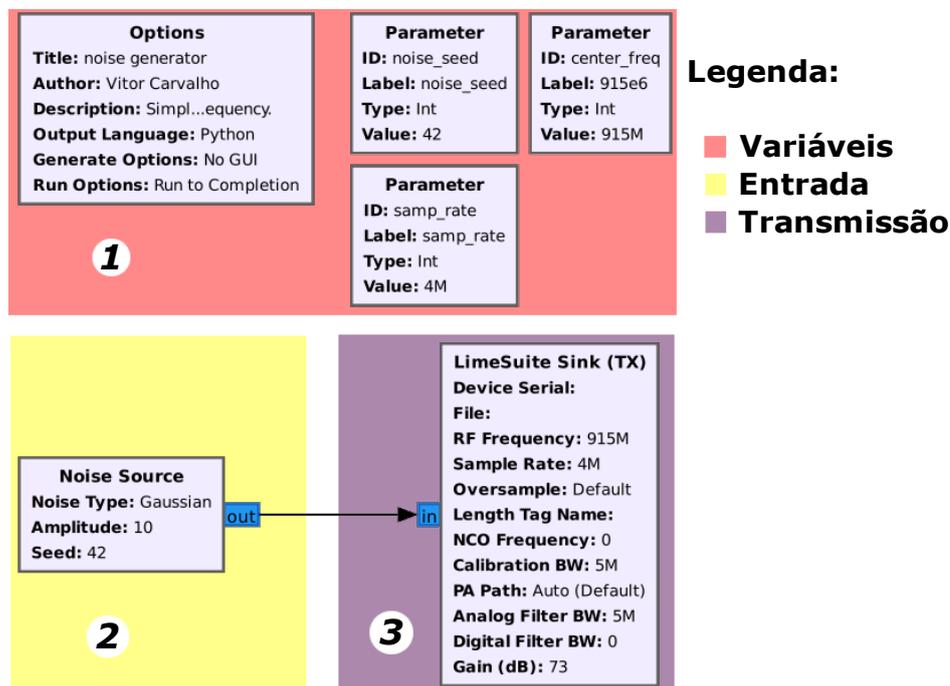
Vale notar que a taxa de amostragem é definida em 1,0 MHz para o RTL-SDR. Esse valor deve ser um múltiplo exato da largura de banda devido a restrições na implementação proposta por (TAPPAREL et al., 2020).

O processo de demodulação é o mesmo ao utilizado na simulação. Logo, a subdivisão do fluxograma conta com, além da saída para um arquivo, a saída para um *socket* ZMQ. Esse *socket* foi inserido nesse fluxograma para facilitar a visualização em tempo real do dados recebidos.

3.3.5 Fluxograma GNU Radio Ruído AWGN

Para a geração da perturbação no canal, optou-se por utilizar um SDR como transmissor. Essa solução é de fácil integração com os demais *scripts* que foram desenvolvidos e, de igual forma, mostra a versatilidade de *hardware* SDR. A Figura 19 apresenta o fluxograma desenvolvido.

Figura 19: Fluxograma do GNU Radio para inserir ruído AWGN no sistema



Fonte: do Autor

Como é possível observar na Figura 19, o fluxo proposto para a geração do ruído é composto do bloco de ruído AWGN disponível no próprio GNU Radio, seguido da saída para o transmissor. No caso, é utilizada a saída para o Lime SDR Mini. Vale notar que os parâmetros de ganho e semente do ruído podem ser configurados externamente durante a criação dos objetos desse fluxograma.

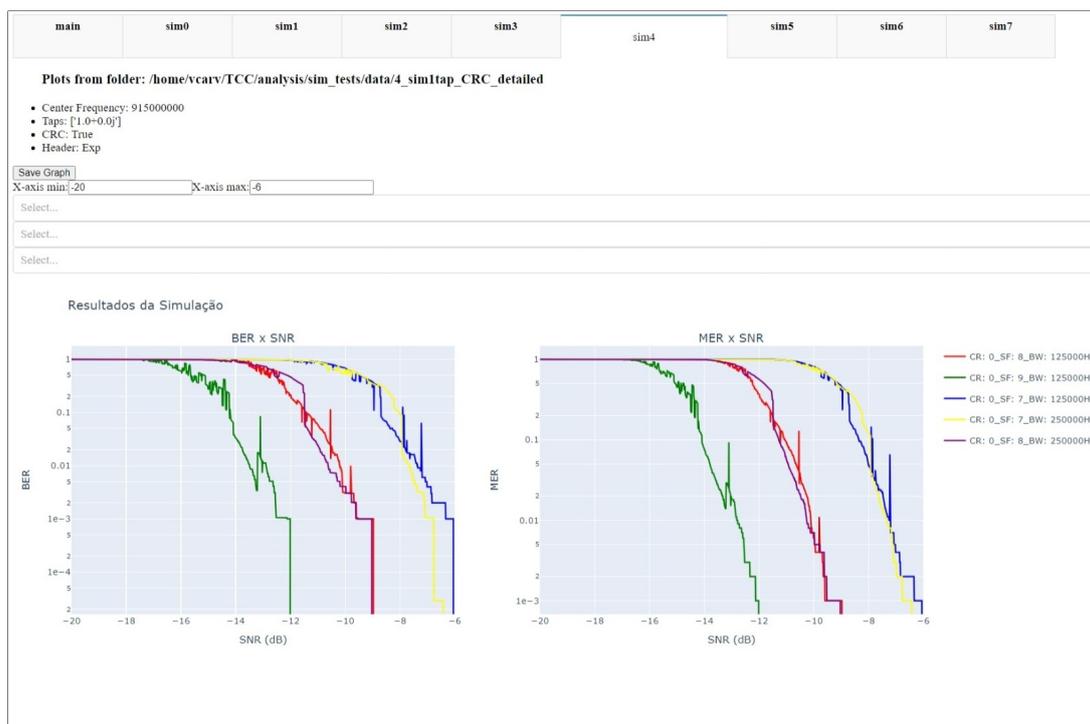
3.3.6 Visualização de dados

Buscando facilitar a visualização e interpretação inicial dos dados de simulação e experimentos, foi desenvolvido um sistema interativo para a visualização de gráficos. Seu principal objetivo é possibilitar, por meio de uma análise qualitativa, a verificação da consistência dos dados obtidos.

Com a aplicação, é possível validar que a coleta de dados foi feita de maneira coerente e auxiliar a remover eventuais erros de implementação dos supervisórios que pudessem corromper os resultados. Essa aplicação conta com um sistema de seleção de parâmetros (por meio de operações de lógica AND, entre os filtros) de forma a exibir somente gráficos de interesse. A Figura 20 apresenta uma captura de tela desse visualizador mostrando alguns gráficos.

No caso dos gráficos da Figura 20, não existe nenhum filtro ativo e observa-se que as curvas são pouco suaves. Os dados que geram essa curva são provenientes de uma única simulação onde mil pontos foram utilizados para cada valor de SNR e a distância entre valores de SNR foi de um décimo. Com isso, percebeu-se a necessidade de aumentar significativamente o número de mensagens transmitidas para suavizar a curva. De igual forma, verificou-se que a tendência geral da curvatura desses resultados é consistente com o esperado, mostrando que a forma de coleta e processamento inicial dos dados é coerente.

Figura 20: Aplicação interativa para visualização inicial dos resultados.



Fonte: do Autor

Na Figura 20, o gráfico da direita apresenta as curvas $BER \times SNR$ onde são analisadas as relações entre *bits* errados e o total de *bits* enviados. O gráfico da esquerda apresenta a curvas $MER \times SNR$, uma métrica que analisa somente as mensagens recebidas com sucesso.

3.4 EXECUÇÃO

Após implementação dos *scripts* supervisórios e fluxogramas descritos na Seção 3.3 teve início a etapa de execução dos experimentos. Buscando facilitar a leitura, esta seção foi subdividida em subseções dedicadas a simulação e a prática.

3.4.1 Simulações

O primeiro experimento consistiu em executar algumas simulações para validar o sistema como um todo. Nesses testes notou-se que, caso o número de mensagens transmitidas fosse superior a 2500 por valor de SNR, a execução do fluxograma no GNU Radio apresentava algumas inconsistências, devido a forma de manipulação de arquivos proveniente dos blocos do GNU Radio. De igual forma, o problema estava associado a um ponto específico de SNR pois, ao interromper a execução do código naquele ponto, o sistema continuava funcional. Além disso, esse problema ocorreu somente em valores de SNR baixos onde o BER era próximo de um.

Dessa forma, definiu-se um total de 1000 mensagens por ponto. A Tabela 4 apresenta os parâmetros com os quais a primeira simulação, ou seja, o caminho direto usando o modelo de canal AWGN foi executada.

Considerando uma combinação de parâmetros (CR, SF e BW) da Tabela 4, um total de 126 mil mensagens foram enviadas (1000 mensagens por ponto, 126 pontos no intervalo fechado entre o valor máximo e mínimo de SNR). Ao se executar uma dessas simulações e avaliar os resultados de uma configuração de parâmetros usando a aplicação de visualização rápida, notou-se que a curva obtida apresentava pontos de variação abrupta. Logo, visando obter uma curva mais suave e devido a limitação do número de mensagens por ponto observada na implementação desenvolvida, optou-se por executar novas simulações com os mesmos parâmetros, somente variando o valor da semente de ruído. Depois disso, os dados dessas diversas simulações com a mesma configuração mas sementes de ruído distintas foram combinados de forma a serem analisados como único experimento com um número mais expressivo de mensagens por valor de SNR.

