



Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica



Trabalho de Conclusão de Curso

Desenvolvimento de uma Bancada de Emulação de Redes de Comunicação

Romario Henrique Souza Lage

João Monlevade, MG
2023

Romario Henrique Souza Lage

Desenvolvimento de uma Bancada de Emulação de Redes de Comunicação

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.

Orientador: Prof. Márcio Feliciano Braga

Universidade Federal de Ouro Preto
João Monlevade
2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

L174d Lage, Romario Henrique Souza.
Desenvolvimento de uma bancada de emulação de redes de
comunicação. [manuscrito] / Romario Henrique Souza Lage. - 2023.
55 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Marcio Feliciano Braga.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia
Elétrica .

1. Comunicações digitais. 2. Emuladores (Programas de computador).
3. Protocolo de aplicação sem fio. 4. Redes de computadores -
comunicação. 5. Redes de sensores sem fio. I. Braga, Marcio Feliciano. II.
Universidade Federal de Ouro Preto. III. Título.

CDU 004:621.3

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



FOLHA DE APROVAÇÃO

Romario Henrique Souza Lage

Desenvolvimento de uma bancada de emulação de redes de comunicação

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia Elétrica

Aprovada em 30 de agosto de 2023.

Membros da banca

Dr. Márcio Feliciano Braga — Orientador — Universidade Federal de Ouro Preto
Dr. Marcelo Moreira Tiago — Convidado — Universidade Federal de Ouro Preto
Dra. Wendy Yadira Eras Herrera — Convidada — Universidade Federal de Ouro Preto

Márcio Feliciano Braga, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 24/09/2023.



Documento assinado eletronicamente por **Marcio Feliciano Braga, PROFESSOR DE MAGISTERIO SUPERIOR**, em 24/09/2023, às 19:18, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0594058** e o código CRC **692EEC5F**.

Dedico este trabalho aos meus pais, namorada e amigos, que me apoiam e me dão forças todos os dias.

Agradecimentos

Agradeço a Deus por iluminar a minha jornada e por não me deixar desistir desse sonho.

Aos meus pais, Tarcísio e Maximina, pela confiança e todo o apoio, sem vocês isso não teria acontecido.

A minha namorada, Aline, por trilhar junto comigo nessa caminhada, sempre me apoiando, incentivando e me dando confiança.

Aos familiares e amigos, fica o meu agradecimento especial também por fazerem parte dessa conquista.

Aos amigos da UFOP, vocês foram fundamentais nessa caminhada.

Aos professores da UFOP, em especial Prof. Dr. Márcio Feliciano Braga e Prof. Dr. Marcelo Moreira Tiago pela paciência, ensinamentos repassados e por contribuírem para a minha formação.

“O único lugar onde o Sucesso vem antes do Trabalho é no dicionário”
Albert Einstein

Resumo

A velocidade com que as tecnologias têm avançado nas últimas décadas afetam diretamente a maneira como vivemos, trabalhamos e nos comunicamos. No entanto, essas tecnologias enfrentaram uma série de desafios complexos antes de serem inovadoras em grande escala. As incertezas sobre o seu funcionamento, necessidade de atender requisitos impostos e a preocupação com a segurança são apenas alguns dos obstáculos que devem ser superados. Com isso, este trabalho visa desenvolver um ambiente de emulação de canal de rádio que permita considerar as perturbações comumente sofridas por sinais de radiofrequência, sem a necessidade de lidar com as dificuldades decorrentes da implementação e controle de redes em ambiente real. Para realizar a construção, tem-se a necessidade de estudos voltados para as Rede de Sensores Sem Fio (RSSF), protocolos de comunicação e modelos de propagação do sinal de Radiofrequência (RF). A bancada desempenha um papel essencial no avanço das RSSF, permitindo testes de protocolos e componentes em ambientes simulados, tornando fundamental a garantia que as redes sem fio sejam confiáveis e eficazes em diversas aplicações. Com a construção, foram realizados testes, utilizando o modelo simples de propagação por espaço livre, para comprovar o funcionamento da bancada de emulação. Concluiu-se que a bancada, na atual configuração, é capaz de emular um canal de comunicação de RF cujas antenas estejam espaçadas de até 12 metros.

Palavras-chave: Bancada de emulação, Redes de Sensores sem Fio, ESP32.

Abstract

The speed with which technologies have advanced in recent decades directly affects the way we live, work and communicate. However, these technologies faced a number of complex challenges before they were innovative at scale. Uncertainties about its operation, the need to meet imposed requirements and the concern with safety are just some of the obstacles that must be overcome. Thus, this project aims to develop a radio channel emulation environment that allows considering the disturbances commonly suffered by radio frequency signals, without the need to deal with the difficulties arising from the implementation and control of networks in a real environment. To carry out the construction, there is a need for studies focused on *Wireless Sensor Network* (WSN), communication protocols and RF signal propagation models. The bench plays an essential role in the advancement of WSN, allowing testing of protocols and components in simulated environments, making it essential to ensure that wireless networks are reliable and effective in various applications. With the construction, tests were carried out, using the simple free space propagation model, to prove the operation of the emulation bench. It was concluded that the bench, in its current configuration, is capable of emulating an RF communication channel whose antennas are spaced up to 12 meters apart.

Keywords: Emulation bench, Wireless Sensor Networks, ESP32.

Lista de ilustrações

Figura 1 – Componentes básicos de uma RSSF.	4
Figura 2 – Componentes de um NS.	5
Figura 3 – Modelo de propagação no espaço livre.	7
Figura 4 – Comparação das camadas dos protocolos OSI e IEEE 802.11.	10
Figura 5 – Placa <i>Arduino Uno</i>	11
Figura 6 – Atenuador de RF HMC472.	11
Figura 7 – Tabela verdade do atenuador.	12
Figura 8 – Atenuação normalizada.	12
Figura 9 – Diagrama de blocos do módulo ESP32.	13
Figura 10 – <i>Layout</i> dos periféricos do ESP32.	14
Figura 11 – Módulo ESP32 com antena interna.	14
Figura 12 – Módulo ESP32 com antena externa.	15
Figura 13 – Modo Estação.	16
Figura 14 – Modo Ponto de Acesso.	16
Figura 15 – Diagrama das interligações dos componentes da bancada.	19
Figura 16 – Conexão entre o atenuador e o módulo ESP32 (Estação).	20
Figura 17 – Conexão entre Arduino e atenuador.	21
Figura 18 – Bancada de emulação construída.	22
Figura 19 – Funcionamento de todos os componentes da bancada.	23
Figura 20 – RSSI com a atenuação mínima de 0 dB na bancada em funcionamento.	23
Figura 21 – RSSI com a atenuação máxima de 31,5 dB na bancada em funcionamento.	24
Figura 22 – Características do Rádio Wi-Fi do ESP32.	25
Figura 23 – Potências obtidas utilizando o modelo de propagação no espaço livre com distância de 1 a 80 metros.	26
Figura 24 – Potências obtidas utilizando o modelo de propagação no espaço livre com distância de 1 a 80 metros.	27

Lista de tabelas

Tabela 1 – Parâmetros utilizados para o cálculo da potência de recepção do modelo (2.2.1) de propagação no espaço livre.	25
--	----

Lista de siglas

ANATEL	Agência Nacional de Telecomunicações
AP	Ponto de Acesso (AP, do inglês <i>Access Point</i>)
ERB	Estação Rádio Base
IDE	Ambiente de Desenvolvimento Integrado (IDE, do inglês <i>Integrated Development Environment</i>)
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE, do inglês <i>Institute of Electrical and Electronics Engineers</i>)
LAN	Rede de Área Local (LAN, do inglês <i>Local Area Networks</i>)
NS	Nó Sensor
OSI	Interconexão de Sistemas Abertos (OSI, do inglês <i>Open Systems Interconnection</i>)
RF	Radiofrequência
RSSF	Rede de Sensores Sem Fio
RSSI	Indicador da Intensidade do Sinal Recebido (RSSI, do inglês <i>Received Signal Strength Indication</i>)
T_x	Transmissor
R_x	Receptor
Wi-Fi	<i>Wireless Fidelity</i>
WLAN	Redes Locais Sem Fio (WLAN, do inglês <i>Wireless Local Area Networks</i>)
WPAN	Redes Pessoais Sem Fio (WPAN, do inglês <i>Wireless Personal Area Networks</i>)
WSN	<i>Wireless Sensor Network</i>

Lista de símbolos

P_t	Potência transmitida (dBm);
G_t	Ganho da antena transmissora (dBi);
G_r	Ganho da antena receptora (dBi);
λ	Comprimento de onda;
A_e	Abertura efetiva de uma antena;
c	Velocidade da luz em metros por segundos (m/s);
f	Frequência da portadora em Hertz (Hz);
ω_c	Frequência da portadora em radianos por segundo (rad/s);

Sumário

1	INTRODUÇÃO	1
1.1	Justificativa	2
1.2	Objetivos	2
1.3	Estrutura do Trabalho	3
2	REVISÃO TEÓRICA	4
2.1	Redes de Sensores Sem Fio (RSSF)	4
2.1.1	Limitações e interferências das RSSF	6
2.2	Modelos de Propagação	7
2.2.1	Modelo de propagação no espaço livre	7
2.3	Protocolo de comunicação	8
2.3.1	IEEE 802.11	9
2.4	Hardware	10
2.4.1	Critérios técnicos	10
2.4.2	Arduino UNO	10
2.4.3	Atenuador de RF HMC472	11
2.4.4	Módulo ESP32	12
2.4.4.1	Periféricos	13
2.4.4.2	Antena	13
2.4.5	Bibliotecas	14
2.4.5.1	Estação	15
2.4.5.2	Ponto de Acesso	15
2.4.6	Indicador da Intensidade de Sinal Recebido	16
2.4.7	Justificativa dos componentes	17
2.5	Considerações finais	17
3	DESENVOLVIMENTO DA BANCADA DE EMULAÇÃO	19
3.1	Projeto da bancada	19
3.1.1	Construção da bancada	21
3.2	Ensaios de calibração	22
3.3	Considerações finais	23
4	RESULTADOS	25
4.1	Validação dos dados por Espaço Livre	25
4.2	Considerações finais	27

5	CONCLUSÃO	28
5.1	Trabalhos Futuros	28
	REFERÊNCIAS	30
A	ANEXOS	32
A.1	Código para o Controle da Atenuação	32
A.2	Código para o AP	34
A.3	Código para a Estação	36
A.4	Código para coletar os sinais e plotar os histogramas	39
A.5	Código para o cálculo do modelo de propagação espaço livre . .	41

1 Introdução

A comunicação é uma peça primordial para o desenvolvimento da nossa sociedade, com a transmissão de informações e conhecimentos. Desde os primórdios, os meios de comunicação vêm evoluindo em paralelo com o avanço da sociedade e suas tecnologias, de tal maneira que o ser humano aprimora e cria novas formas de se conectar e comunicar, apesar da distância, visto que se torna, cada vez mais, uma necessidade presente na sociedade.

Com o crescente desenvolvimento tecnológico, a maneira que interagimos com um mundo ao nosso redor também muda. Em um mundo onde dispositivos, sistemas e pessoas estão cada vez mais conectados, a comunicação eficiente é o principal ponto que sustenta essa interligação. Por meio desse desenvolvimento, tem-se a necessidade de construir dispositivos para atuação em residências e indústrias para a substituição do ser humano, uma vez que garantem mais confiabilidade, que não possui um alto custo e que conseguem realizar tarefas que possam ser perigosas.