Tabela 4: *Resumo dos parâmetros da simulação.*

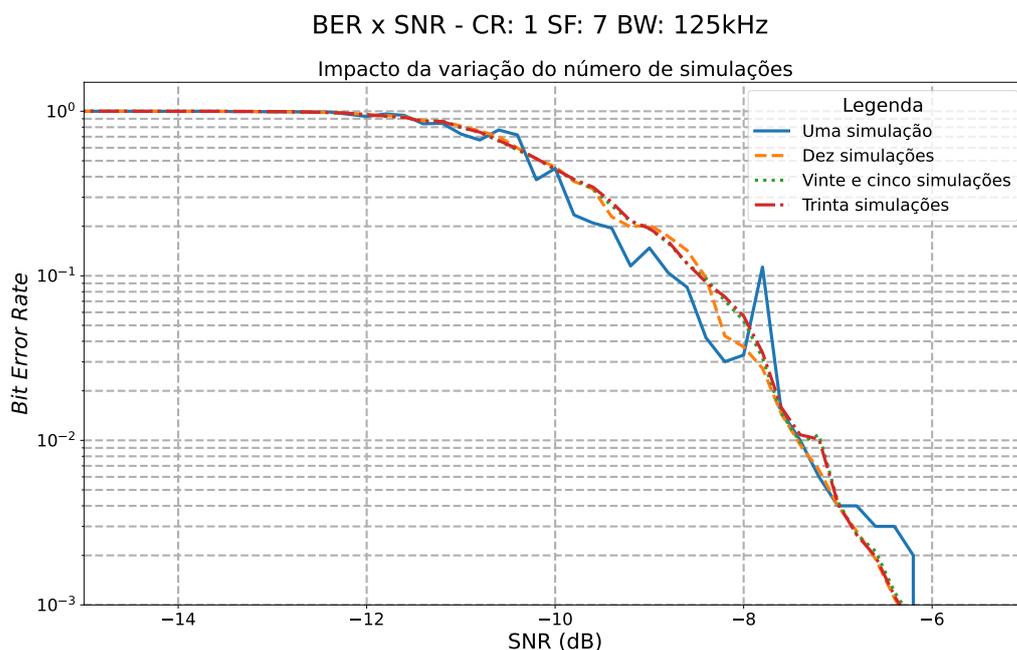
Parâmetro	Valores
SF	7 ou 9 ou 10
BW	125 kHz ou 250 kHz
CR	1 ou 4
Semente do ruído	$0 \leq x \leq 100$
SNR Mínimo	-25 dB
SNR Máximo	0 dB
Espaçamento entre pontos	0,2 dB
Mensagens por semente de ruído	1000
Mensagem	b'TCC-LoRa\n' ¹

Fonte: do Autor

Assim, diversas simulações isoladas foram executadas para atingir uma curva combinada mais suave. No caso desse experimento o valor de 30 simulações foi escolhido. Isso se deve ao fato de que, após uma análise qualitativa, a variação entre 25 e 30 simulações não foi considerada expressiva. A Figura 21 apresenta as curvas $BER \times SNR$ para CR igual a 1, SF igual à 7 e BW igual à 125 kHz, considerando os dados de uma simulação, 10 simulações, 25 simulações e 30 simulações.

¹O uso de b" serve para indicar a codificação em binário da mensagem.

Figura 21: Impacto do número de simulações na suavidade da curva.



Fonte: do Autor

De igual forma, foi efetuada a mesma metodologia para o experimento com interferência de caminhos múltiplos. No caso dessa verificação, somente dez simulações foram efetuadas, totalizando dez mil mensagens transmitidas por ponto de interesse. Esse número reduzido de simulações, frente a simulação com um único caminho, foi escolhido uma vez que a simulação multicaminho exige um tempo que ultrapassaria o cronograma previsto. Os parâmetros da simulação foram definidos conforme a Tabela 4 e os caminhos utilizados conforme descritos na subseção 3.2, Tabela 2.

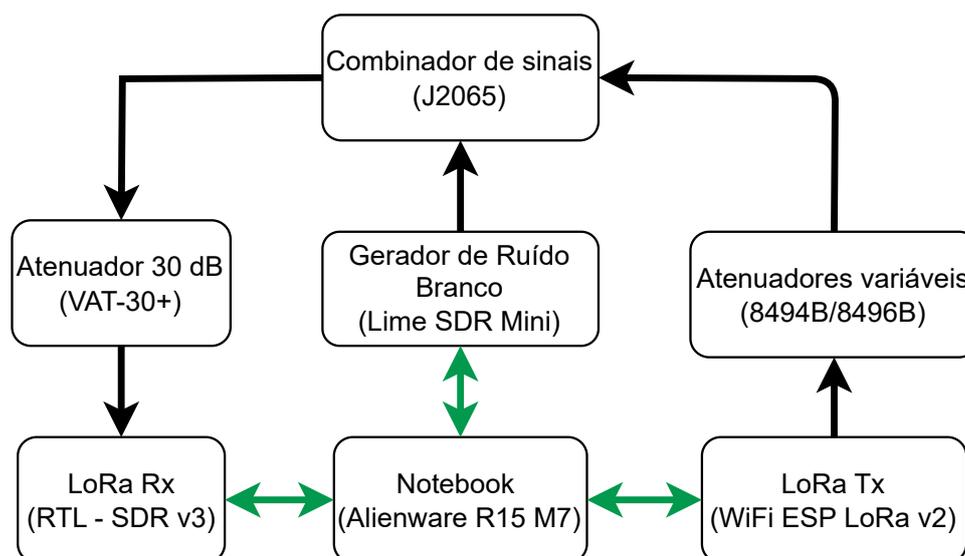
Outro experimento simulado que foi realizado, foi a comparação entre o uso de decodificação suave (*soft decoding*) ao invés de uma decodificação dura (*hard decoding*). A decodificação suave utiliza fatores estatísticos do próprio sinal de entrada para definir se o conteúdo recebido é equivalente a uma ou outra representação binária. Por sua vez, uma decodificação dura não conta com essa decisão probabilística utilizando somente um limite fixo para efetuar a decisão sobre a conversão do sinal de entrada (VISWANATHAN, 2009). De igual forma, um total de dez simulações foram efetuadas com e sem decodificação suave, utilizando os parâmetros da Tabela 4. Com isso, foram concluídos os experimentos a nível de simulação.

3.4.2 Prática

O experimento prático consistiu em validar a implementação LoRa SDR (TAPPAREL et al., 2020) do ponto de vista de recepção em um ambiente com ruído AWGN. Para isso, foi utilizado o RTL SDR v3.0 como receptor, a placa WiFi ESP Lora V2 como transmissor e Lime SDR Mini como gerador de ruído AWGN. As conexões entre esses elementos, o computador que executa o supervisor do experimento, atenuadores e combinadores são apresentadas na Figura 22. Na Figura 22, as conexões com dupla

flecha representam troca de dados, já as conexões com flechas unidirecionais representam o caminho e sentido que o sinal de RF deve percorrer.

Figura 22: Configuração proposta para o experimento



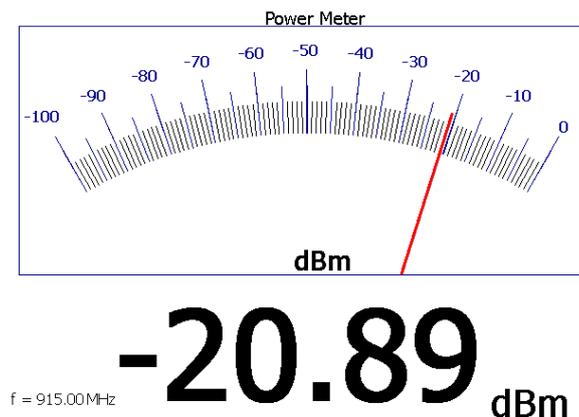
Fonte: do Autor

Antes de finalizar as conexões, foi necessário verificar se as medidas de potência de transmissão placa WiFi LoRa V2 e do Lime SDR Mini eram consistentes com o esperado. Para isso, buscando também considerar eventuais perdas por parte dos cabos de RF, o ponto onde as medidas foram efetuadas foi justamente antes da conexão do atenuador de 30 dB com o RTL SDR v3.

Como a potência máxima entregue pelo Lime SDR Mini é de 10 dBm, optou-se por fixar esse valor como potência saída desse equipamento. Assim, após ser conectado ao combinador e medido na saída do atenuador de 30 dBm, o valor de potência esperado era de -20 dBm. Para executar essas medidas foram utilizadas a ponteira de potência e o analisador de espectro apresentados anteriormente.

Conforme pode ser observado na Figura 23 (capturada utilizando a funcionalidade de acesso remoto do analisador de espectro utilizado), o valor medido foi -20,89 dBm. Dessa forma, existe uma divergência de quase -1 dBm entre o valor teórico esperado e o valor medido. Essa divergência é justificada principalmente pela pequena perda inserida pelo combinador de sinais, adaptador tipo N - SMA, cabos de RF e eventual variação do sinal de potência de saída Lime SDR Mini em função da frequência de operação (MICROSYSTEMS, 2023).

Figura 23: Medida da potência percebida na saída do atenuador de 30 dB quando somente o Lime SDR Mini está transmitindo a 10 dBm.

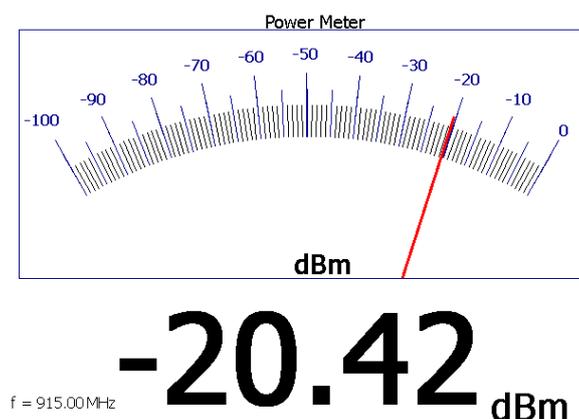


Fonte: do Autor

Assim, o valor de -21 dBm foi considerado como valor de referência para o sinal de ruído que será recebido pelo RTL SDR v3. Logo, a potência que o sinal do transmissor LoRa deve atingir quando os atenuadores variáveis estiverem zerados, o transmissor de ruído desabilitado, considerando o ponto de medida após o atenuador de 30 dB deve ser de -21 dBm ou próximo disso para que essa configuração caracterize o ponto inicial de relação sinal-ruído de 0 dB. Dessa forma, julgando que o caminho de RF era equivalente, iniciou-se configurando a potência de saída da placa WiFi ESP LoRa V2 para 10 dBm. Todavia, quando efetuada a medida na saída do atenuador fixo, verificou-se um valor próximo a -25 dBm — 5 dBm abaixo do valor teórico esperado (-20 dBm). Essa divergência significativa pode ser explicada principalmente pela sequência de adaptadores que são utilizados para conectar a saída da placa de U.FL para SMA que, por sua vez, é adaptada para tipo N, para então ser conectada aos acopladores variáveis ligados em série. Também vale notar que a potência entregue pelo transmissor pode variar entre ± 1 dBm.

Assim, o valor de potência de saída do transmissor foi aumentado para 14 dBm. Depois desse ajuste, uma nova medida foi efetuada. A Figura 24 apresenta o resultado dessa medida.

Figura 24: Medida da potência percebida na saída do atenuador de 30 dB quando somente o transmissor LoRa está operando a 14 dBm

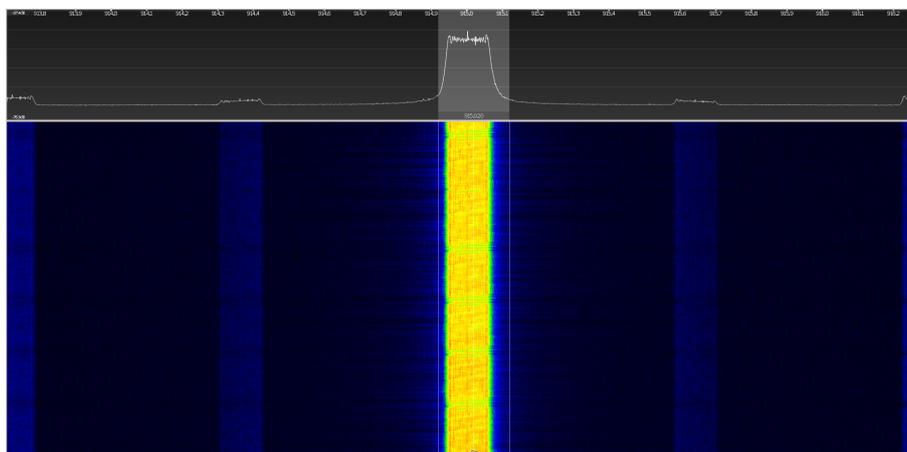


Fonte: do Autor

Ao observar o valor da Figura 24, nota-se que esse valor está próximo de -21 dBm, embora ainda um pouco abaixo ($\approx 0,6$ dBm). Todavia, considerando os equipamentos envolvidos, a configuração de circuito fechado de RF proposta, a potência de saída 14 dBm foi definida para o transmissor LoRa utilizado. Na etapa de análise de dados, uma correção de 0,6 dB (valor de potência do transmissor - valor de potência em dBm do gerador de ruído) será efetuada para gerar um gráfico mais coerente.

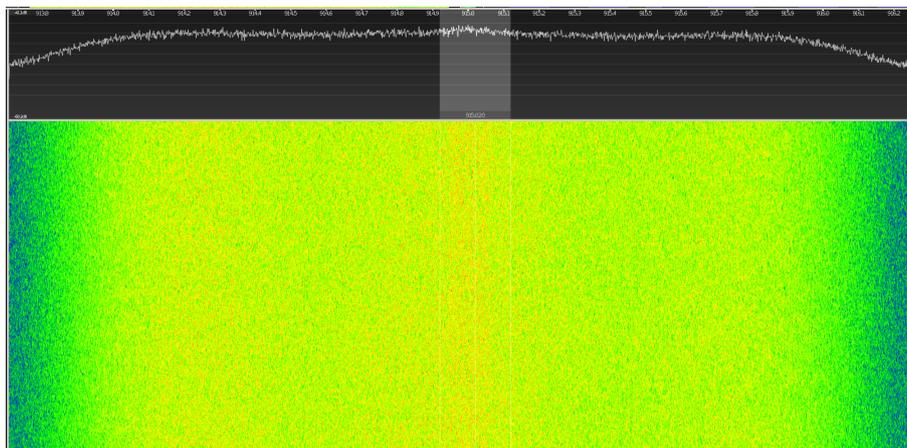
Finalmente, buscando validar o comportamento do sistema, uma análise do espectro foi efetuada utilizando o RTL SDR V3 como coletor de amostras. Na Figura 25 é possível ver o gráfico de cascata somente com o sinal do transmissor habilitado. Esse sinal é facilmente percebido pela sua cor acentuada em relação ao resto dos sinais capturados. Por sua vez, na Figura 26, tanto o sinal de perturbação quando o sinal do transmissor estão habilitados de forma a gerar uma relação sinal-ruído unitária (0 dB). Nesse caso, verifica-se que é impossível notar a presença do sinal de interesse visualmente, conforme esperado. Todavia, vale notar que, na configuração com 0 dB de SNR, o protocolo LoRa é capaz de receber um número próximo a 100% das mensagens enviadas.

Figura 25: Gráfico de cascata somente com o sinal LoRa habilitado



Fonte: do Autor

Figura 26: Gráfico de cascata com o sinal LoRa e perturbação habilitados e SNR de 0 dB



Fonte: do Autor

Uma vez mapeado o comportamento do sistema, os parâmetros apresentados na Tabela 5 foram definidos. Durante as execuções de validações do *script* supervisor para essa atividade, notou-se que um número elevado de mensagens para transmissão não causava travamentos (foram efetuados testes de até 15000 mensagens por ponto). Todavia, para otimizar a análise, optou-se por executar uma transmissão de 5000 mensagens por ponto de interesse. Esse valor foi definido considerando um compromisso de tempo entre um ponto de SNR de interesse e outro e número de dados disponíveis.

Tabela 5: *Resumo dos parâmetros da atividade prática*

Parâmetro	Valores
SF	7 ou 9 ou 10
BW	125 kHz ou 250 kHz
CR	1 ou 4
Semente do ruído	$0 \leq x \leq 100$
SNR Mínimo	-25 dB
SNR Máximo	0 dB
Mensagens por ponto	5000
Mensagem	b'TCC-LoRa\ n' ²
Potência do Transmissor	14 dBm

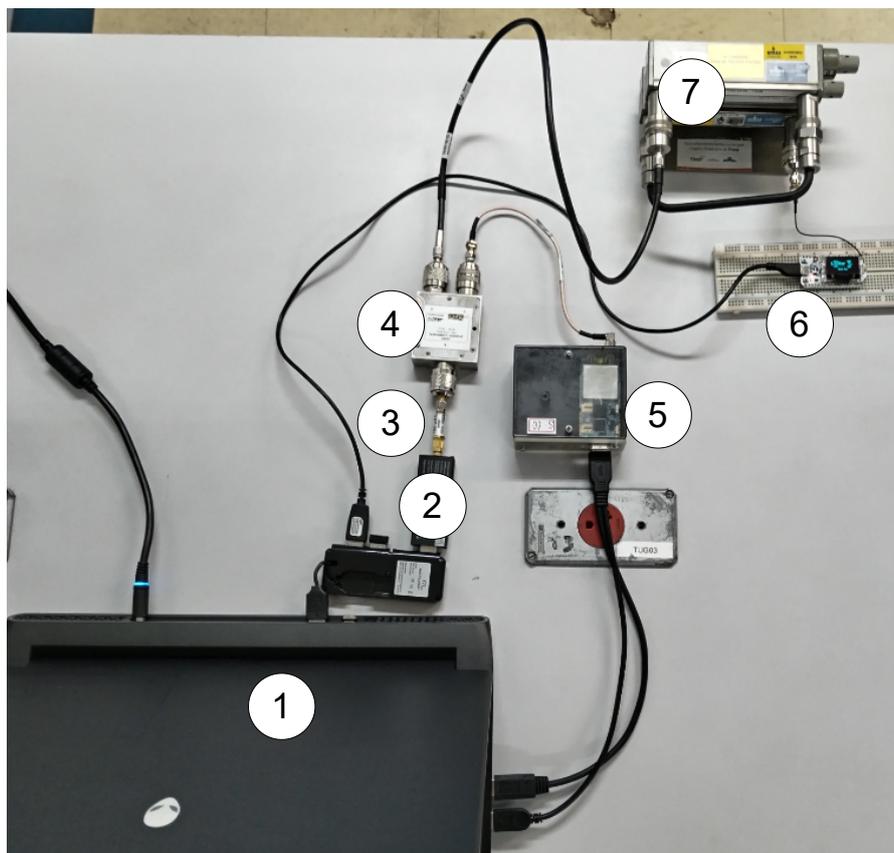
Fonte: do Autor

Após o fim de uma transmissão, conforme descrito no algoritmo do *script* supervisor para recepção LoRa por SDR (subseção 3.3.3), os valores da atenuação devem ser modificados manualmente pois os dispositivos utilizados não permitem fácil automação. Para facilitar o registro, deve ser informado o valor da relação sinal-ruído definido. Considerando os limites de SNR utilizados da Tabela 5, os atenuadores variáveis 8496A e 8494A foram configurados de forma que sua atenuação total inicial fosse de 25 dB e, entre cada ponto de interesse a atenuação fosse reduzida em 1 dB. Dessa forma a relação SNR obtida será informada em ordem crescente. Por fim, vale notar que por configuração dos parâmetros LoRa um total de 150 mil mensagens são transmitidas (5000 mensagens por 26 pontos).

Finalmente, a Figura 27 apresenta o ambiente de testes montado. Nessa figura, conforme mostra sua legenda, foi indicado cada equipamento.

²O uso de b" serve para indicar a codificação em binário da mensagem.

Figura 27: Circuito de RF montado para o experimento prático.



Legenda:

- | | | |
|-------------------------------------|-----------------|--------------------|
| ① Notebook | ② RTL-SDR v3 | ③ Atenuador 30 dB |
| ④ Combinador de Sinais | ⑤ Lime SDR Mini | ⑥ WiFi ESP LoRa v2 |
| ⑦ Conjunto de atenuadores variáveis | | |

Fonte: do Autor

4 RESULTADOS E DISCUSSÕES

Visando facilitar a leitura e a exibição dos resultados, o presente capítulo é dividido em seções de acordo com os resultados obtidos em cada etapa do trabalho. Nesse capítulo também será detalhada a quantidade de dados que foram coletados em cada etapa.

4.1 RESULTADOS DA SIMULAÇÃO

Considerando a sistemática para a simulação do protocolo LoRa apresentada na subseção 3.3.1, foram executados um total de cinquenta simulações completas (30 com caminho direto, 10 com interferência multicaminhos e 10 com *hard decoding*). Em todas essas simulações o valor de SNR variou entre -25 dB e 0 dB. A Tabela 6 resume as configurações utilizadas nessas simulações, ignorando as especificidades de cada uma.

Tabela 6: *Resumo dos parâmetros da simulação*

Parâmetro	Valores
SF	7 ou 9 ou 10
BW	125 kHz ou 250 kHz
CR	1 ou 4
Semente do ruído	$0 \leq x \leq 100$
SNR Mínimo	-25 dB
SNR Máximo	0 dB
Espaçamento entre pontos	0,2
Mensagens por ponto	1000
Mensagem	b'TCC-LoRa\n' ¹

Fonte: do Autor

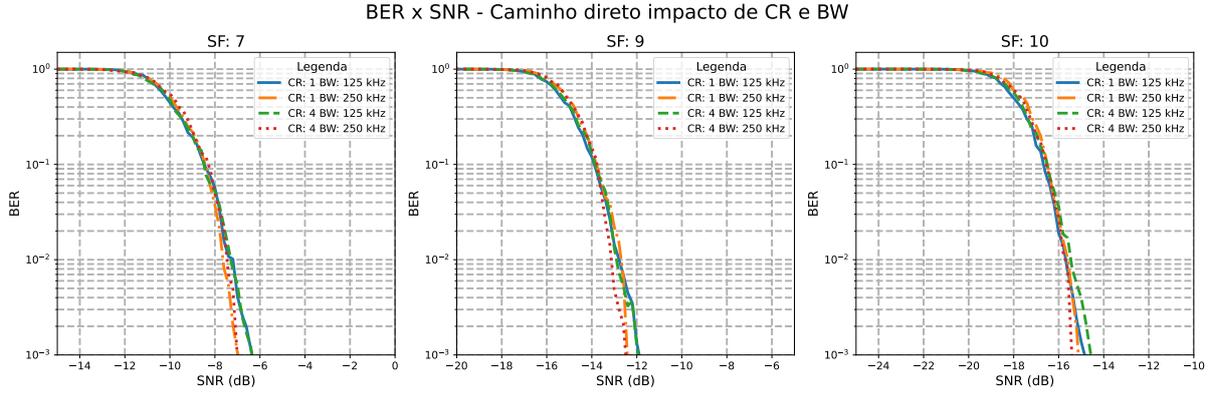
A seguir, são detalhados os resultados de cada uma dessas simulações dentro de suas respectivas subseções. A última subseção irá efetuar um comparativo geral bem como uma apreciação dos resultados da simulação.