Uma alternativa para a construção de dispositivos de baixo custo é a utilização de rede sem fio, que apresentam significativas vantagens, incluindo maior flexibilidade, baixo custo e facilidade de instalação e manutenção (GUNGOR; HANCKE, 2009). Rede de Sensores Sem Fio (RSSF) são amplamente utilizadas nas aplicações que requer um monitoramento e controle do ambiente, e podem ser compostas por vários sensores, que comunicam entre si para realizar a transmissão de informações até um ponto central. Esses sensores podem ser alocados em locais remotos e de difícil acesso, tornando-os uma ferramenta bastante eficiente.

As redes de sensores têm o potencial de serem empregadas em áreas como controle (para prover algum mecanismo de controle), ambiente (para monitorar variáveis ambientais em locais internos e externos), tráfego (para monitorar tráfegos de veículos em rodovias, malhas viárias urbanas, etc.), segurança (para prover segurança em centros comerciais, estacionamentos, etc.), medicina/biologia (monitorar o funcionamento de órgãos) e militar (detectar movimentos inimigos, explosões, presença de material perigoso como gás venenoso ou radiação, etc.) (LOUREIRO et al., 2003).

RSSF diferem de redes de computadores tradicionais em vários aspectos. Normalmente essas redes possuem inúmeros nós distribuídos, têm restrições de energia, e devem possuir mecanismos para autoconfiguração e adaptação devido a problemas como falhas de comunicação e perda de nós (LOUREIRO et al., 2003). Com isso, são necessárias algumas técnicas de gerenciamento de energia e processamento de dados para poder prolongar a vida útil dos sensores, garantindo assim uma maior confiabilidade dos resultados obtidos.

Os profissionais que irão criar essas redes necessitam de treinamento ou de um ambiente de ensaios para a realização de projetos, para atender, principalmente, peque-

nas e médias empresas e pequenas cidades, uma vez que os empreendimentos de grande porte terão condições de adotar soluções comerciais fechadas, oriundas de fornecedores de tecnologia tradicionais do mercado (DELFORNO, 2020).

Portanto, este trabalho visa desenvolver uma Bancada de Emulação de Rede de Comunicação de baixo custo, tornando-a acessível a um público mais amplo e permitindo a realização de testes precisos e repetitivos em condições de rede simuladas, evitando a necessidade de testes em campo, que podem ter custos elevados e maior tempo de duração.

1.1 Justificativa

Com o avanço das tecnologias, a RSSF já pode ser considerada uma realidade e está presente em ações úteis que visam a melhoria de controle e processos, tanto em residências quanto no meio industrial. A compreensão de suas vantagens e desvantagens é de extrema importância para projeto, instalação e implementação de redes sem fio.

À medida que novas tecnologias de comunicação e arquiteturas de rede são desenvolvidas, é crucial avaliá-las em um ambiente controlado antes de aplicá-las em uma infraestrutura de produção. Uma bancada de emulação de redes permite criar ambientes de teste controlados e reprodutíveis, onde é possível simular diferentes condições de rede.

Avaliar novas tecnologias em uma bancada de emulação pode gerar economia de tempo e dinheiro, uma vez que possibilita identificar possíveis problemas ou limitações antes de investir em uma implantação em grande escala.

Em Delforno (2020), é apresentada o desenvolvimento de uma bancada de emulação de canal de RF de baixo custo, apresentando uma estratégia de emulação das condições de comunicação sem fio, considerando as perturbações sofridas pelo sinal de radiofrequência, impondo ao sinal de rádio as flutuações de intensidade do sinal com base em modelos de propagação. O protocolo de comunicação utilizado na bancada foi o Radiuino.

Neste trabalho, pretende-se construir uma bancada com uma abordagem diferente do que a proposta por Delforno (2020) de baixo custo para a emulação de redes de comunicação sem fio. A bancada será configurada com um módulo transmissor e um módulo receptor de sinal e deverá conseguir reproduzir diferentes condições de uma rede sem fio, facilitando a análise e teste de diferentes configurações de RSSF e permitindo comparações entre resultados medidos e calculados.

1.2 Objetivos

O objetivo deste trabalho é realizar a montagem de uma bancada de emulação de baixo custo, que possa ser utilizada para implementações de projetos em instituições de ensino e pesquisa, ou, para ensaios em empresas de projeto, tudo feito de maneira remota.

Os objetivos específicos são:

- Estudo de protocolos de comunicação associados as redes sem fio;
- Levantamento e estudo *hardwares* a serem utilizados na construção da bancada;
- Realizar testes de atenuação de redes de comunicação conforme modelos de propagação;
- Controlar a atenuação e com isso, realizar a comparação entre os resultados teóricos baseados nos modelos de propagação e os resultados obtidos via bancada.

1.3 Estrutura do Trabalho

No Capítulo 1, é apresentada uma introdução sobre comunicação e RSSF, justificativa e os objetivos do presente trabalho.

O Capítulo 2 apresenta uma revisão teórica dos estudos e conceitos sobre a RSSF, modelo de propagação, protocolo de comunicação utilizados em redes de sensores e os *hardwares* utilizados para o desenvolvimento do trabalho.

O Capítulo 3 apresenta o projeto da elaboração do desenvolvimento da bancada de emulação fisicamente e os ensaios de calibração e validação.

O Capítulo 4 apresenta os resultados obtidos com os testes e validações da bancada em funcionamento em comparação com o modelo de propagação estudado.

E, por fim, o Capítulo 5 apresenta as conclusões e a indicação das atividades seguintes a serem desenvolvidas como trabalhos futuros.

2 Revisão Teórica

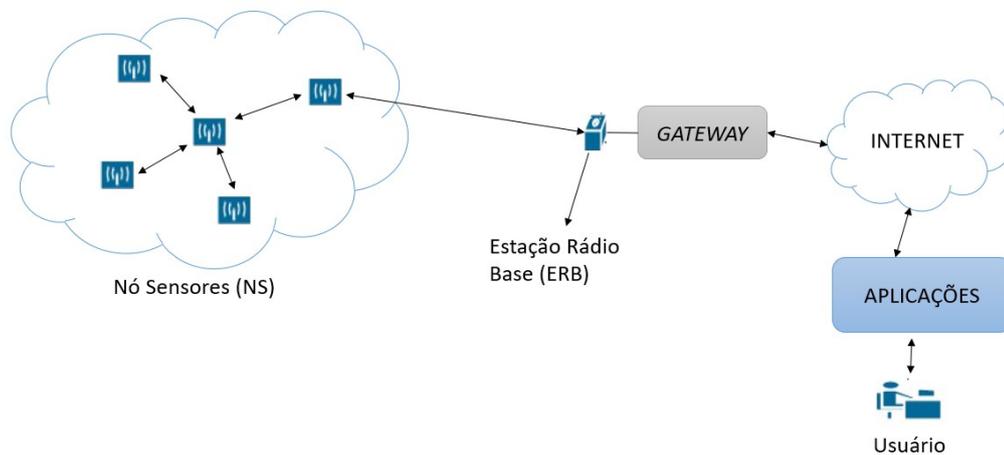
Neste capítulo, será feita uma revisão teórica sobre rede de sensores sem fio (estrutura, limitações e interferências), modelo de propagação, protocolo de comunicação e, por fim, os *hardwares* que serão utilizados para o desenvolvimento deste trabalho.

2.1 Redes de Sensores Sem Fio (RSSF)

Uma RSSF é composta por um conjunto de dispositivos sensores que coletam informações do ambiente físico, como temperatura, pressão, potência de sinais, e os transmitem para um nó central ou uma estação base. Como mostrados na Figura 1, tradicionalmente as RSSF são compostas por Nó Sensor (NS), Estação Rádio Base (ERB) e *Gateway*.

Os dados obtidos e processados pelo NS são encaminhados até uma estação central de agrupamento e tratamento dos dados. A estação de controle deve ter a capacidade de manipular os dados encaminhados pelos nós sensores e executar tarefas de gerenciamento e controle (CARVALHO et al., 2012).

Figura 1 – Componentes básicos de uma RSSF.

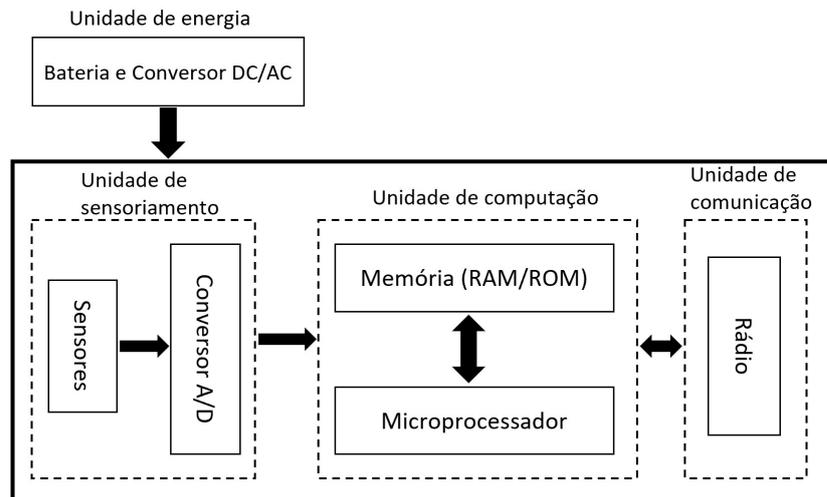


Fonte: Adaptado de Freitas e Branquinho (2017).

O NS são dispositivos capazes de monitorar grandezas, como temperatura, luminosidade ou umidade, e estão aptos para realizar processamento local e a comunicação desses dados. Os nós sensores também podem servir exclusivamente como nós roteadores das informações produzidas por outros nós comuns da rede. Os dados coletados são enviados para a ERB por meio de uma comunicação sem fio.

Um NS é composto, basicamente, por um elemento de computação baseado em um microcontrolador e memória, sensores e atuadores, fonte de energia e dispositivos de comunicação (FREITAS, 2018). A Figura 2 apresenta os principais componentes de um NS.

Figura 2 – Componentes de um NS.



Fonte: Adaptado de Freitas (2018).

- Unidade de energia: a unidade de energia é importante para manter o funcionamento de todos os componentes no NS. Como normalmente não há fonte de alimentação conectada, é necessário algum tipo de bateria para fornecer energia. às vezes, alguma forma de recarga através da obtenção de energia do meio ambiente também está disponível (KARL; WILLIG, 2007).
- Unidade de sensoriamento: são compostos pelos sensores e atuadores. Este componente representa a interface com o mundo físico responsável por observar e controlar parâmetros físicos do ambiente monitorado (SAUSEN et al., 2008).
- Unidade de computação: consiste em um microprocessador e memórias. Serão executadas informações sobre o monitoramento e o armazenamento dos dados e códigos a serem executados.
- Unidade de comunicação: é composto por um rádio que oferece uma comunicação sem fio entre o NS e ERB.

Segundo Ilyas e Mahgoub (2014), a ERB consiste em um componente responsável por receber, armazenar e processar os dados de vários NS. A ERB é ligada a um computador e comunica-se com os NS sem a utilização de cabos. Portanto, a função da ERB é fazer a conexão entre a RSSF a uma rede cabeada.

O nome geral para uma máquina que faz uma conexão entre duas ou mais redes e oferece a conversão necessária, tanto em termos de *hardware* quanto de *software*, é um *Gateway* (em português, porta de entrada) (TANENBAUM; WETHERALL, 2011). Funciona como um portal, ou seja, age como uma espécie de “fio condutor” da conexão da RSSF com a Internet. O propósito é realizar uma passagem do meio de comunicação sem fio para um meio cabeado e garantir que os dados possam ser transmitidos e recebidos de maneira eficiente e segura.