4.1.1 Simulação Caminho direto

As primeiras 30 simulações realizadas buscaram verificar o impacto do *coding rate*, da largura de banda e do fator de espalhamento em um único canal com ruído AWGN. A Figura 28 apresenta as curvas de BER por SNR obtidas deste conjunto de dados.

¹O uso de b" serve para indicar a codificação em binário da mensagem.

Figura 28: Resultados da simulação por SF, variando BW e CR



Fonte: do Autor

Analisando a Figura 28, nota-se que as quatro combinações de CR e BW utilizadas para cada SF encontram-se significativamente próximas. Assim, para verificar se esse comportamento é consistente buscou-se referencial teórico para as curvas de erro de *bit* no protocolo LoRa.

A aproximação utilizada foi proposta por (FERREIRA DIAS; RODRIGUES DE LIMA; FRAIDENRAICH, 2019). Essa expressão aproximada para o BER foi obtida considerando a Equação 4, a qual define a probabilidade do erro de *bit* (P_b) proposta por (ELSHABRAWY; ROBERT, 2018). Essa equação é diretamente deduzida no trabalho de (ELSHABRAWY; ROBERT, 2018).

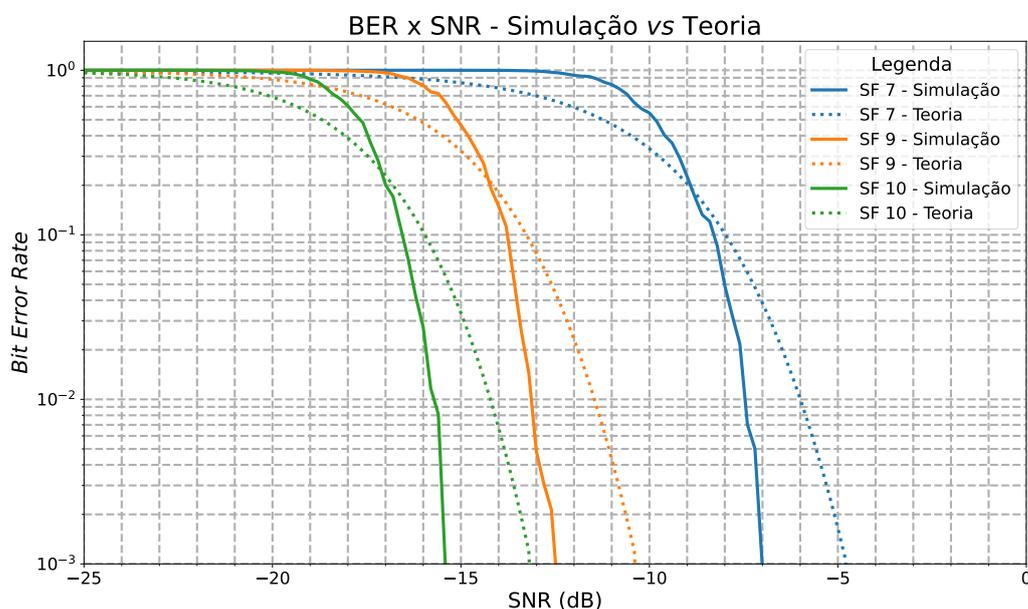
Essa expressão matemática utiliza a função $Q(\cdot)$, que é a função de distribuição acumulada do complemento da distribuição normal padrão. Por sua vez, Γ é o valor de SNR em uma escala linear e SF o fator de espalhamento. As constantes presentes no equacionamento proposto são provenientes de aproximações resultantes dos números harmônicos. Mais detalhes podem ser obtidos no trabalho original de (ELSHABRAWY; ROBERT, 2018).

$$P_b \approx 0,5 \cdot Q(\sqrt{\Gamma \cdot 2^{SF+1}} - \sqrt{1,386 \cdot SF + 1,154}) \quad (4)$$

É relevante comentar que, o estudo de (FERREIRA DIAS; RODRIGUES DE LIMA; FRAIDENRAICH, 2019) foca em modelos de canais distintos deste trabalho, mas o canal AWGN foi utilizado como curva base para comparações. Além disso, uma implementação baseada nesse artigo encontra-se disponível em (B-COM, 2024) e foi utilizada por esse trabalho uma vez que problemas de característica numérica foram observados na tentativa de implementação manual da curva desejada.

Considerando a Equação 4 é possível perceber que a probabilidade de erro de *bit* não depende do valor de CR ou de BW. O trabalho de (FERREIRA DIAS; RODRIGUES DE LIMA; FRAIDENRAICH, 2019) estende esse resultado para o comportamento do BER no canal AWGN. Portanto, é correto afirmar que as simulações efetuadas conseguiram capturar de maneira correta esse comportamento da modulação LoRa. Todavia, ao comparar a aproximação numérica proposta por (FERREIRA DIAS; RODRIGUES DE LIMA; FRAIDENRAICH, 2019) com os valores de BER obtidos das simulações, notou-se uma inconsistência significativa. A Figura 29 apresenta as curvas teóricas e a curva média obtida para cada valor de SF.

Figura 29: Comparação da curva média de cada SF com a curva teórica



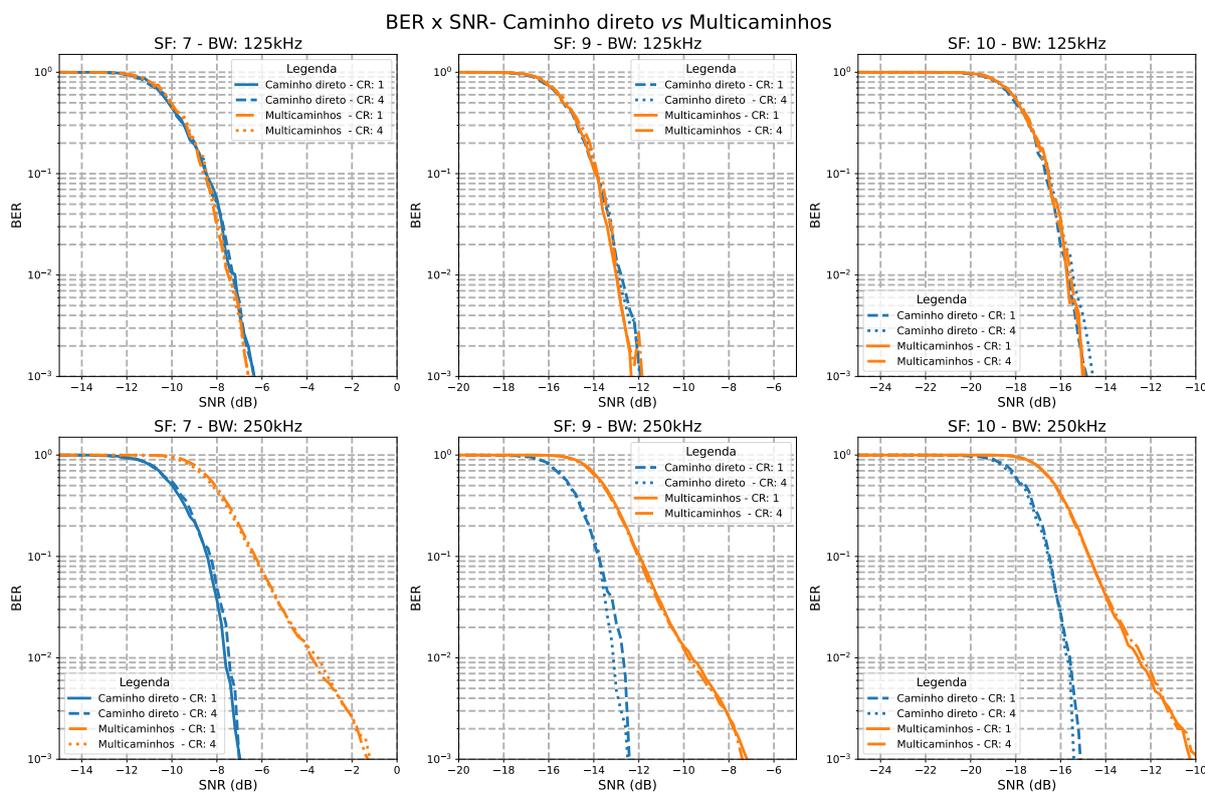
Fonte: do Autor

Um primeiro aspecto que justifica essa divergência é o número total de mensagens. Afinal, recepções de baixa probabilidade podem não ter ocorrido em função do número de mensagens da simulação. Além disso, o tamanho da mensagem (9 bytes) também é outro fator que pode influenciar no resultado. No trabalho de (TAPPAREL et al., 2020), mensagens de 64 bytes foram transmitidas. Além disso, o trabalho (TAPPAREL et al., 2020) compara seus resultados com uma implementação de Monte Carlo, mas não deixa claro quais equações foram utilizadas nessa experimentação, dificultado assim uma eventual verificação de seus resultados. Todavia, nota-se que o padrão de inclinação nas curvas obtidas por simulação é similar para os SFs analisados, indicando que as características da análise, das mensagens e do protocolo levaram a esse comportamento similar.

4.1.2 Simulação Caminhos múltiplos

No caso dessa verificação, dez simulações (com valores de sementes de ruído distintas) foram efetuadas, totalizando dez mil mensagens transmitidas por ponto de interesse. Os parâmetros utilizados foram o da Tabela 6 e considerando a simulação de caminhos múltiplos descritos seção na 3.2, Tabela 2. A Figura 30 apresenta os resultados da simulação com e sem a interferência multicaminho (resultados da subseção 4.1.1).

Figura 30: Resultados da simulação com e sem a interferência multi caminhos



Fonte: do Autor

Conforme é possível observar da Figura 30, o valor de CR não influencia significativamente na resposta obtida. Todavia, a largura de banda utilizada apresenta um impacto visível na curva de erro de *bit*. Na simulação com caminhos múltiplos e com banda de 250 kHz, observa-se que a qualidade das mensagens foi inferior — foi necessário uma relação SNR superior para atingir o mesmo erro de *bit*. Esse resultado é consistente com a teoria, uma vez que canais de banda maiores tendem a sofrer mais com a interferência de caminhos múltiplos (COLLINS; WYGLINSKI; GETZ, 2018).

Afinal, o aumento da largura de banda tende a diminuir o tempo de coerência do sinal. O tempo de coerência é uma métrica utilizada para definir por quanto tempo o modelo de canal pode ser considerado invariante no tempo em função de uma resposta ao impulso (RAPPAPOUR, 2002). Essencialmente, pode ser utilizado para caracterizar o quanto o sistema de comunicações como um todo é sensível a variações rápidas do canal. Como a análise proposta conta com o ruído AWGN em nível igual ou superior ao sinal LoRa no modelo do canal, as variações do ruído causam uma degradação significativa na recepção das mensagens.

De forma a verificar quantitativamente os resultados da Figura 30 foi calculado o erro quadrático médio entre curvas de diferente CRs e caminhos com o mesmo fator de espalhamento e largura de banda 250 kHz. Os resultados e comparações efetuadas são apresentados na Tabela 7.