2.1.1 Limitações e interferências das RSSF

As RSSF possuem algumas limitações que devem ser consideradas no ato de sua implementação e gerenciamento. Segundo Bento et al. (2009), os sensores sem fios são limitados em termos de capacidade de processamento e também de armazenamento. Com isso, tende a dificultar o desenvolvimento de medidas de segurança. As redes de sensores sofrem das restrições das redes móveis *ad-hoc*¹, tais como a falta de confiabilidade do canal de comunicação, falta de uma infraestrutura e problemas de colisão ou danificação de pacotes.

A interferência consiste na alteração de alguma das características do sinal transmitido por efeito de outro sinal exterior, ou interior, ao sistema de transmissão. Interferências internas podem ocorrer quando os nós pertencentes à RSSF transmitem ao mesmo tempo. Por outro lado, a externa pode ser causada por outras redes ou dispositivos operando na mesma banda de frequência que a RSSF.

Devido ao aumento na quantidade de aplicações que utilizam tecnologias de comunicação sem fio, o espectro disponível para comunicação tende a ficar mais poluído, aumentando a interferência e reduzindo a qualidade de serviço das redes que coexistem em um mesmo ambiente (GOMES et al., 2014). Esse tipo de interferência torna-se cada vez mais comum, à medida que aumenta a quantidade de dispositivos que aproveitam essa nova tecnologia para reduzir o número de conexões utilizando cabos (CAVALCANTI, 2009).

No Brasil, há pelo menos três diferentes faixas de frequências que podem ser usadas sem a necessidade de obtenção de licença pela Agência Nacional de Telecomunicações (ANATEL), 928 MHz, 2,4 GHz e 5 GHz. A faixa de frequência de 2,4 GHz, por ela ser utilizada por vários equipamentos e serviços, como aparelhos de telefone sem fio e *Bluetooth*, podem levar a interferências e congestionamentos (RUFINO, 2019).

¹ As redes *ad hoc* são redes sem fio formadas de maneira espontânea entre os dispositivos próximos uns dos outros, sem a necessidade de uma infraestrutura de rede existente.

2.2 Modelos de Propagação

A rede de comunicação móvel impõe limitações fundamentais para o desempenho dos sistemas de comunicação sem fio. O caminho que o sinal pode tomar, entre o transmissor e o receptor, pode variar desde uma simples linha de visão até um que seja seriamente obstruído por prédios, montanhas, etc.

O estudo da propagação é importante para a compreensão das comunicações sem fio, pois fornece a modelagem física necessária que leva a uma boa estimativa da potência requerida para o estabelecimento dos enlaces de comunicações para fornecer uma comunicação confiável (HAYKIN; MOHER, 2009).

Quando o nível de um sinal não pode ser medido com equipamentos próprios em um receptor, ele pode ser estimado por meio de modelos matemáticos determinísticos e/ou empíricos que tentam, por aproximação, simular o comportamento das ondas eletromagnéticas durante a propagação em um determinado meio. A seguir, será apresentado um modelo de propagação usado para prever a potência recebida nos sistemas de comunicação.

2.2.1 Modelo de propagação no espaço livre

O modelo de propagação no espaço livre é o mais básico, por ser usado para prever a intensidade do sinal recebido quando Transmissor (T_x) e Receptor (R_x) possuem um caminho de linha de visão limpo, sem nenhum tipo de obstrução, entre eles, conforme Figura 3.

Figura 3 – Modelo de propagação no espaço livre.



Fonte: Do autor.

O modelo prevê que a potência recebida diminui com uma função da distância de separação T_x-R_x elevada a alguma potência. A potência no espaço livre recebida por uma antena receptora separada de uma antena transmissora, irradiando, por uma distância d é dada pela equação do espaço livre de Friis (RAPPAPORT, 2009), que considera uma antena isotrópica ² na transmissão e a potência P_t ,

² É uma antena ideal que irradia potência com ganho unitário uniformemente em todas as direções, e é normalmente usado para referenciar ganhos de antena em sistema sem fio.

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2} \quad (2.1)$$

em que P_t é a potência transmitida, $P_r(d)$ é a potência recebida, sendo uma função da separação $T_x—R_x$, G_t e G_r são os ganhos da antena transmissora e receptora, respectivamente, d é a distância de separação $T_x—R_x$ em metros e λ é o comprimento de onda em metros. O ganho máximo de transmissão ou de recepção de uma antena, em qualquer direção (HAYKIN; MOHER, 2009), está relacionado à sua abertura efetiva, A_e , por

$$G = \frac{4\pi}{\lambda^2} A_e.$$

A abertura efetiva A_e é uma medida que está relacionada ao tamanho físico da antena, descreve a capacidade de uma antena de receber ou transmitir sinais eletromagnéticos em uma direção específica e λ está relacionado à frequência da portadora por meio de

$$\lambda = \frac{c}{f} = \frac{2\pi c}{\omega_c}.$$

A perda no espaço livre L_p , entre duas antenas, é expressa por

$$L_p = \left(\frac{4\pi d}{\lambda} \right)^2. \quad (2.2)$$

Em termos logarítmicos, (2.2) pode ser reescrita como em

$$L_p = 10 \log_{10} \left(\frac{4\pi d}{\lambda} \right)^2.$$

Essa definição de perda depende, de um modo surpreendente, do comprimento de onda irradiada, consequência da dependência da área efetiva de uma antena isotrópica com o comprimento de onda (HAYKIN; MOHER, 2009).

Reescrevendo (2.1) em função de (2.2), tem-se

$$P_r(d) = \frac{P_t G_t G_r}{L_p} \quad (2.3)$$

Apesar de ser considerado um modelo simples, é um primeiro modelo para poder comprovar como é o comportamento da propagação da potência em um espaço sem obstruções.

2.3 Protocolo de comunicação

Protocolo de comunicação são regras que definem como os dispositivos devem comunicar entre si em uma rede de computadores. São essenciais para garantir que a

informação seja fornecida de forma confiável através da rede. Os protocolos devem ser eficientes em termos de energia, uma vez que os sensores geralmente têm recursos limitados de bateria. A seguir, será apresentado o protocolo de comunicação que será utilizado para o desenvolvimento do trabalho.

2.3.1 IEEE 802.11

O IEEE 802.11 é um padrão de rede sem fio para Redes Locais Sem Fio (WLAN, do inglês *Wireless Local Area Networks*) desenvolvido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE, do inglês *Institute of Electrical and Electronics Engineers*). Esse padrão define uma série de especificações que basicamente define como deve ser a comunicação entre um dispositivo cliente e um concentrador ou a comunicação entre dois dispositivos clientes. O padrão 802.11 original (também conhecido como *Wireless Fidelity* (Wi-Fi)), em termos de velocidade de transmissão, provê, no máximo, 2 Mbps, trabalhando com a banda de 2,4 GHz (RUFINO, 2019). Hoje, o Wi-Fi também pode ser transmitido pela frequência de 5 GHz.

Pela sua versatilidade, o IEEE 802.11 permite que vários dispositivos como *smartphones*, computadores e outros se conectem à internet e se comuniquem sem a necessidade de uma conexão com fio.

Contudo, o IEEE 802.11 apresenta alguns desafios como a necessidade de gerenciamento de interferência e congestionamento nas frequências densamente povoadas, definição de vários mecanismos de segurança para a proteção das RSSF de ameaças externas.

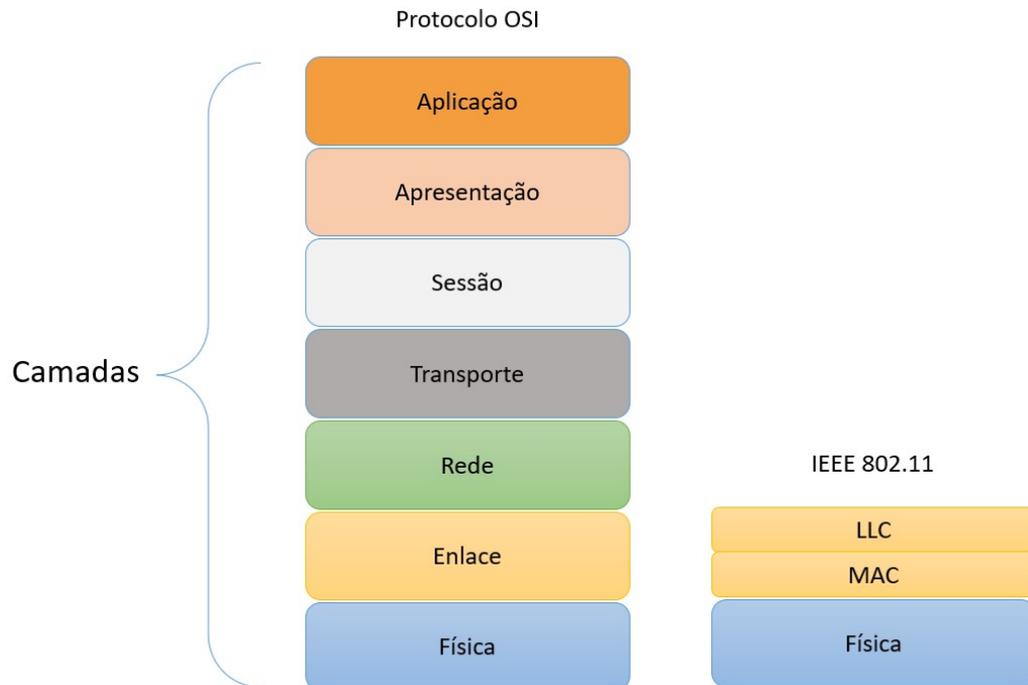
As camadas são geralmente organizadas de maneira semelhante à Interconexão de Sistemas Abertos (OSI, do inglês *Open Systems Interconnection*)³, mas o IEEE 802.11 tem apenas três camadas:

- Camada Física: Responsável pela transmissão e recepção sinais de rádio através do meio físico, incluindo modulação, frequência de operação e codificação de dados.
- Camada de Controle de Acesso ao Meio (MAC): Responsável por controlar o acesso ao meio físico de transmissão e gerenciar alocação de recursos de rede. Define as regras para a transmissão e recepção dos dados, incluindo o protocolo de acesso múltiplo com detecção de portadora e evitação de colisão de dados (CSMA/CA).
- Camada de Controle do Enlace Lógico (LLC): Responsável por especificar os mecanismos para endereçamento, fornece uma interface padronizada entre a camada de rede e a MAC. Tem a importante de garantir que as informações sejam transmitidas com eficiência e segurança. Fornece mecanismos para controle de erro, detecção e correção de dados e gerenciamento de fluxo de tráfego.

³ O modelo OSI é composto por 7 camadas em sua estrutura sendo: Física, Enlace, Rede, Transporte, Sessão, Apresentação e Aplicação

A Figura 4 apresenta uma comparação das camadas dos protocolos OSI e IEEE 802.11.

Figura 4 – Comparação das camadas dos protocolos OSI e IEEE 802.11.



Fonte: Do autor.

2.4 Hardware

Nesta seção, serão apresentados os *hardwares* que serão utilizados para o desenvolvimento do trabalho, critérios técnicos e a justificativa pela escolha deles.

2.4.1 Critérios técnicos

Para a construção da bancada, deve-se considerar alguns critérios técnicos, como o *hardware* da bancada ser poderoso o suficiente para emular as redes de comunicação desejada, que possua uma elevada confiabilidade, baixo consumo de energia e um custo reduzido. A escolha de um *software* adequado que possibilite a configuração e a simulação de redes e estabelecimento de conexões físicas e virtuais entre os dispositivos para permitir testes realistas.

2.4.2 Arduino UNO

O *Arduino UNO*, apresentado na Figura 5, é um *hardware* que contém uma placa microcontroladora baseada no ATmega328P. Possui 14 pinos de entrada/saída digital (dos

quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um ressonador cerâmico de 16 MHz (ARDUINO, 2022), uma linguagem de programação padrão, a qual tem origem em *Wiring*, e é essencialmente C/C++. Uma das vantagens do microcontrolador é possuir alto desempenho, baixo consumo de energia e uma memória EEPROM que não é apagada quando desligado.

Figura 5 – Placa *Arduino Uno*.

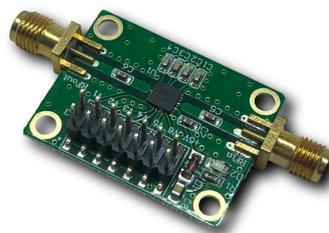


Fonte: Arduino (2022).

2.4.3 Atenuador de RF HMC472

HMC472, Figura 6, é um atenuador digital de banda larga de 6 bits com uma faixa de atenuação de 0,5 dB a 31,5 dB e um erro típico de $\pm 0,25$ dB. A sua faixa de frequência operacional é 1 MHz a 3,8 GHz. A potência máxima do sinal de entrada é de 20 dBm⁴.

Figura 6 – Atenuador de RF HMC472.



Fonte: Do autor.

O controle de atenuação é realizado através dos seus 6 pinos (V1 a V6) na placa. Possui uma alimentação de 5 V (DEVICES, 2012). Para obter esses níveis de atenuação, o *datasheet* apresenta uma tabela da verdade informando os níveis que cada um dos 6 pinos do atenuador deva ter, como apresentado na Figura 7, para atingir a atenuação desejada do sinal.

⁴ Unidade de medida para expressar a potência de um sinal de energia em decibéis (dB) em relação a um miliwatt (mW).

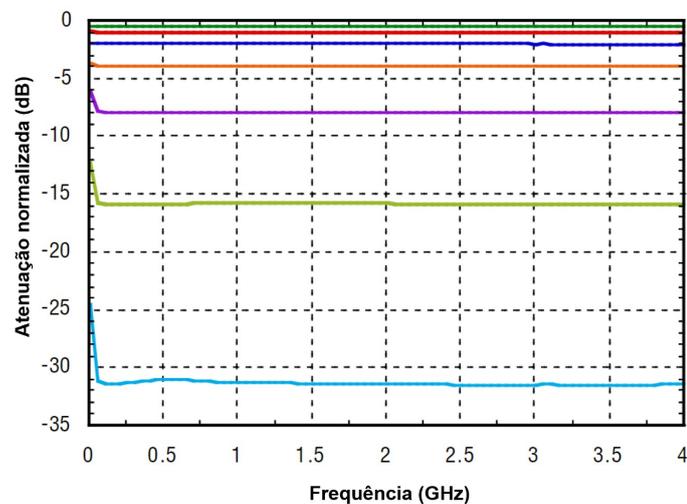
Figura 7 – Tabela verdade do atenuador.

Control Voltage Input						Attenuation State RF1 - RF2
V1 16 dB	V2 8 dB	V3 4 dB	V4 2 dB	V5 1 dB	V6 0.5 dB	
High	High	High	High	High	High	Reference I.L.
High	High	High	High	High	Low	0.5 dB
High	High	High	High	Low	High	1 dB
High	High	High	Low	High	High	2 dB
High	High	Low	High	High	High	4 dB
High	Low	High	High	High	High	8 dB
Low	High	High	High	High	High	16 dB
Low	Low	Low	Low	Low	Low	31.5 dB

Fonte: Devices (2012).

A Figura 8 apresenta a atenuação normalizada no atenuador, mostrando apenas os estados normalizados. Analisando o gráfico, o atenuador tem uma resposta plana perto da banda de 2,4 GHz, frequência utilizada na comunicação de dispositivos Wi-Fi.

Figura 8 – Atenuação normalizada.



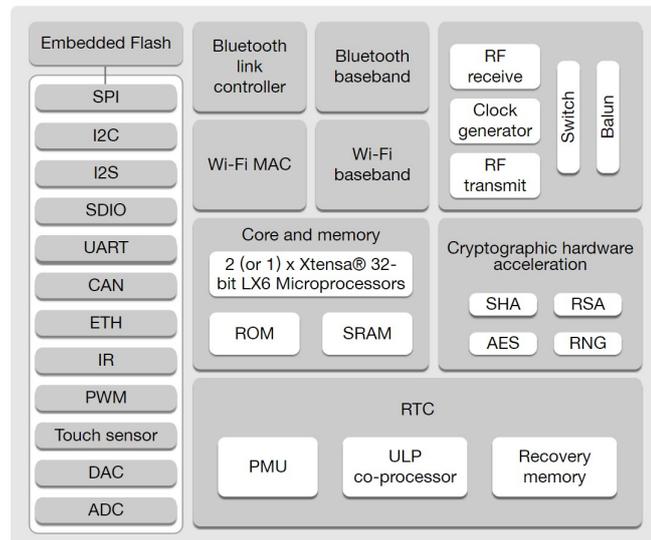
Fonte: Devices (2012).

2.4.4 Módulo ESP32

O módulo ESP32 é um microcontrolador desenvolvido pela *Espressif Systems* e amplamente utilizado em projetos de RSSF devido à sua capacidade de se conectar à internet e interagir com outros dispositivos. O ESP32 possui dois microcontroladores Xtensa 32-bit LX6, suporta uma frequência de *clock* de até 240 MHz, tendo como comum uma frequência de 160 MHz, possui uma memória interna 448 KB de ROM para inicialização e

funções principais, 520 KB de SRAM no chip para dados e instruções (SYSTEMS, 2023b), conectividade Wi-Fi (802.11b/g/n) e *Bluetooth* integradas, facilitando a comunicação sem fio com outros dispositivos. A Figura 9 apresenta o diagrama de blocos do módulo ESP32.

Figura 9 – Diagrama de blocos do módulo ESP32.



Fonte: (SYSTEMS, 2023b).

2.4.4.1 Periféricos

O ESP32 possui uma variedade de periféricos, 38 pinos programáveis, como GPIO (Entrada/Saída de Objetivo Geral), UART (Receptor/Transmissor Assíncrono Universal), SPI (Interface Periférica Serial), I2C (Circuito Inter-Integrado), PWM (Modulação por Largura de Pulso), ADC (Conversor Analógico-Digital), DAC (Conversor Digital-Analógico) (SYSTEMS, 2023a). A Figura 10 apresenta o *layout* dos periféricos do ESP32.

2.4.4.2 Antena

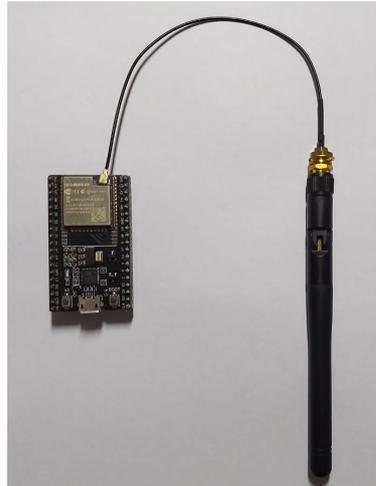
Antena visa a captação e/ou a transmissão de ondas eletromagnéticas. O módulo ESP32 pode ser encontrado na versão com antena interna ou com antena externa. O ganho de uma antena⁵ pode variar dependendo do tipo.

Módulos que possuem antena interna possuem indicação para operações em curtas distâncias. A Figura 11 apresenta o módulo com antena interna. Possuem um ganho baixo, em torno de 2 dBi a 3 dBi.

Módulos com antena externa dispõem de um conector IPEX que permite a conexão de uma antena externa. Possuem indicação para melhorar o alcance e a qualidade da

⁵ Medida da capacidade da antena de direcionar ou concentrar a energia radiada em uma determinada direção.

Figura 12 – Módulo ESP32 com antena externa.



Fonte: Do autor.

do módulo para se conectar a redes Wi-Fi. As configurações podem variar dependendo do dispositivo ou sistema que irá utilizar. Para utilizar a biblioteca “WiFi.h”, é necessário incluir no princípio do código implementado na IDE usando o seguinte comando: `#include <WiFi.h>`.

Dentre essas diversas configurações, a seguir, serão abordadas duas configurações utilizadas para o desenvolvimento do trabalho, Estação e Ponto de Acesso (AP, do inglês *Access Point*).

2.4.5.1 Estação

Uma estação é um dispositivo cliente que se conecta a rede para transmitir, receber ou processar dados. Pode ser representada por um computador, *smartphone*, módulos de comunicação ou qualquer outro tipo de dispositivo capaz de se conectar a uma rede, como apresentado na Figura 13. É caracterizada como um ponto final de comunicação em uma rede. Esses dispositivos se conectam a uma rede por meio de um ponto de acesso e podem trocar dados com outros dispositivos na rede.

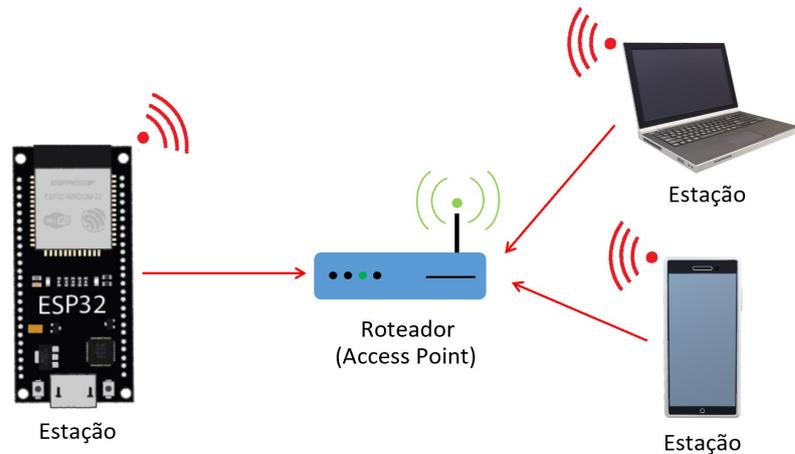
Cada estação possui um endereço de IP único na rede, que a identifica e permite a comunicação com outros dispositivos na rede.

2.4.5.2 Ponto de Acesso

O AP é um dispositivo de rede sem fio que permite a conexão de dispositivos a uma Rede de Área Local (LAN, do inglês *Local Area Networks*). Atua como um ponto central para a transmissão e recepção de dados sem fio entre os dispositivos conectados e a rede, como apresentado na Figura 14. Suporta padrões como o IEEE 802.11.

Esse dispositivo funciona como uma ponte entre os dispositivos sem fio e uma rede com fio, recebendo os sinais de rádio dos dispositivos sem fio, converte em sinais de rede

Figura 13 – Modo Estação.



Fonte: Do autor.

Figura 14 – Modo Ponto de Acesso.



Fonte: Do autor.

com fio e os transmite aos servidores e/ou outros dispositivos na rede. Da mesma forma, realiza o processo ao contrário.

O AP pode ser dividido em dois tipos: Doméstico e Empresarial. O Ponto de Acesso Doméstico é utilizado em residências ou em escritórios de menor porte, fornecendo conectividade Wi-Fi em uma pequena faixa de área. O Ponto de Acesso Empresarial é projetado para ambientes maiores, pode suportar uma maior carga de dispositivos conectados simultaneamente, oferecendo melhor desempenho e estabilidade.