Tabela 7: Matriz com o erro quadrático médio entre cada curva de erro de bit para cada fator de espalhamento com largura de banda 250 kHz e fator de codificação variável, considerando efeito de caminhos múltiplos

	Caminho				Erro Quadrático Médio		
	Único		Múltiplo		SF 7	SF 9	SF 10
	CR 1	CR 4	CR 1	CR 4	-	-	-
Curvas comparadas	✓	✓			$1,12 \cdot 10^{-4}$	$5,31 \cdot 10^{-5}$	$1,80 \cdot 10^{-4}$
	✓		✓		$3,28 \cdot 10^{-2}$	$2,46 \cdot 10^{-2}$	$2,23 \cdot 10^{-2}$
	✓			✓	$2,83 \cdot 10^{-2}$	$2,28 \cdot 10^{-2}$	$2,15 \cdot 10^{-2}$
		✓	✓		$3,28 \cdot 10^{-2}$	$2,46 \cdot 10^{-2}$	$2,23 \cdot 10^{-2}$
		✓		✓	$2,83 \cdot 10^{-2}$	$2,28 \cdot 10^{-2}$	$2,15 \cdot 10^{-2}$
			✓	✓	$1,82 \cdot 10^{-4}$	$4,00 \cdot 10^{-5}$	$1,12 \cdot 10^{-5}$

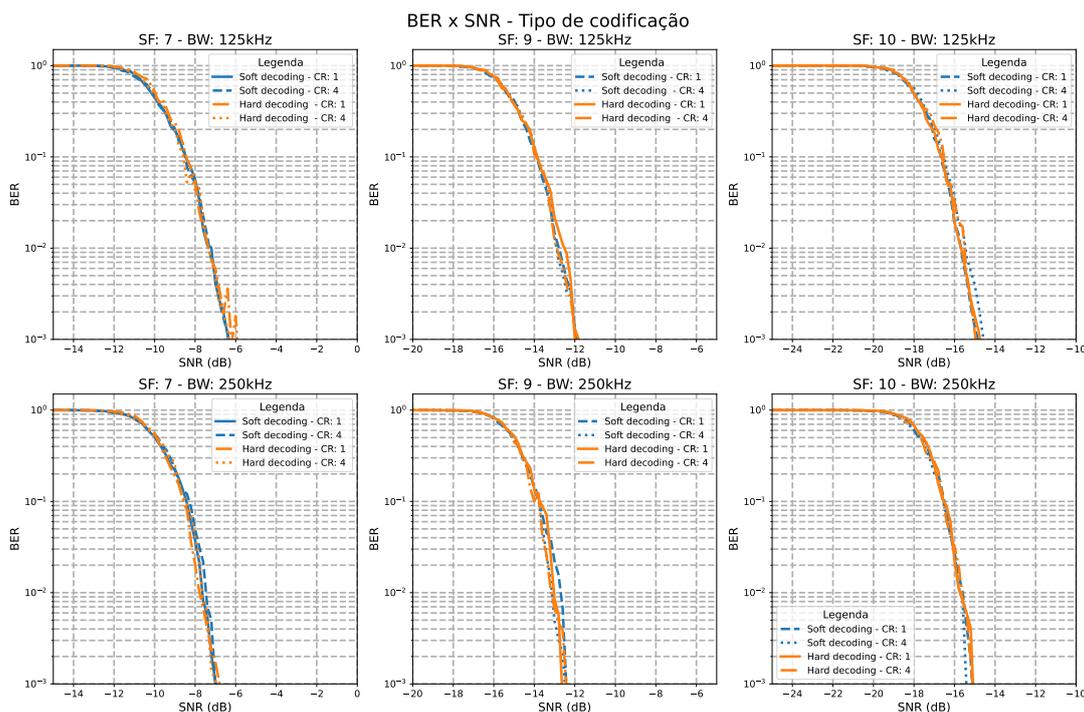
Fonte: do Autor

Na Tabela 7 os elementos marcados com "✓" são comparados entre si com o erro quadrático médio obtido na respectiva coluna. Os resultados são condizentes com a análise qualitativa. O erro quadrático médio de uma curva com o fenômeno de caminhos múltiplos contra caminhos múltiplos foi pelo menos duas ordem de grandeza maior para todos os SFs analisados. Assim, é possível dizer que existe de fato uma influência da largura de banda na qualidade das mensagens quando na presença de multicaminhos.

4.1.3 Simulação hard decoding

Outras dez simulações foram realizadas com os mesmos parâmetros da simulação da subseção 4.1.1, exceto pela forma de decisão de *bit*. As curvas obtidas com *hard decoding* e o *soft decoding* são apresentados na Figura 31. Todavia, conforme verifica-se na Figura 31 o impacto dessa não alterou significativamente o formato da curva.

Figura 31: Resultados da simulação com hard e soft decoding



Fonte: do Autor

Assim como para a subseção 4.1.2, foi montada uma tabela de erros quadráticos médios para verificar o impacto do tipo de decisão de *bit*. Nessa tabela, a comparação foi efetuada variando o tipo de codificação e largura de banda. O resultados obtidos podem ser visualizados na Tabela 8.

Tabela 8: Matriz com o erro quadrático médio entre cada curva de erro de *bit* para cada fator de espalhamento com largura de banda 250 kHz e fator de codificação variável, considerando efeito da decisão de *bit*

	<i>Decoding</i>				Erro Quadrático Médio		
	<i>Soft</i>		<i>Hard</i>		SF 7	SF 9	SF 10
	CR 1	CR 4	CR 1	CR 4	-	-	-
Curvas comparadas	✓	✓			$1,12 \cdot 10^{-4}$	$5,31 \cdot 10^{-5}$	$1,80 \cdot 10^{-4}$
	✓		✓		$2,03 \cdot 10^{-4}$	$9,05 \cdot 10^{-5}$	$1,70 \cdot 10^{-4}$
	✓			✓	$1,05 \cdot 10^{-4}$	$1,24 \cdot 10^{-4}$	$2,32 \cdot 10^{-4}$
		✓	✓		$2,03 \cdot 10^{-4}$	$9,05 \cdot 10^{-5}$	$1,70 \cdot 10^{-4}$
		✓		✓	$1,05 \cdot 10^{-4}$	$1,24 \cdot 10^{-4}$	$2,32 \cdot 10^{-4}$
			✓	✓	$3,33 \cdot 10^{-4}$	$1,99 \cdot 10^{-4}$	$3,08 \cdot 10^{-4}$

Fonte: do Autor

Os valores dos erros quadráticos médios da Tabela 8 confirmam o que a inspeção visual indica, as curvas são essencialmente iguais. Esse tipo de comportamento não era esperado, uma vez que uma decodificação dura tenderia a causar maiores erros de *bit*. Novamente, nota-se que o fato da simulação não ter sido capaz de capturar essa eventual tendência indica que a forma como os dados estão sendo capturados poderia ser aprimorada. Os dados capturados foram obtidos no final da cadeia de decodificação. Ou seja, eventuais correções de erro, por exemplo da codificação Hamming, poderiam ter atuado.

Assim, os melhores pontos para se efetuar uma comparação seriam logo após a modulação e depois da demodulação/captura do símbolo bruto. Com isso, poderia ser calculado o erro de símbolo, onde o impacto dos limites de decisão teria uma influência mais visível. Todavia, essa abordagem exigiria modificações significativas nos fluxogramas propostos, além de inserir uma nova métrica de análise. De igual forma, para não influenciar significativamente no desempenho da implementação LoRa SDR, o código fonte de (TAPPAREL et al., 2020) deveria ser modificado. Como essa modificação exige uma revalidação completa da implementação tal modificação não foi executada. Tal abordagem poderá ser desenvolvida em trabalhos futuros.

4.1.4 Considerações Gerais dos Resultados da Simulação

Considerando os resultados obtidos da simulação, de maneira geral, são consistentes com o esperado antes da execução da simulação. Algumas características específicas do comportamento do protocolo puderam ser observadas. Todavia, também notou-se que a eventual forma de aquisição dos dados para a cálculo da taxa de erro de *bits* poderia ser aprimorada. Dessa forma, em possíveis iterações sobre o código desenvolvido e novas abordagens poderão ser adotadas.

De igual forma, como foi notada a influência do efeito de multicaminhos, os fluxogramas do GNU Radio poderiam ser alterados para considerar um modelo de canal que apresente maior foco nesse tipo de interferência (por exemplo, o modelo de canal de Rayleigh e Rician). Assim, também poderiam ser efetuadas comparações com aproximações teóricas do BER do protocolo LoRa nesses modelos de canais (ELSHABRAWY; ROBERT, 2018).

4.2 RESULTADOS DO EXPERIMENTO

Nesta seção são apresentados os resultados obtidos do experimento prático. Nessa atividade, um total de 5000 mensagens por ponto de interesse foram transmitidas. O processo de variação do SNR foi efetuado conforme descrito na Subseção 3.4.2 do Capítulo 3. A Tabela 9 apresenta a configuração geral utilizada neste experimento.

Tabela 9: *Resumo dos parâmetros da atividade prática*

Parâmetro	Valores
SF	7 ou 9 ou 10
BW	125 kHz ou 250 kHz
CR	1 ou 4
SNR Mínimo	-25 dB
SNR Máximo	0 dB
Mensagens por ponto	5000
Mensagem	b'TCC-LoRa\n' ²

Fonte: do Autor

O tempo total utilizado para a coleta de dados foi de aproximadamente 65 horas. Durante algumas das configurações e pontos de interesse, notou-se que algumas exceções sem tratamento ocorreram. Esses erros eram provenientes do bloco *Frame sync* desenvolvido por (TAPPAREL et al., 2020) para identificar a recepção de uma nova mensagem (identificação de *up-chirp/down-chirp*). Essas exceções impactaram significativamente nos resultados do experimento pois, embora o sistema continue operando, um número significativo de mensagens era perdido/ignorado.

Afinal, da forma como o supervisor do experimento foi elaborado, o transmissor mandava mensagens de maneira essencialmente contínua. Com o fluxograma de recepção de mensagens em estado de erro, inúmeras mensagens transmitidas não tinham sequer a possibilidade de serem analisadas. Assim, o número total de mensagens perdidas, mesmo com um relação sinal ruído baixa era elevado.