2.4.6 Indicador da Intensidade de Sinal Recebido

O Indicador da Intensidade do Sinal Recebido (RSSI, do inglês *Received Signal Strength Indication*) é uma medida estimada do nível de potência que um dispositivo

cliente de RF está recebendo de um AP ou um roteador. Mede a potência recebida no nó remoto (RSSId - pacote de *download*) e na base (RSSIu - pacote de *upload*). O RSSI é medido em dBm e é a informação de mais baixo nível sobre a qualidade do link, podendo ser considerado um parâmetro de camada física. O decibel (dB) é a diferença (ganho ou perda) entre dois níveis de potência, em escala logarítmica. Portanto, se a diferença for 0 dB, os dois níveis de potência são iguais. A intensidade do sinal muitas vezes segue uma escala logarítmica, onde pequenas variações em dB representam grandes diferenças em termos de potência real. Normalmente, é usado valores negativos para a medição da intensidade para poder indicar uma queda na potência do sinal com relação a um ponto de referência. Então quanto maior for o valor (mais próximo de zero), melhor será a qualidade do sinal.

2.4.7 Justificativa dos componentes

A escolha dos componentes que serão apresentados a seguir levou-se em consideração a alguns fatores. O *Arduino Uno*, por se tratar de uma plataforma *Open Source*, não há necessidade de aquisição do *software* para programação — o *software* Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*) é disponibilizado gratuitamente, fazendo com que o custo do projeto não seja tão elevado. O ESP32 por ser um microcontrolador de baixo custo e alto desempenho, possui suporte para comunicação Wi-Fi e *Bluetooth*. O atenuador de sinal HMC472 possui um baixo custo e possui uma resposta na frequência utilizada em comunicação de dispositivos Wi-Fi. Além disso, a utilização do atenuador permite simular uma atenuação como ocorre em redes reais.

2.5 Considerações finais

Neste capítulo, foram apresentados os estudos dos componentes necessários para o desenvolvimento de uma bancada de emulação de redes de comunicação.

A RSSF é composta por um conjunto de sensores distribuídos para a coleta de informações, monitoramento ou execução de tarefas. O modelo de propagação descreve como os sinais de RF das RSSF se propagam no ambiente. O modelo de propagação no espaço livre prevê a intensidade do sinal entre o transmissor e o receptor quando possuem um caminho sem obstruções. Apesar de ser considerado um modelo simples, é um primeiro modelo capaz de representar o comportamento da intensidade de potência recebida pela antena. Para aprimoramento dessas comprovações e como trabalhos futuros, deve-se utilizar protocolos mais sofisticados. Os protocolos de comunicação utilizados na RSSF definem como os dispositivos devem comunicar entre si. O protocolo IEEE 802.11 (Wi-Fi) permite realizar a comunicação sem fio dos módulos ESP32. E, por fim, a seleção

dos *hardwares* que serão utilizados para a construção da bancada, levando em consideração alguns fatores como baixo custo e baixo consumo de energia.

Diante desses estudos, será possível desenvolver a bancada e realizar testes para comprovar o seu funcionamento.

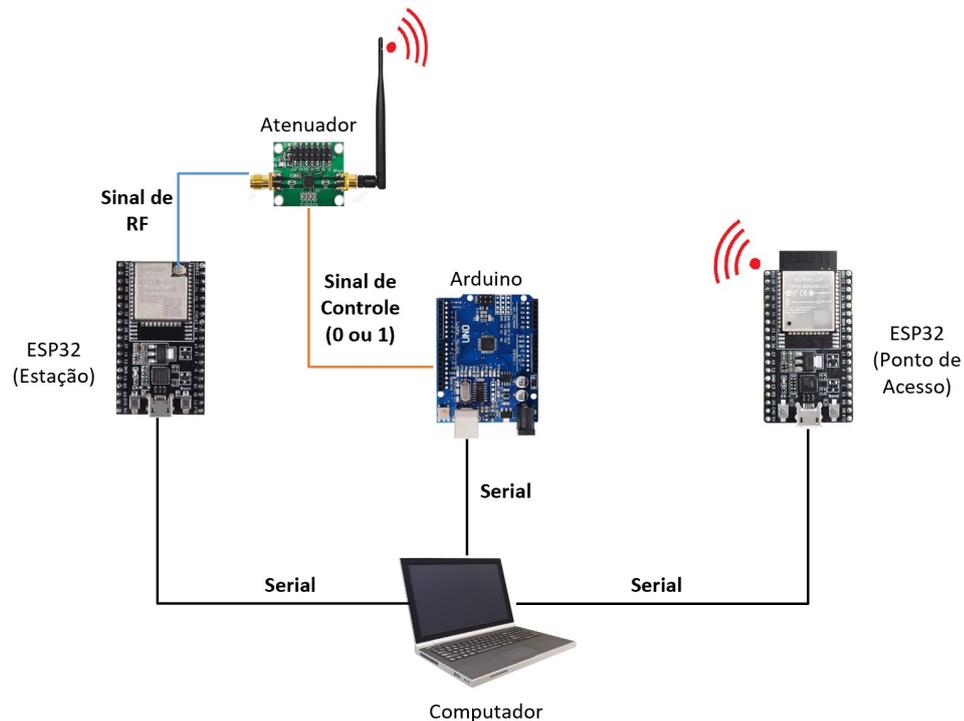
3 Desenvolvimento da Bancada de Emulação

A bancada de emulação, construída neste trabalho, visa desenvolver um ambiente de emulação de canal de rádio com o intuito de avaliar e realizar teste de comunicação utilizando uma plataforma *open source* (*hardware, firmware e software*) para o desenvolvimento das RSSFs, observando as perturbações pela transmissão do sinal de RF. A princípio, objetiva-se a reprodução fiel do protótipo desenvolvido em Delforno (2020). Devido a alguns empecilhos encontrados, uma nova versão da bancada foi desenvolvida. A seguir, serão apresentadas as etapas para o desenvolvimento da bancada.

3.1 Projeto da bancada

Para a construção da bancada, será necessário um Arduino Uno (controlador), um atenuador digital e dois módulos ESP32. Para o começo do desenvolvimento, foram consideradas as conexões entre os componentes. A Figura 15 apresenta um diagrama das interligações dos componentes.

Figura 15 – Diagrama das interligações dos componentes da bancada.



Fonte: Do autor.

É necessário configurar os módulos ESP32 para a comunicação entre eles possa ser estabelecida. Na configuração da bancada, serão utilizados dois módulos ESP32, sendo um configurado como um AP e outro como Estação. O módulo configurado como AP apresenta uma antena interna e o módulo configurado como Estação uma antena externa. Quando o ESP32 é definido como ponto de acesso, é possível conectar qualquer dispositivo com recursos Wi-Fi a ele, sem a necessidade de um roteador. Em resumo, cria-se uma própria rede Wi-Fi e os dispositivos podem conectar-se a ele. Por outro lado, na configuração de estação, o ESP32 se conecta à rede criada pelo ponto de acesso.

Com relação à transmissão e recepção dos dados, o módulo Estação será o transmissor e o módulo Ponto de Acesso (AP, do inglês *Access Point*) (AP) será o receptor. O ESP32 Estação será configurado com um *firmware* para enviar pacotes de dados de 52 *bytes* apenas requisitando a qualidade do sinal RSSI de *downlink* e *uplink*. Com isso, será possível verificar se a potência de recepção está coerente com a potência de transmissão.

Um computador é conectado a um módulo ESP32 (configurado no modo Estação) por meio de uma porta serial. Utilizando o *software* MATLAB, será possível avaliar, por meio de gráficos, a qualidade do sinal RSSI de *downlink* e *uplink*.

Em sequência, é conectado ao módulo Estação o atenuador digital por meio de um cabo *pigtail* de 150 mm. A conexão de entrada do atenuador (RFin) se conecta com o módulo ESP32 e a conexão de saída (RFin) é conectada a uma antena externa. A Figura 16 apresenta essa conexão entre o atenuador e o módulo ESP32. A comunicação entre os módulos AP e a antena conectada ao atenuador será sem fio (feita por Wi-Fi).

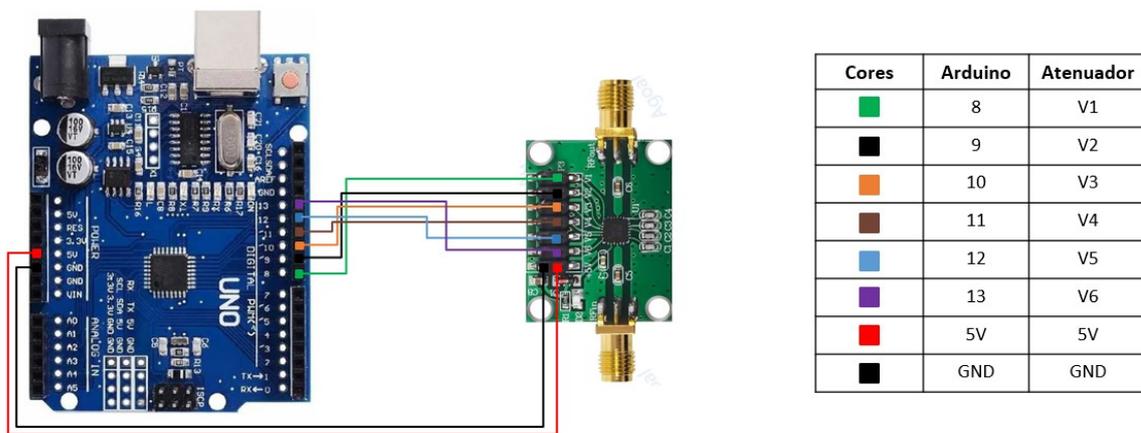
Figura 16 – Conexão entre o atenuador e o módulo ESP32 (Estação).



Fonte: Do autor.

Um microcontrolador (Arduino) é conectado, por meio de cabos de cobre, aos pinos de controle do atenuador, tendo o papel de controlar a atenuação da potência do sinal entre os dois módulos ESP32. A Figura 17 apresenta a conexão física entre o Arduino e o atenuador. Para a configuração da atenuação do sinal no Arduino, foi consultado o *datasheet* do atenuador HMC472 e desenvolvido o algoritmo para a configuração do controle. Para uma variação de 0 dB a 31,5 dB de atenuação, as configurações de tensões nos pinos de saída do Arduino variam de *high* (alto) e *low* (baixo). Para a alimentação do atenuador foi utilizada a saída de 5 V do próprio Arduino.

Figura 17 – Conexão entre Arduino e atenuador.



Fonte: Do autor.

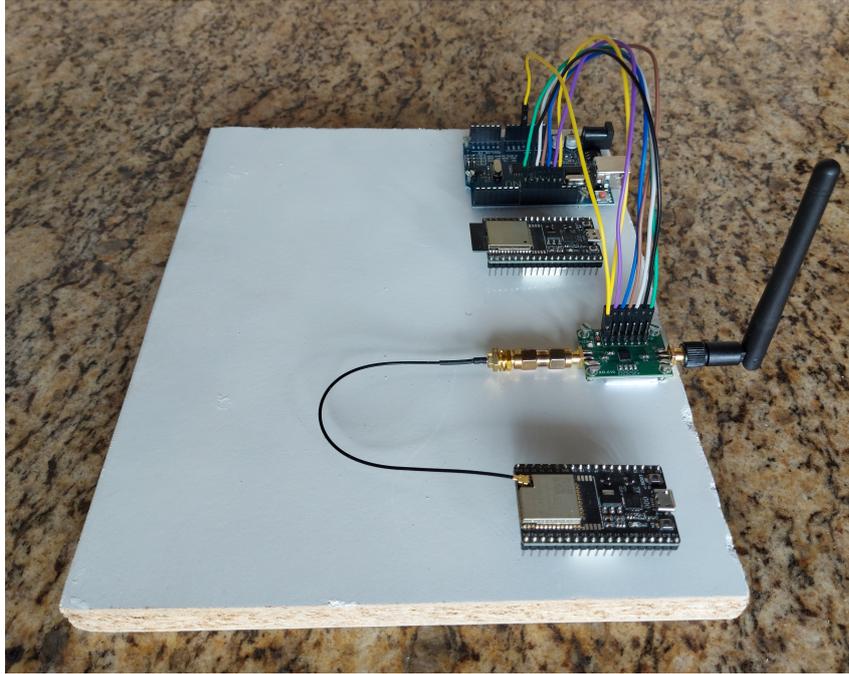
A comunicação serial há entre o Arduino e o ESP32 (AP) com o computador é para apenas funcionamento dos mesmos.