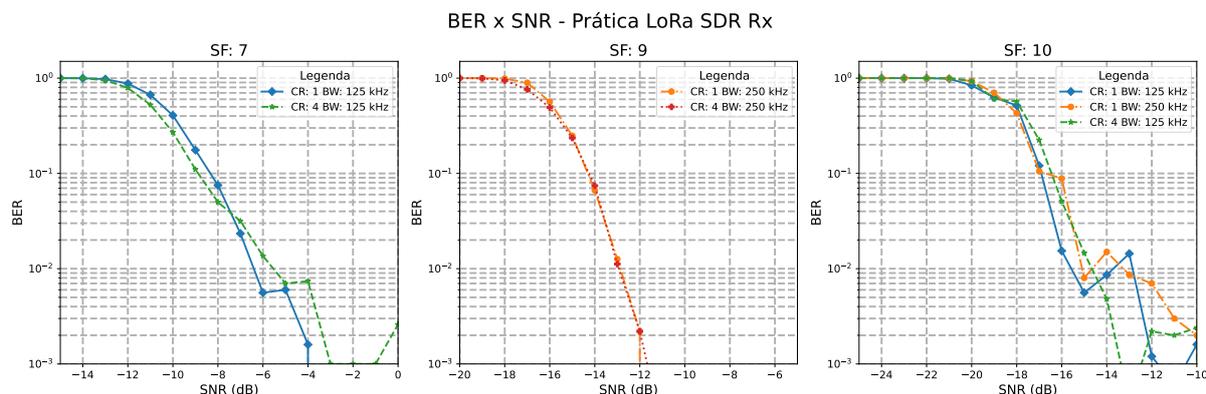
Algumas tentativas foram efetuadas buscando contornar esse problema, como: variar os parâmetros da simulação, tamanho de mensagem, semente do ruído, tempo entre mensagens, entre outros. Infelizmente, todas essas tentativas foram infrutíferas.

Dessa forma, para contornar de fato esse problema, seria necessária uma investigação e eventual alteração da implementação do bloco *Frame sync*. Todavia, essa modificação não se encontrava no escopo original deste trabalho. Assim, os resultados considerados inválidos foram desconsiderados da presente análise. As configurações que apresentaram esse comportamento encontram-se listadas abaixo:

- SF 7: 125 kHz com CR 1 e 4;
- SF 9: 250 kHz com CR 1 e 4;
- SF 10: 125 kHz com CR 1 e 4.

A Figura 32 apresenta os resultados brutos do experimento prático.

²O uso de b' serve para indicar a codificação em binário da mensagem.

Figura 32: Resultados brutos do experimento prático

Fonte: do Autor

Conforme é possível observar na Figura 32, as curvas obtidas não são necessariamente suaves (exceto pelas obtidas com SF9). O principal motivo para esse comportamento é a própria distância entre os pontos de interesse e o número de mensagens enviadas. Todavia, outro aspecto que influencia nesse comportamento é a própria natureza aleatória do ruído introduzido. Buscando minimizar esses efeitos, poderiam ser efetuados inúmeros outros experimentos e posteriormente combiná-los em um única curva utilizando um procedimento similar ao das simulações. Todavia, devido a restrições do cronograma proposto, esses novos experimentos não foram executados.

Assim, optou-se por efetuar uma regressão de forma a encontrar uma curva que melhor se ajusta-se aos pontos obtidos. A função proposta para essa curva foi uma função sigmoide. Esse tipo de função apresenta uma natureza consistente com a resposta esperada da curva $BER \times SNR$. A expressão matemática genérica para as curvas $BER \times SNR$ que serão traçadas é apresentada na Equação 5. Nessa equação x é o valor de SNR e, a o valor que controla a inclinação da curva e b o valor do ponto médio da curva.

$$BER(x) = \frac{1}{1 + e^{-a(x-b)}} \quad (5)$$

Portanto, é necessário definir os parâmetros a e b para cada uma das curvas. Para isso foi utilizado o método de regressão por mínimos quadrados não-linear (*non-linear least squares*). Os resultados obtidos estão agrupados na Tabela 10.

Tabela 10: Parâmetros das curvas sigmóides ajustadas aos dados do experimento

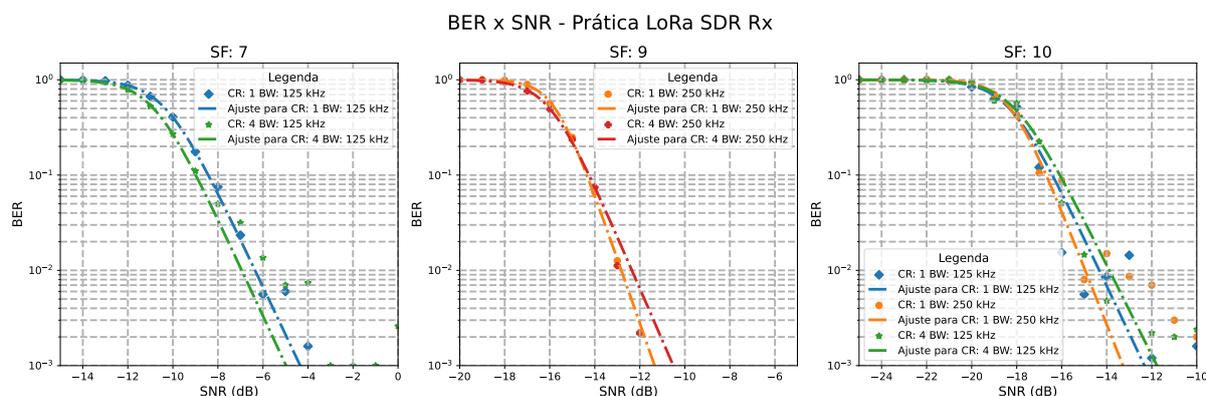
Parâmetro	SF 7		SF 9		SF 10			
	125 kHz		250 kHz		125 kHz		250 kHz	
	CR 1	CR 4	CR 1	CR 4	CR 1	CR 4	CR 1	CR 4
a	-1.1521	-1.1730	-1.5613	1.2569	-1.1505	-1.0911	-1.3785	-
b	-10.3385	-10.8600	-15.7644	-16.00	-18.32	-18.0976	-18.2877	-

Fonte: do Autor

A Figura 33 apresenta os dados dos experimentos, assim como as curvas ajustadas. Considerando as curvas com fator de espalhamento de 7, nota-se que elas apresentam uma divergência inferior a 1 dB. Todavia, essa divergência não pode ser considerada significativa ou mesmo atribuída ao diferente tipo de codificação utilizada. Afinal, o próprio

transmissor LoRa utilizado apresenta uma variação de saída de ± 1 dBm. Além disso, o baixo número de mensagens também permite justificar essa divergência. Tendo em vista essas condições de contorno, é possível afirmar que, assim como na simulação, o valor de CR não influenciou significativamente no desempenho. Esse resultado é coerente, uma vez que CR superiores são utilizados para tornar a transmissão mais robusta em relação aos efeitos de desvanecimento (ELSHABRAWY; ROBERT, 2018), aspecto não analisado nesse experimento. Um comportamento similar as curvas de fator de espalhamento 7 pode ser observado nas curvas de fator de espalhamento 9.

Figura 33: Curvas ajustadas para os dados do experimento prático

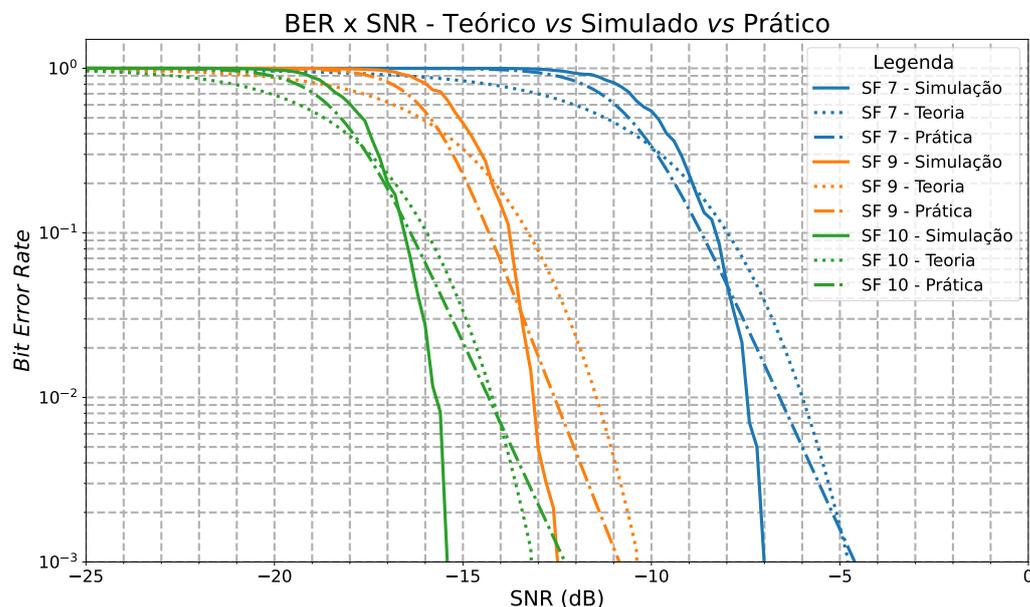


Fonte: do Autor

Por sua vez, as curvas com fator de espalhamento 10 apresentam uma divergência significativamente maior. Esse fato é decorrente dos dados coletados com SF 10, apresentarem uma dispersão maior onde o padrão de ajuste não é de fácil visualização. Nesse caso, embora o ajuste proposto seja consistente para os demais fatores de espalhamento, ele pode não ser recomendado para essa configuração. Todavia, devido ao tempo que foi planejado para a atividade prática, novos experimentos focando na coleta de dados para esse SF não foram realizados. Logo, como as simulações propostas, a teoria e prática indicaram que as curvas de $BER \times SNR$ em um canal AWGN não dependem da largura de banda ou codificação, a divergência encontrada foi considerada uma anomalia proveniente do baixo número de dados coletados.

Considerando esses aspectos, foi efetuada a comparação entre a curva média encontrada para cada SF com os resultados teóricos e da simulação. A Figura 34 apresenta essas curvas para cada um dos SFs analisados.

Figura 34: Curvas $BER \times SNR$ para as aproximações teórica, da simulação e dos dados práticos



Fonte: do Autor

Analisando as curvas da Figura 34 nota-se que existe uma discrepância de aproximadamente 2 dB entre as curvas teóricas e as curvas dos experimentos. Essa divergência pode ser justificada pelo número de mensagens analisadas assim como a qualidade do transmissor e receptor envolvidos. De igual forma, a aproximação utilizada também pode inserir certo viés sobre os dados. Vale notar que a inclinação obtida entre as curvas provenientes dos experimentos e da teoria são mais próximas entre si do que as curvas da simulação. Esse aspecto é o mais relevante, porque percebe-se que a implementação SDR de (TAPPAREL et al., 2020), do ponto de vista do receptor, capta com sucesso as mensagens enviadas e preserva as características de resistência ao ruído da modulação LoRa.

Além disso, vale notar que a curva teórica apresenta o melhor desempenho com o SNR mais negativo. Esse aspecto é condizente, uma vez que, devido ao número de mensagens transmitidas, a recepção com sucesso nesses valores é estatisticamente menos provável. Também nota-se que durante certos intervalos de SNR o desempenho medido na prática foi superior ao teórico. Novamente, no número de mensagens enviadas bem como a forma de ajustes das sigmóides traçadas inseriram esse viés.

Por sua vez, a divergência das curvaturas de simulação com as curvaturas provenientes da prática indica que, possivelmente, a implementação SDR de (TAPPAREL et al., 2020) do ponto de vista do transmissor apresenta algumas divergências em relação a modulação proprietária da *Semtech*. Para verificar esse aspecto, experimentos futuros onde tanto o receptor como o transmissor LoRa são SDRs poderiam ser efetuados. Esse novo experimento, aliado a um número maior de mensagens transmitidas em uma metodologia análoga a utilizada na etapa de simulação poderia verificar essa hipótese.