Para realizar a implementação dos códigos de controle de atenuação, ponto de acesso e a estação foi utilizada a própria IDE do Arduino, por se tratar de um *software* gratuito e de fácil manuseio. Para a coleta dos resultados, foi desenvolvido um *script* no MATLAB, onde poderá ser plotados os gráficos com os valores da RSSI de *uplink* e *downlink*. Os algoritmos desenvolvidos encontram-se no Apêndice A.

3.1.1 Construção da bancada

Diante das informações apresentadas, foram fixados os componentes em uma estrutura de base de madeira com dimensões de 25 × 20 cm. A escolha do tamanho da base foi feita com base em projetos futuros para aprimorar a bancada. O resultado da bancada montada é apresentada na Figura 18.

Figura 18 – Bancada de emulação construída.



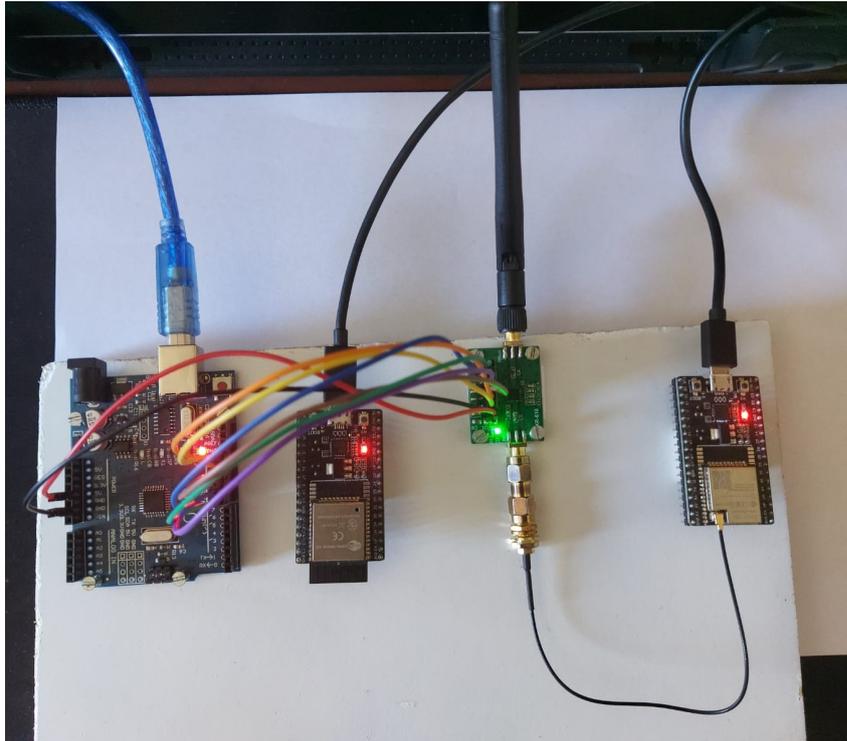
Fonte: Do autor.

3.2 Ensaio de calibração

Após a fixação dos componentes na base, foram realizados alguns ensaios de calibração para verificar o funcionamento da bancada. A Figura 19 apresenta os componentes da bancada em funcionamento. O primeiro ensaio foi de verificar o RSSI entre os módulos transmissor e receptor, ou seja, entre os módulos ESP32 no modo Estação e modo AP, mas com a atenuação no atenuador em 0 dB, como mostrado na Figura 20. O segundo ensaio verificou o RSSI com atenuação máxima de 31,5 dB, como apresentado na Figura 21. Para a obtenção dos valores, o *script* do MATLAB executou uma rotina que captava, ao total, 400 medições de RSSI.

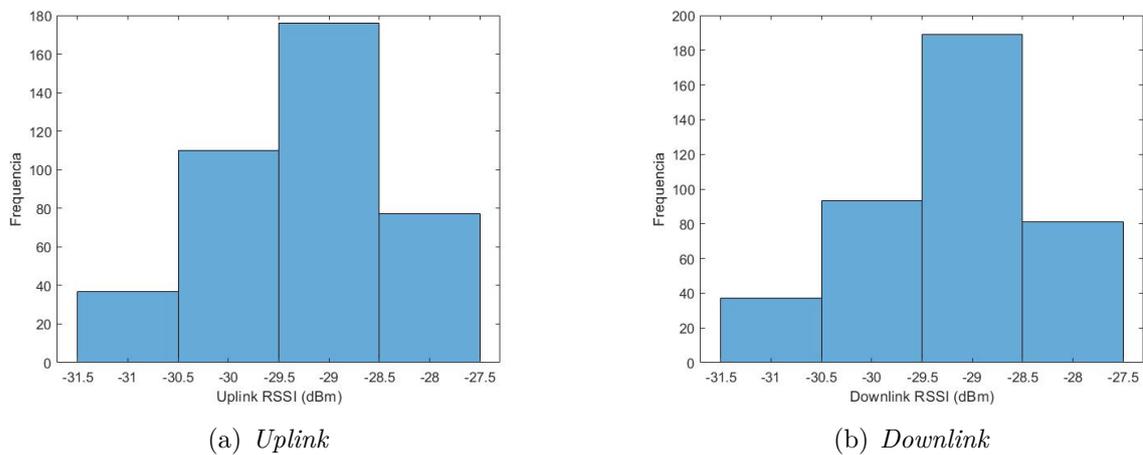
Analisando os resultados obtidos, percebe-se que quando não há nenhuma atenuação, 0 dB, o RSSI fica na faixa de -28 dBm e -31 dBm para *uplink* e *downlink*. A diferença entre os dois gráficos se dá pela quantidade de repetições (eixo y) que tiveram cada nível de potência. Quando há uma atenuação máxima fornecida pelo atenuador de 31,5 dB, tem-se uma faixa de RSSI de -60 dBm e -67 dBm para *uplink* e de -61 dBm e -67 dBm para *downlink*, concentrando as maiores medições entre -64 dBm a -66 dBm.

Figura 19 – Funcionamento de todos os componentes da bancada.



Fonte: Do autor.

Figura 20 – RSSI com a atenuação mínima de 0 dB na bancada em funcionamento.

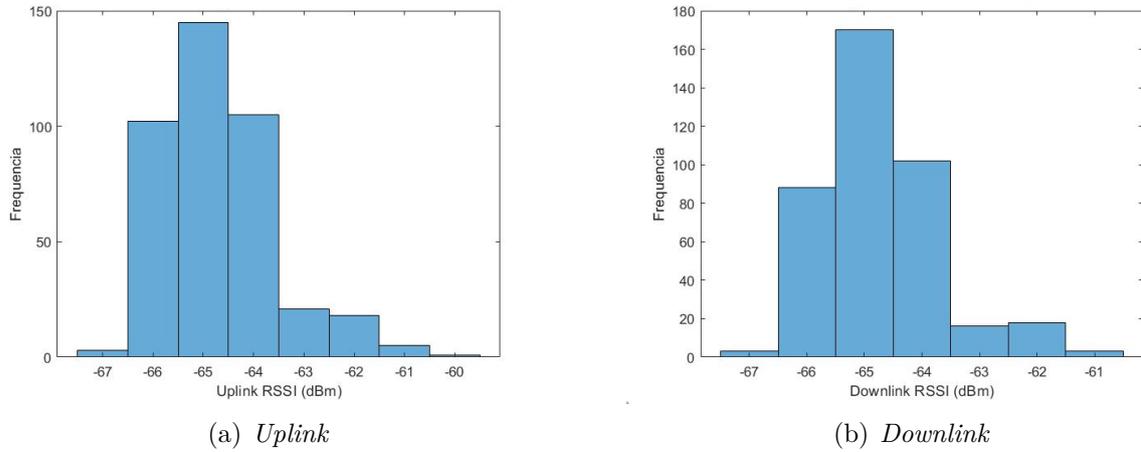


Fonte: Do autor.

3.3 Considerações finais

Neste capítulo, foi apresentado o projeto do desenvolvimento da bancada de emulação. A importância de estruturar como serão realizadas as conexões, as configurações e as disposições dos componentes para a montagem da bancada é de suma importância para o projeto. Diante dos dados apresentados no ensaio de calibração, avalia-se que a bancada apresenta o comportamento de atenuação considerável, pois quando apresentado

Figura 21 – RSSI com a atenuação máxima de 31,5 dB na bancada em funcionamento.



Fonte: Do autor.

com uma atenuação de 0 dB, teve por volta de -30 dBm e, quando inserida uma atenuação máxima de 31,5 dBm, esperava-se uma adição de potência chegando na casa de -60 dBm, como apresentado na Figura 21.

No próximo capítulo, será feita a validação dos resultados da bancada utilizando o modelo de propagação no espaço livre.

4 Resultados

Neste capítulo, serão apresentados os principais resultados obtidos com o modelo de propagação em espaço livre e comparado com testes realizados na bancada de emulação para a validação dos resultados.

4.1 Validação dos dados por Espaço Livre

Para realizar a validação utilizando o modelo de propagação por espaço livre, será utilizado (2.3), apresentada na subseção 2.2.1, com os parâmetros descritos na Tabela 1. O valor da potência de transmissão utilizada é a potência do próprio módulo ESP32, Figura 22. A frequência de operação utilizada é a própria do Wi-Fi e os ganhos das antenas são com base nas antenas interna (recepção) e externa (transmissão).

Figura 22 – Características do Rádio Wi-Fi do ESP32.

Parameter	Condition	Min	Typical	Max	Unit
Operating frequency range <i>note1</i>	-	2412	-	2484	MHz
Output impedance <i>note2</i>	-	-	<i>note 2</i>	-	Ω
TX power <i>note3</i>	11n, MCS7	12	13	14	dBm
	11b mode	18.5	19.5	20.5	dBm
Sensitivity	11b, 1 Mbps	-	-98	-	dBm
	11b, 11 Mbps	-	-88	-	dBm
	11g, 6 Mbps	-	-93	-	dBm
	11g, 54 Mbps	-	-75	-	dBm
	11n, HT20, MCS0	-	-93	-	dBm
	11n, HT20, MCS7	-	-73	-	dBm
	11n, HT40, MCS0	-	-90	-	dBm
11n, HT40, MCS7	-	-70	-	dBm	
Adjacent channel rejection	11g, 6 Mbps	-	27	-	dB
	11g, 54 Mbps	-	13	-	dB
	11n, HT20, MCS0	-	27	-	dB
	11n, HT20, MCS7	-	12	-	dB

Fonte: Do autor.

Tabela 1 – Parâmetros utilizados para o cálculo da potência de recepção do modelo (2.2.1) de propagação no espaço livre.

Parâmetros	Valores	Unidade de medida
Potência de transmissão (P_t)	20,5	dBm
Ganho da antena de transmissão (G_t)	3	dBi
Ganho da antena de recepção (G_r)	1	dBi
Velocidade da luz no vácuo (c)	3×10^8	m/s
Frequência (f)	2,4	GHz
Distância (d)	1 – 80	m

Fonte: Do Autor.

Para utilizar a potência de transmissão em (2.3), deve-se convertê-la na unidade de *watts* por meio de

$$P_w = \frac{10^{\frac{P_{dBm}}{10}}}{1000}.$$

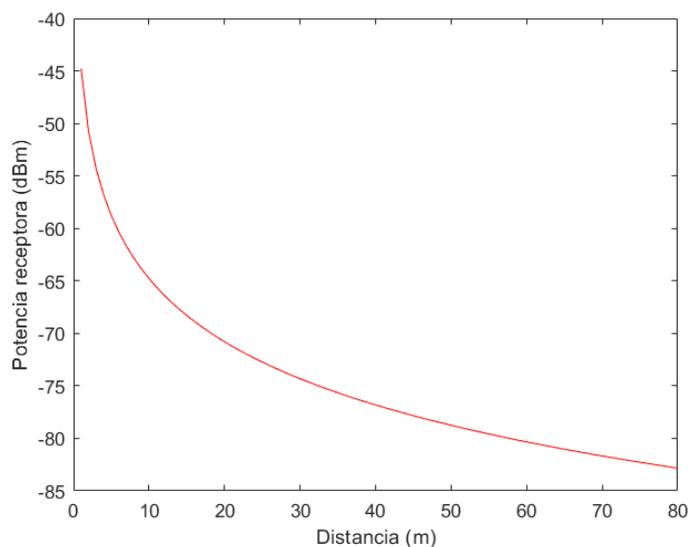
Foi utilizado um vetor de distância de 1 a 80 metros com a intenção de demonstrar mais valores de potências nos resultados. Após, os valores obtidos das potências foram convertidos de *watts* para dBm utilizando

$$P_{dBm} = 10\log_{10}(P_r).$$

Utilizar $10\log_{10}$ auxilia na compreensão dos dados, por poder facilmente comparar e entender a relação entre duas potências, mesmo quando as diferenças são grandes. Um aumento de 10 dB, por exemplo, significa que a potência é 10 vezes maior, enquanto uma diminuição de 10 dB significa que a potência é 10 vezes menor.

Utilizando um *script* no MATLAB, os resultados calculados das potências de recepção são apresentados na Figura 23. Avaliando o gráfico, percebe-se que a potência de recepção diminui enquanto a distância entre as antenas de recepção e transmissão vão aumentando. O modelo de propagação no espaço livre prevê que a potência recebida diminui pela distância de separação entre as antenas de transmissão e recepção. Os cálculos iniciam da distância de 1 metro, pois não seria lógico começar com 0 porque isso implicaria que a antena transmissora está localizada no mesmo lugar da antena receptora.

Figura 23 – Potências obtidas utilizando o modelo de propagação no espaço livre com distância de 1 a 80 metros.

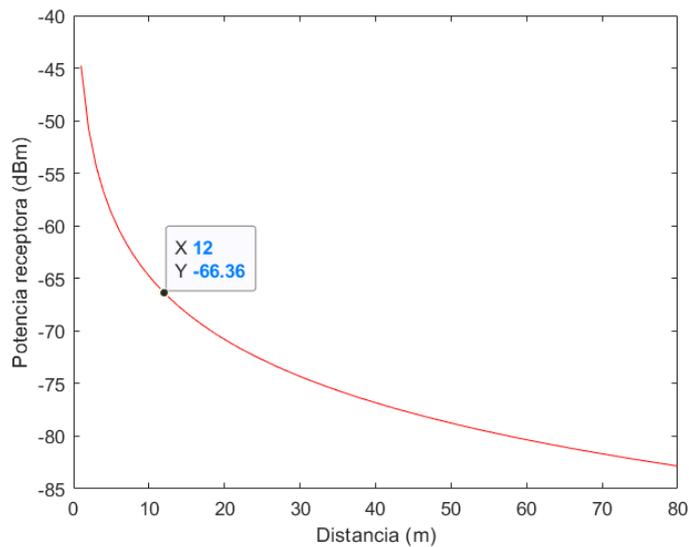


Fonte: Do autor.

Recorrendo à Figura 21 que apresenta os resultados práticos obtidos com a bancada de emulação, nota-se que houve uma variação do sinal entre -61 dBm a -67 dBm, com

a maioria das medições entre -64 dBm a -66 dBm. Comparando os resultados práticos com a curva calculada, Figura 24, pelo modelo de propagação do espaço livre (2.2.1), observa-se que para uma distância entre antenas de 12 metros, a atenuação é de -66 dBm. Portanto, pode-se concluir que a bancada na atual configuração é capaz de emular um canal de comunicação de RF cujas antenas estejam espaçadas de até 12 metros.

Figura 24 – Potências obtidas utilizando o modelo de propagação no espaço livre com distância de 1 a 80 metros.



Fonte: Do autor.

4.2 Considerações finais

Neste capítulo, foram apresentados os resultados para a validação da bancada de emulação com base no modelo de propagação por espaço livre. Apesar da utilização de um modelo matemático simples de propagação, o valor da potência encontrada na bancada ficou próximo aos valores encontrados no modelo, constatando uma emulação de uma rede com a separação das antenas de transmissão e recepção de até 12 metros.

5 Conclusão

O principal objetivo deste trabalho foi realizar estudos e o desenvolvimento referentes a construção de uma bancada de testes de Emulação de Rede de Comunicação sem Fio. As RSSF desempenham um papel significativo em diversas áreas de aplicação devido à sua capacidade de coleta de dados em tempo real de forma eficiente e econômica. O modelo de propagação visa prever a potência recebida nos sistemas de comunicação, pois o caminho, que um sinal pode tomar entre o transmissor e o receptor, pode variar desde a simples linha de visão até um que seja seriamente obstruído por prédios, montanhas, etc. Os protocolos de comunicação utilizados na RSSF tem por objetivo definir como os dispositivos devem comunicar entre si.

Para o começo do desenvolvimento do protótipo, foi utilizada como referência uma bancada produzida em Delforno (2020). Embora tendo alguns percalços para a reprodução fiel da bancada, foram levantados outros equipamentos utilizados na construção da bancada com uma nova abordagem. Foram utilizados módulos ESP32 nas configurações de ponto de acesso e estação e o protocolo de comunicação IEEE 802.11 (Wi-Fi) que define como deve ser a comunicação sem fio entre um dispositivo e um concentrador ou dois dispositivos.

Avaliando os resultados obtidos, utilizando o modelo de propagação em espaço livre e a bancada, demonstra-se que é possível emular o canal de comunicação entre os dois módulos, sem a necessidade de afastar os módulos. Vale ressaltar que para se ter resultados mais precisos, a utilização de outros modelos de propagação mais sofisticados permitiria uma melhor validação dos resultados.

Sendo assim, a bancada de emulação tem um papel crucial em diversos campos da engenharia, pesquisa e desenvolvimento, proporcionando a capacidade de testar, validar e aprimorar sistemas de forma eficiente e segura. É uma ferramenta essencial para a transferência de inovações e progresso tecnológico em diversas áreas da ciência e da engenharia.

5.1 Trabalhos Futuros

Como trabalhos futuros, destacam-se alguns pontos a serem considerados para a complementação desta pesquisa como:

- a) Utilização de outros modelos de propagação para validação dos resultados da bancada.
- b) Desenvolvimento de um banco de dados com informações sobre outros testes realizados, com o intuito de aprofundar as comparações e exemplos com diversos

parâmetros.

- c) Desenvolvimento de uma interface *online*, que possa ser acessada, controlada e testada via internet. Tornando a bancada acessível para pesquisadores do Brasil.
- d) Substituição do Arduino por um ESP32 para padronizar os módulos na bancada e diminuir o número de componentes na bancada.
- e) Adição de um atenuador em série com o existente para verificar o comportamento da transmissão de informações, além de permitir aumentar a atenuação máxima do sinal pela bancada.
- f) Aplicações de técnicas de controle em rede de comunicação.

Referências

- ARDUINO. *Arduino Uno R3*. 2022. Disponível em: <<https://docs.arduino.cc/hardware/uno-rev3>>. Acesso em 16 de abril de 2022.
- BENTO, J. F. L. et al. *Segurança em redes de sensores wireless*. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Electrotécnica e de Computadores Major Telecomunicações, Faculdade de Engenharia da Universidade do Porto, Portugal, 2009.
- CARVALHO, F. B. S. de et al. Aplicacoes ambientais de redes de sensores sem fio. *Revista de tecnologia da informação e comunicação*, v. 2, n. 1, p. 14–19, 2012.
- CAVALCANTI, A. B. d. R. M. *Uma avaliação da interferência entre redes 802.11 G*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2009.
- DELFORNO, L. Bancada de Emulação de Canal de Rádio para RSSF em Internet das Coisas. PUC-Campinas, 2020.
- DEVICES, A. *Datasheet HMC472ALP4/472ALP4E*. [S.l.], 2012.
- FREITAS, R. N. Impacto no desempenho de RSSF com diferentes técnicas de cifragem. Pontifícia Universidade Católica de Campinas, 2018.
- FREITAS, R. N. de; BRANQUINHO, O. C. Segurança em redes de sensores sem fio. *Brazilian Technology Symposium*, 2017.
- GOMES, R. D. et al. Desafios de redes de sensores sem fio industriais. *Revista de Tecnologia da Informação e Comunicação*, v. 4, n. 1, p. 16–27, 2014.
- GUNGOR, V. C.; HANCKE, G. P. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on industrial electronics*, IEEE, v. 56, n. 10, p. 4258–4265, 2009.
- HAYKIN, S.; MOHER, M. *Sistemas modernos de comunicações wireless*. [S.l.]: Bookman Editora, 2009.
- ILYAS, M.; MAHGOUB, I. *Handbook of sensor networks: compact wireless and wired sensing systems*. [S.l.]: CRC press, 2014.
- KARL, H.; WILLIG, A. *Protocols and architectures for wireless sensor networks*. [S.l.]: John Wiley & Sons, 2007.
- LOUREIRO, A. A. et al. Redes de sensores sem fio. In: SN. *Simpósio Brasileiro de Redes de Computadores (SBRC)*. [S.l.], 2003. p. 179–226.
- RAPPAPORT, T. S. *Comunicações sem fio: Princípios e práticas, 2a. edição*. [S.l.]: Sao Paulo: Pearson Education do Brasil, 2009.
- RUFINO, N. M. d. O. *Segurança em redes sem fio: aprenda a proteger suas informações em ambientes wi-fi e bluetooth*. [S.l.]: Novatec Editora, 2019.
- SAUSEN, P. S. et al. Gerenciamento integrado de energia e controle de topologia em redes de sensores sem fio. Universidade Federal de Campina Grande, 2008.

SYSTEMS, E. *ESP32-DevKitC V4 Getting Started Guide*. 2023. <<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>>. Acesso em 06 de setembro de 2023.

SYSTEMS, E. *ESP32 DevKits*. 2023. Disponível em: <<https://www.espressif.com/en/products/devkits>>. Acesso em 03 de maio de 2023.

TANENBAUM, A.; WETHERALL, D. *Redes de computadores*. [S.l.]: Pearson Prentice Hall, 2011. ISBN 9788576059240.