4.2.1 Considerações Gerais dos Resultados do Experimento

Os resultados obtidos nos experimentos, de maneira geral, podem ser considerados satisfatórios. Algumas características específicas do comportamento do protocolo puderam ser novamente observadas. Todavia, também notou-se que o número de mensagens e experimentos por ponto de interesse poderia ser repensado. Principalmente, a implementação SDR de (TAPPAREL et al., 2020) poderia ser modificada buscando solucionar o problema observado no bloco *Frame Sync*.

De igual forma, notou-se que o valor de SNR é calculado localmente dentro desse próprio bloco. Assim, seria de interesse alterar a implementação para que esse valor fosse disponibilizado para níveis superiores, de forma a comparar o valor de SNR percebido pelo sistema de identificação de mensagens em relação ao proposto no circuito de RF utilizado. Também poderia ser utilizada a funcionalidade de SNR detectado por um transceptor LoRa da *Semtech* que recebesse o mesmo sinal do receptor SDR, para verificar eventuais divergências.

Finalmente, o sistema do experimento bem como o código supervisorio poderiam ser modificados para utilizar um atenuador variável programável. Esse equipamento facilitaria a automatização dessa etapa do experimento além de possibilitar que o mesmo fosse executado com passos menores. Alternativamente, a intensidade do ruído de gerado pelo Lime SDR Mini poderia ser alterada, eliminando a necessidade de um atenuador variável, caso verificada a precisão na troca de potências de transmissão por parte desse equipamento.

5 CONCLUSÃO

O presente trabalho investigou a robustez do protocolo LoRa ao ruído AWGN de maneira simulada e prática utilizando plataformas SDR. Para isso, foi selecionada a implementação da modulação LoRa não proprietária proposta por (TAPPAREL et al., 2020). Dessa forma, efetuou-se a devida revisão teórica dos conceitos fundamentais de transmissão sem fio, SDRs e das principais propriedades da modulação LoRa.

A principal métrica analisada foi a análise das curvas de erro de *bit* por relação sinal-ruído (BER x SNR). Para a obtenção dos dados necessários para a construção dessas curvas, foram construídos fluxogramas na plataforma GNU Radio para a execução do processamento digital necessário. De igual forma, *scripts* supervisórios foram desenvolvidos para automatizar o processo de coleta de dados da simulação e da prática. A aproximação fechada para a curva de BER x SNR da modulação LoRa em canal AWGN proposta por (ELSHABRAWY; ROBERT, 2018) foi considerada referência do comportamento esperado.

Ao longo desse desenvolvimento, foram observados alguns pontos de melhoria na implementação proposta por (TAPPAREL et al., 2020). Todavia, devido ao escopo do trabalho essas melhorias não foram executadas. Entretanto, esses aspectos da implementação da modulação LoRa SDR utilizada contribuíram para a análise dos fatores de espalhamento (SF) 7, 9 e 10, com larguras de banda (BW) de 125 kHz e 250 kHz. Outro elemento que foi variado nas análises foi a codificação de *bit* (CR) a qual assumiu o valor de 1 ou 4.

Os resultados da simulação possibilitaram observação de algumas características do protocolo. Entre elas, verificou-se que sua robustez ao ruído em canal AWGN não apresenta um vínculo tão forte em relação a largura de banda utilizada e uma dependência significativa do fator de espalhamento conforme proposto por (ELSHABRAWY; ROBERT, 2018). Todavia, as inclinações obtidas das curvas BER x SNR provenientes da simulação foram significativamente divergentes em relação as curvas teóricas. Um aspecto que foi considerado como motivo para esse tipo de comportamento foi a forma como o supervisório e fluxogramas do GNU Radio atuavam em conjunto para a coleta dos dados. Afinal, a coleta era realizada no final da cadeia de processamento de símbolos. Outra hipótese para esse comportamento é o tamanho e número de mensagens utilizadas na simulação.

De igual forma, foi levantada a hipótese de que a implementação da modulação e/ou demodulação LoRa proposta por (TAPPAREL et al., 2020) pode ser distinta (embora compatível) com a modulação LoRa proprietária. Essa eventual divergência seria justamente mais visível em cenários de verificação de desempenho conforme o proposto neste trabalho.

Analisando os resultados do experimento, no qual foi utilizado um chip comercial LoRa como transmissor e um SDR como receptor, notou-se que a curvatura obtida após um ajuste dos dados foi mais próxima da abordagem teórica. De igual forma, a dependência significativa do fator de espalhamento no traçado das curvas BER x SNR também foi observada. Assim, a hipótese de discrepância entre a modulação LoRa proprietária e a implementação SDR ganhou mais um argumento. Além disso, foram encontradas algumas dificuldades em executar o experimento com configurações específicas. Esses aspectos

indicam que existem outros elementos da implementação de (TAPPAREL et al., 2020) que poderiam ser melhorados em trabalho futuros.

De igual forma, em trabalhos futuros, uma reestruturação do experimento poderá ser efetuada para possibilitar melhor controle das variáveis envolvidas e possibilitar maior controle estatístico sobre o experimento. Com isso, aumentaria-se a consistência das análises propostas facilitaria-se a reprodução dos resultados encontrados.

Ainda assim, foi possível perceber a robustez do protocolo LoRa no canal AWGN, uma vez que seu desempenho com relações sinal-ruído onde a maioria dos protocolos não consegue atuar foi observada. Como trabalhos futuros, outros experimentos práticos considerando um transmissor LoRa SDR e um receptor LoRa SDR poderiam ser efetuados. De igual forma, um teste utilizando um transmissor e receptor LoRa com a modulação proprietária seria interessante para também ser utilizada como comparativo. Os testes poderiam ser mais automatizados pelo uso de um atenuador variável, facilitando assim novos experimentos que variassem de diversas outras maneiras o número de mensagens, e parâmetros de transmissão. Outro aspecto que também poderia ser investigado é a influência e a resposta de diferentes tipos de modelo de canal. Adicionalmente, os supervisórios e fluxogramas desenvolvidos podem ser modificados para permitir a coleta de outras métricas de análise que se fizerem necessárias.

REFERÊNCIAS

- ACETO, G.; PERSICO, V.; PESCAPÉ, A. A Survey on Information and Communication Technologies for Industry 4.0: State-of-the-Art, Taxonomies, Perspectives, and Challenges. *IEEE Communications Surveys & Tutorials*, v. 21, n. 4, p. 3467–3501, 2019. DOI: 10.1109/COMST.2019.2938259.
- AGILENT TECHNOLOGIES. *Agilent 8494/95/96A/B Attenuators Operating and Service Manual*. [S.l.]. Acessado em 7 de janeiro de 2024. Disponível em: <<http://legacy.libertytest.com/assetmanager/uploaded/pdf-2012125-164819-8496a-b.pdf>>.
- AGILENT TECHNOLOGIES. *Folha de Especificações Agilent N9912A*. [S.l.]. Acessado em 17 de janeiro de 2024. Disponível em: <https://www.aaatesters.com/pub/media/datasheets/agilent_n9912a_specifications_spec_sheet_1h310.pdf>.
- AGILENT TECHNOLOGIES. *Guia do Usuário dos Sensores de Potência da Série U2000A*. [S.l.]. Acessado em 17 de janeiro de 2024. Disponível em: <https://www.instrumex.de/datasheet/U2000A_Serie.PDF>.
- AHMED, M. A. et al. LoRa Based IoT Platform for Remote Monitoring of Large-Scale Agriculture Farms in Chile. *Sensors*, MDPI AG, v. 22, n. 8, p. 2824, abr. 2022. DOI: 10.3390/s22082824. Disponível em: <<https://doi.org/10.3390/s22082824>>.
- AKEELA, R.; DEZFOULI, B. Software-defined Radios: Architecture, state-of-the-art, and challenges. *Computer Communications*, v. 128, p. 106–125, 2018. ISSN 0140-3664. DOI: <https://doi.org/10.1016/j.comcom.2018.07.012>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0140366418302937>>.
- ALLIANCE, L. *LoRa Alliance*. Disponível em: <<https://lora-alliance.org/>>. Acesso em: 5 ago. 2023.
- ALLIANCE, L. *TS001-1.0.4 LoRaWAN® L2 1.0.4 Specification*. Fremont - US, 2020. P. 90.
- ANATEL. *ANATEL - Institucional*. Disponível em: <<https://www.gov.br/anatel/pt-br/aceso-a-informacao/institucional>>. Acesso em: 24 jul. 2023.
- ANATEL. *INTERNET DAS COISAS IoT/M2M*. Edição: ANATEL. [Cartilha Online; Acesso 10 Julho de 2023]. Mar. 2023b. Disponível em: <<https://sistemas.anatel.gov.br/anexar-api/publico/anexos/download/a028ab5cc4e3f97442830bba0c8bd1dd/>>.
- ARDUINO. *Arduino Software*. Disponível em: <<https://www.arduino.cc/en/software>>. Acesso em: 7 jan. 2024.
- AUTOMATION, H. *WiFi LoRa 32 (v2)*. Fremont - US, 2020. P. 13.

- B-COM. *Taxa de Erro de Bit (BER) para Tecnologia LoRa*. [S.l.: s.n.], 2024. https://github.com/b-com/ber_LoRa/tree/master. Acesso em: 10 de janeiro de 2024.
- CHEN, Y. et al. A low power software-defined-radio baseband processor for the Internet of Things. In: 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA). [S.l.: s.n.], 2016. P. 40–51. DOI: 10.1109/HPCA.2016.7446052.
- CLIFFORD, C. J. *CubicSDR 0.2.5 - Release Builds*. Disponível em: <<https://github.com/cjcliffe/CubicSDR/releases/tag/0.2.5>>. Acesso em: 7 jan. 2024.
- COLLINS, T. F.; WYGLINSKI, A. M.; GETZ, R. *Software-Defined Radio for Engineers*. [S.l.]: Analog Devices, 2018. Disponível em: <<https://www.analog.com/media/en/training-seminars/design-handbooks/Software-Defined-Radio-for-Engineers-2018/SDR4Engineers.pdf>>.
- COOK, P.; BONSER, W. Architectural overview of the SPEAKeasy system. *IEEE Journal on Selected Areas in Communications*, v. 17, n. 4, p. 650–661, 1999. DOI: 10.1109/49.761042.
- DELL. *Notebook Gamer Alienware M15 R7*. Disponível em: <<https://www.dell.com/pt-br/shop/povw/alienware-m15-r7-gaming-laptop>>. Acesso em: 6 ago. 2023.
- ELSHABRAWY, T.; ROBERT, J. Closed-Form Approximation of LoRa Modulation BER Performance. *IEEE Communications Letters*, v. 22, n. 9, p. 1778–1781, 2018. DOI: 10.1109/LCOMM.2018.2849718.
- ESPRESSIF. *ESP32*. Disponível em: <<https://www.espressif.com/en/products/socs/esp32>>. Acesso em: 6 ago. 2023.
- ETTUS. *GNU Radio opens an unseen world*. Disponível em: <<https://www.ettus.com/>>. Acesso em: 6 ago. 2023.
- FERREIRA DIAS, C.; RODRIGUES DE LIMA, E.; FRAIDENRAICH, G. Bit Error Rate Closed-Form Expressions for LoRa Systems under Nakagami and Rice Fading Channels. *Sensors*, v. 19, n. 20, 2019. ISSN 1424-8220. DOI: 10.3390/s19204412. Disponível em: <<https://www.mdpi.com/1424-8220/19/20/4412>>.
- FILHO, H. V.; GALDINO, J. F.; MOURA, D. F. C. Pesquisa e Desenvolvimento de Produtos de Defesa: Reflexões e Fatos sobre o Projeto Rádio Definido por Software do Ministério da Defesa à luz do Modelo de Inovação em Tríplice Hélice. *REVISTA MILITAR DE CIÊNCIA E TECNOLOGIA*, v. 34, n. 1, p. 6–19, 2017.
- GADGETS, G. S. *HackRF One*. Disponível em: <<https://greatscottgadgets.com/hackrf/one/>>. Acesso em: 1 ago. 2023.
- GARG, V. K. CHAPTER 23 - Fourth Generation Systems and New Wireless Technologies. In: GARG, V. K. (Ed.). *Wireless Communications and Networking*. Burlington: Morgan Kaufmann, 2007. (The Morgan Kaufmann Series in Networking). P. 23-1-23–22. DOI: <https://doi.org/10.1016/B978-012373580-5/50057-0>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780123735805500570>>.
- GNU RADIO WIKI. *Installing GNU Radio*. Acessado em 18 de janeiro de 2024. Disponível em: <<https://wiki.gnuradio.org/index.php?title=InstallingGR>>.
- GTA-UFRJ. *Modulação e protocolo LoRaWAN*. Disponível em: <<https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/lora/modulacao.html>>. Acesso em: 5 ago. 2023.