A Anexos

A.1 Código para o Controle da Atenuação

```

// Controle do atenuador com Arduino

int attPins[] = {8, 9, 10, 11, 12, 13};
int attControl[] = {0, 0, 0, 0, 0, 0};
int atenuacao = 1;
void setup() {
    // initialize the serial communication:
    Serial.begin(115200);
    // initialize the ledPin as an output:
    for (int i = 0; i < 6; i++) {
        pinMode(attPins[i], OUTPUT);
    }
}
void loop() {
    // check if data has been sent from the computer:
    // read the most recent byte (which will be from 0 to 255):
    Serial.println(atenuacao);
    // set the attenuator control vector:
    if (atenuacao == 0) {
        // Caso a atenuacao seja 0, desliga todos os pinos:
        for (int i = 0; i < 6; i++) {
            attControl[i] = 1;
        }
    } else if (atenuacao == 1) {
        // Com base na tabela verdade no datasheet do atenuador,
        // foram definidos os valores dos inputs
        // Caso a atenuacao seja 0.5 dB
        attControl[0] = 1;
        attControl[1] = 1;
        attControl[2] = 1;
        attControl[3] = 1;
        attControl[4] = 1;
        attControl[5] = 0;
    } else if (atenuacao == 2) {
        // Caso a atenuacao seja 1 dB
        attControl[0] = 1;
    }
}

```

```
    attControl[1] = 1;
    attControl[2] = 1;
    attControl[3] = 1;
    attControl[4] = 0;
    attControl[5] = 1;
} else if (atenuacao == 3) {
    // Caso a atenuacao seja 2 dB
    attControl[0] = 1;
    attControl[1] = 1;
    attControl[2] = 1;
    attControl[3] = 0;
    attControl[4] = 1;
    attControl[5] = 1;
} else if (atenuacao == 4) {
    // Caso a atenuacao seja 4 dB
    attControl[0] = 1;
    attControl[1] = 1;
    attControl[2] = 0;
    attControl[3] = 1;
    attControl[4] = 1;
    attControl[5] = 1;
} else if (atenuacao == 5) {
    // Caso a atenuacao seja 8 dB
    attControl[0] = 1;
    attControl[1] = 0;
    attControl[2] = 1;
    attControl[3] = 1;
    attControl[4] = 1;
    attControl[5] = 1;
} else if (atenuacao == 6) {
    // Caso a atenuacao seja 16 dB
    attControl[0] = 0;
    attControl[1] = 1;
    attControl[2] = 1;
    attControl[3] = 1;
    attControl[4] = 1;
    attControl[5] = 1;
} else if (atenuacao == 7) {
    // Caso a atenuacao seja 31 dB
    attControl[0] = 0;
    attControl[1] = 0;
    attControl[2] = 0;
```

```
    attControl[3] = 0;
    attControl[4] = 0;
    attControl[5] = 0;
}
// turn the pins high or low according to the attenuator
// control
vector:
for (int i = 0; i < 6; i++) {
    digitalWrite(attPins[i], attControl[i]);
}
Serial.write(atenuacao);
atenuacao++;
if (atenuacao == 8) {
    atenuacao = 0;
}
// Aguarda um tempo antes de passar para a proxima atenuacao:
delay(20000);

}
```

A.2 Código para o AP

```
//ESP32 configurado para o modo de Ponto de Acesso (Access
// Point)

// Bibliotecas
#include <WiFi.h>
#include <WiFiAP.h>
#include <WiFiClient.h>

// Credenciais da rede
const char* ssid      = "Rede_ESP_Romario";
const char* password  = "s3nh4c3r74";

// Porta utilizada pelo servidor
const int serverPort = 80;
// Tamanho dos dados esperados
const int dataSize = 52.0;

WiFiServer server(serverPort);
```

```
void setup() {
  Serial.begin(115200);

  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password);

  Serial.println("Ponto de Acesso (AP) configurado");

  IPAddress IP = WiFi.softAPIP();
  Serial.print("Endereço IP do Access Point: ");
  Serial.println(IP);

  // Inicia o servidor na porta especificada
  server.begin();
  Serial.print("Servidor iniciado na porta: ");
  Serial.println(serverPort);
}

void loop() {
  // Aguarda uma conexão de cliente
  WiFiClient client = server.available();

  if (client) {
    Serial.println("Cliente conectado");

    // Le os dados enviados pelo cliente
    byte data[dataSize];
    size_t bytesRead = client.readBytes(data, dataSize);

    if (bytesRead > 0) {
      delay(100);

      if (bytesRead == dataSize) {
        Serial.println("Dados recebidos corretamente");

        // Verifica se os dados recebidos são um array de 52
        // bytes com valor zero em todas as posições
        bool isZeroData = true;
        for (int i = 0; i < dataSize; i++) {
          if (data[i] != 0) {
            isZeroData = false;
          }
        }
      }
    }
  }
}
```

```
        break;
    }
}

if (isZeroData) {
    Serial.println("Dados_sao_um_array_de_52_bytes_com_valor_zero_em_todas_as_posicoes");
    // Envie uma resposta opcional ao cliente se necessario
    // Envia os dados para o servidor
    WiFiClient client = server.available();
    if (client) {
        int recebido = 1;
        client.println(recebido);
    }
} else {
    Serial.println("Dados_nao_correspondem_ao_esperado");
}

// Encerra a conexao com o cliente
client.stop();

Serial.println("Cliente_desconectado");
}
} else {
    Serial.println("Erro_na_recepcao_dos_dados");
}
}
}
```

A.3 Código para a Estação

```
#include <WiFi.h>
#include <WiFiClient.h>

const char* ssid = "Rede_ESP_Romario"; // Nome da sua rede Wi-Fi
const char* password = "s3nh4c3r74"; // Senha da sua rede Wi-Fi
int aux = 0; //variavel auxiliar
```

```
IPAddress serverIP(192, 168, 4, 1); // IP do servidor (ESP32
    access_point)
int serverPort = 80; // Porta do servidor

void initWiFi() {
    //Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

    Serial.print("Conectando a rede..");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(1000);
    }
    Serial.println("Conectado a rede!");
    Serial.print("Nome:");
    Serial.println(ssid);
    //aux = 1;
    // Serial.print("Endereco IP: ");
    // Serial.println(WiFi.localIP());
}

int sendUplinkData(byte* data, size_t dataSize) {
    //int rssi = WiFi.RSSI(); // Medicao do RSSI antes do envio
    // dos dados
    //int rssi_uplink_before = WiFi.RSSI();
    // Cria uma conexao com o servidor
    WiFiClient client;
    if (client.connect(serverIP, serverPort)) {
        // Envia os dados para o servidor
        client.write(data, dataSize);
        client.println(); // Envia uma quebra de linha para
            // indicar o fim dos dados
        client.flush(); // Limpa o buffer de envio
    }
    float rssi = WiFi.RSSI(); // Medicao do RSSI apos o envio
    // dos dados
    return rssi;
}

int receiveDownlinkData(byte* data, size_t dataSize) {
```

```
//int rssi = WiFi.RSSI(); // Medicao do RSSI antes da
    recepcao dos dados
// Cria uma conexao com o servidor
// Medicao de downlink
WiFiClient client;
client.connect(serverIP, serverPort);

while (client.connected()) {
    while (client.available()) {
        String response = client.readString();
        int numero = response.toInt();
        Serial.print("Recebido/NaoRecebido:␣");
        Serial.println(numero);
    }
}
client.stop();

float rssi = WiFi.RSSI(); // Medicao do RSSI apos a recepcao
    dos dados
return rssi;
}

void setup() {
    Serial.begin(115200);
    initWiFi();
}

void loop() {

    // Cria um array de 52 bytes com valor zero em todas as
        posicoes
    byte dataArray[52] = { 0 };

    // Medicao de uplink
    unsigned long startTime = millis();
    int uplinkRSSI = sendUplinkData(dataArray, sizeof(dataArray))
        ;
    unsigned long endTime = millis();
    unsigned long uplinkTime = endTime - startTime;
    // Medicao de downlink
    startTime = millis();
```

```
int downlinkRSSI = receiveDownlinkData(dataArray, sizeof(
    dataArray));
endTime = millis();
unsigned long downlinkTime = endTime - startTime;

Serial.print("Amostra:␣");
Serial.println(aux);
Serial.print("Uplink␣Time:␣");
Serial.println(uplinkTime);
Serial.print("Uplink␣RSSI:␣");
Serial.println(uplinkRSSI);
Serial.print("Downlink␣Time:␣");
Serial.println(downlinkTime);
Serial.print("Downlink␣RSSI:␣");
Serial.println(downlinkRSSI);

aux++;

delay(1000);
}
```

A.4 Código para coletar os sinais e plotar os histogramas

```
%% Comunicacao Serial
close all
clear all
clc

% Fecha qualquer conexao serial que exista na porta
delete(instrfind({'Porta'},{'COM4'}));

% Cria um objeto de porta serial
s = serial('COM4', 'BaudRate', 115200);

% % Configura as propriedades da porta serial
set(s, 'DataBits', 8);
set(s, 'Parity', 'none');

% Abre a porta serial
fopen(s);
```

```
% Inicializa as variaveis para armazenar os valores recebidos
recebidoNaoRecebido = [];
amostra = [];
uplinkTime = [];
uplinkRSSI = [];
downlinkTime = [];
downlinkRSSI = [];

t = 1;

while t < 2401
a = fgetl(s); % le uma linha de dados da conexao serial e
    armazena em "a"
fprintf('%s',a)

% Extrai os valores de cada linha
valores = strsplit(a, ': ');
valor = str2double(valores{2});

% Armazena os valores nos vetores correspondentes
switch valores{1}
case 'Recebido/NaoRecebido '
recebidoNaoRecebido = [recebidoNaoRecebido, valor];
case 'Amostra '
amostra = [amostra, valor];
case 'Uplink Time '
uplinkTime = [uplinkTime, valor];
case 'Uplink RSSI '
uplinkRSSI = [uplinkRSSI, valor];
case 'Downlink Time '
downlinkTime = [downlinkTime, valor];
case 'Downlink RSSI '
downlinkRSSI = [downlinkRSSI, valor];
end

t = t + 1;
a = 0;
end
```

```
% Fecha a comunicacao serial
fclose(s);

recebidoNaoRecebido_tam = length(recebidoNaoRecebido);
amostra_tam = length(amostra);
uplinkTime_tam = length(uplinkTime);
uplinkRSSI_tam = length(uplinkRSSI);
downlinkTime_tam = length(downlinkTime);
downlinkRSSI_tam = length(downlinkRSSI);

figure()
% Crie um histograma do vetor uplinkRSSI
histogram(uplinkRSSI)
% Adicione rotulos e titulo ao histograma
xlabel('Uplink RSSI (dBm)')
ylabel('Frequencia')

figure()
% Crie um histograma do vetor uplinkRSSI
histogram(downlinkRSSI)
% Adicione rotulos e titulo ao histograma
xlabel('Downlink RSSI (dBm)')
ylabel('Frequencia')
```

A.5 Código para o cálculo do modelo de propagação espaço livre

```
%% Espaco Livre
close all
clear all
clc

Pt = 20.5; %Potencia do sinal transmitido em dBm
Pt1 = (1*10^(Pt/10))/1000; %converte dBm em watts
Gt = 3; %Ganho da antena transmissora
Gr = 1; %Ganho da antena receptora
v = 3e8; %velocidade da luz no vacuo
f = 2.4e9; %frequencia do wi-fi
lambda = v/f; %comprimento de onda
```

```
d = 10; %distancia
L = ((4*pi*d)/lambda)^2; %perdas
Pr = (Pt1*Gt*Gr)/L;
Pr_dBm = 10*log10(Pr);

% Plotar o grafico com as potencias
Pr_dbm_vet = [];

for d1 = 1:1:80
    L_vet = ((4*pi*d1)/lambda)^2;
    Pr_vet = (Pt1*Gt*Gr)/L_vet;
    Pr_dbm_vet(d1) = 10*log10(Pr_vet);
end

figure()
plot(Pr_dbm_vet, 'red')
xlabel('Distancia (m)')
ylabel('Potencia receptora (dBm)')
```