- HELTEC AUTOMATION. *Arduino library for Heltec ESP32 (or ESP32+LoRa) based boards*. Disponível em: <https://github.com/HelTecAutomation/Heltec_ESP32>. Acesso em: 7 jan. 2024.
- HELTEC AUTOMATION. *Heltec Automation*. Disponível em: <<https://heltec.org/>>. Acesso em: 7 jan. 2024.
- ITU. *About International Telecommunication Union (ITU)*. Disponível em: <<https://www.itu.int/en/about/Pages/default.aspx>>. Acesso em: 24 jul. 2023.
- ITU-R. *Definitions of Software Defined Radio (SDR) and Cognitive Radio System (CRS) ITU-R SM.2152 (09/2009)*. [S.l.], 2009.
- JBM INSTRUMENTOS. *Divisor Combinador de Potência de RF Wilkinson 700 MHz - 2.7 GHz Duas Portas*. Disponível em: <<https://www.jbminstrumentos.com.br/produto/divisor-combinador-de-potencia-de-rf-wilkinson-700-mhz-2-7-ghz-duas-portas-12/>>. Acesso em: 17 jan. 2024.
- KHUTSOANE, O.; ISONG, B.; ABU-MAHFOUZ, A. M. IoT devices and applications based on LoRa/LoRaWAN. In: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. [S.l.: s.n.], 2017. P. 6107–6112. DOI: 10.1109/IECON.2017.8217061.
- LACKEY, R.; UPMAL, D. Speakeasy: the military software radio. *IEEE Communications Magazine*, v. 33, n. 5, p. 56–61, 1995. DOI: 10.1109/35.392998.
- LIBELIUM. *Lora Networking Guide - Transmission Modes*. Disponível em: <https://development.libelium.com/lora_networking_guide/transmission-modes>. Acesso em: 5 ago. 2023.
- LIME MICROSYSTEMS. *LimeSDR Mini*. Disponível em: <<https://limemicro.com/products/boards/limesdr-mini/>>. Acesso em: 7 jan. 2024.
- LUO, Z.; ZHANG, W. The Simulation Models for Rayleigh Fading Channels. In: *2007 Second International Conference on Communications and Networking in China*. [S.l.: s.n.], 2007. P. 1158–1163. DOI: 10.1109/CHINACOM.2007.4469592.
- MCHENRY, M. A.; ROBERTSON, D.; MATHESON, R. A New Adaptive Analog Test and Diagnosis System. *IEEE Spectrum*, ago. 2015.
- MICRO, T. *Software-Defined Radio (SDR)*. Disponível em: <<https://www.trendmicro.com/vinfo/us/security/definition/sdr>>. Acesso em: 24 jul. 2023.
- MICROSYSTEMS, L. *LimeSDR*. Disponível em: <<https://limemicro.com/products/boards/limesdr/>>. Acesso em: 6 ago. 2023.
- MINI-CIRCUITS. *Datasheet do Produto VAT-30+*. [S.l.]. Acessado em 7 de janeiro de 2024. Disponível em: <<https://www.minicircuits.com/pdfs/VAT-30+.pdf>>.
- MYRIAD-RF COMMUNITY. *LimeSDR-Mini - Myriad-RF Wiki*. Disponível em: <<https://wiki.myriadrf.org/LimeSDR-Mini>>. Acesso em: 14 jan. 2024.
- MYRIADRF. *LimeSuite: Driver and GUI for LMS7002M-based SDR platforms*. Disponível em: <<https://github.com/myriadrf/LimeSuite>>. Acesso em: 7 jan. 2024.

- NEWMAN, T. R.; EVANS, J. B.; WYGLINSKI, A. M. Chapter 7 - Reconfiguration, adaptation, and optimization. In: WYGLINSKI, A. M.; NEKOVEE, M.; HOU, T. (Ed.). *Cognitive Radio Communications and Networks*. Oxford: Academic Press, 2010. P. 177–198. ISBN 978-0-12-374715-0. DOI: <https://doi.org/10.1016/B978-0-12-374715-0.00007-1>. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780123747150000071>.
- NORTON, Q. *GNU Radio opens an unseen world*. Disponível em: <https://www.wired.com/2006/06/gnu-radio-opens-an-unseen-world/>. Acesso em: 6 ago. 2023.
- ORACLE. *OSI Model*. Disponível em: <https://docs.oracle.com/cd/E19504-01/802-5886/intro-45828/index.html>. Acesso em: 5 ago. 2023.
- OSMOCOM. *rtl-sdr v2.0.1*. Disponível em: <https://gitea.osmocom.org/sdr/rtl-sdr/releases/tag/v2.0.1>. Acesso em: 7 jan. 2024.
- PAWLAN, J. An introduction to Software Defined Radio. In: 2015 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS). [S.l.: s.n.], 2015. P. 1–1. DOI: 10.1109/COMCAS.2015.7360430.
- PROAKIS, J. G.; SALEHI, M. *Communication Systems Engineering*. 2. ed. [S.l.]: Pearson, 2001. ISBN 978-0130617934.
- PROJECT, G. R. *What is GNU Radio?* Disponível em: <https://www.gnuradio.org/about/>. Acesso em: 16 jul. 2023.
- PROJECT, G. R. *GNU Radio Manual and C++ API Reference*. Disponível em: <https://www.gnuradio.org/doc/doxygen/index.html>. Acesso em: 6 ago. 2023.
- PROJECT, G. R. *GNU Radio Project Organization*. Disponível em: <https://www.gnuradio.org/org/organization/>. Acesso em: 6 ago. 2023.
- PROJECT, G. R. *What is GNU Radio*. Disponível em: https://wiki.gnuradio.org/index.php/Main_Page/. Acesso em: 6 ago. 2023.
- PYTHON SOFTWARE FOUNDATION. *Python Release Python 3.10.12*. Disponível em: <https://www.python.org/downloads/release/python-31012/>. Acesso em: 7 jan. 2024.
- RAPPAPORT, T. S. *Wireless Communications: Principles and Practice*. 2. ed. [S.l.]: Prentice Hall, 2002.
- RIZZI, M. et al. Using LoRa for industrial wireless networks. In: 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS). [S.l.: s.n.], 2017. P. 1–4. DOI: 10.1109/WFCS.2017.7991972.
- RTL-SDR. *RTL-SDR*. Disponível em: <https://www.rtl-sdr.com/>. Acesso em: 6 ago. 2023.
- SANDEEP. *Arduino LoRa*. Disponível em: <https://github.com/sandeepmistry/arduino-LoRa>. Acesso em: 7 jan. 2024.
- SCHER, A. D. *RTL-SDR: Block Diagram and Information*. Disponível em: https://aaronscher.com/wireless_com_SDR/rtl_sdr_info.html. Acesso em: 1 ago. 2023.
- SEMTECH. *LoRa*. Disponível em: <https://www.semtech.com/lora>. Acesso em: 5 ago. 2023.

- SEMTECH. *SX1276*. Disponível em: <<https://www.semtech.com/products/wireless-rf/lora-connect/sx1276>>. Acesso em: 6 ago. 2023.
- SEMTECH. *What are LoRa and LoRaWAN*. Disponível em: <<https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>>. Acesso em: 24 jul. 2023.
- SEMTECH. *What is LoRa?* Disponível em: <<https://www.semtech.com/lora/what-is-lora>>. Acesso em: 16 jul. 2023.
- SFORZA, F. *COMMUNICATIONS SYSTEM*. Mar. 2013. US8406275. Disponível em: <https://patentimages.storage.googleapis.com/13/fc/f2/b54df646977fa2/US8406275.pdf>.
- AL-SUDANI, H.; THABIT, A. A.; DALVEREN, Y. Cognitive Radio and Its Applications in the New Trend of Communication System: A Review. In: 2022 5th International Conference on Engineering Technology and its Applications (IICETA). [S.l.: s.n.], 2022. P. 419–423. DOI: 10.1109/IICETA54559.2022.9888674.
- TALUKDAR, J. et al. Implementation of SNR estimation based energy detection on USRP and GNU radio for cognitive radio networks. In: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). [S.l.: s.n.], 2017. P. 304–308. DOI: 10.1109/WiSPNET.2017.8299767.
- TANENBAUM, A.; FEAMSTER, N. *Redes de Computadores*. [S.l.]: Bookman, 2021. P. 624. ISBN 978-8582605608.
- TAPPAREL, J. et al. An Open-Source LoRa Physical Layer Prototype on GNU Radio. In: 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). [S.l.: s.n.], 2020. P. 1–5. DOI: 10.1109/SPAWC48557.2020.9154273.
- VANGELISTA, L. Frequency Shift Chirp Modulation: The LoRa Modulation. *IEEE Signal Processing Letters*, v. 24, n. 12, p. 1818–1821, 2017. DOI: 10.1109/LSP.2017.2762960.
- VISWANATHAN, M. *Hard and Soft Decision Decoding*. Acessado em 18 de janeiro de 2024. Dez. 2009. Disponível em: <<https://www.gaussianwaves.com/2009/12/hard-and-soft-decision-decoding-2/>>.
- XPLORE, I. *IEEE Xplore*. Disponível em: <<https://ieeexplore.ieee.org/>>. Acesso em: 6 ago. 2023.