



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
COLEGIADO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

Matheus de Abreu Bozzi

Avaliação de uma Rede LoRa Mesh Acadêmica para a Universidade Federal do Espírito Santo

Vitória, ES

2022

Matheus de Abreu Bozzi

Avaliação de uma Rede LoRa Mesh Acadêmica para a Universidade Federal do Espírito Santo

Monografia apresentada ao Curso de Engenharia de Computação do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Engenharia de Computação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Colegiado do Curso de Engenharia de Computação

Orientador: Vinícius Fernandes Soares Mota

Vitória, ES

2022

Matheus de Abreu Bozzi

Avaliação de uma Rede LoRa Mesh Acadêmica para a Universidade Federal do Espírito Santo/ Matheus de Abreu Bozzi. – Vitória, ES, 2022-
73 p. : il. (algumas color.) ; 30 cm.

Orientador: Vinícius Fernandes Soares Mota

Monografia (PG) – Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Colegiado do Curso de Engenharia de Computação, 2022.

1. LoRa. 2. LoRaWAN. 3. Rede Mesh. 4. Internet das Coisas I. Bozzi, Matheus de Abreu. II. Universidade Federal do Espírito Santo. IV. Avaliação de uma Rede LoRa Mesh Acadêmica para a Universidade Federal do Espírito Santo

CDU 02:141:005.7

Matheus de Abreu Bozzi

Avaliação de uma Rede LoRa Mesh Acadêmica para a Universidade Federal do Espírito Santo

Monografia apresentada ao Curso de Engenharia de Computação do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Engenharia de Computação.

Trabalho aprovado. Vitória, ES, 01 de Agosto de 2022:

**Prof. Dr. Vinícius Fernandes Soares
Mota**
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Rodolfo da Silva Villaça
Universidade Federal do Espírito Santo

Msc. Víctor Manuel García Martínez
Universidade Federal do Espírito Santo

Vitória, ES
2022

Dedico este trabalho à minha família, em especial, ao meu irmão e amigo Vinícius que esteve comigo nessa caminhada. À minha mãe Márcia, que sempre me deu carinho e apoio nas minhas decisões. Ao meu irmão Fabrício, que foi uma pessoa na qual sempre me espelhei para seguir nos estudos. Ao meu pai Sávio, por todo o aporte que possibilitou minha estadia fora de minha casa. À minha, hoje, namorada e amiga Jéssica, que sempre me apoiou e me ajudou nos momentos mais difíceis, com amor e carinho.

Agradecimentos

Agradeço, primeiramente, a Deus por ter me concedido saúde e sabedoria. Muito obrigado meu Deus.

Ao meu orientador Vinícius, que sempre esteve disponível para tirar minhas dúvidas, e sempre cedeu seu ocupado tempo para me dar explicações.

Aos amigos que fiz no Centro Tecnológico da UFES. Ademais, aos grandes professores dos Departamentos de Informática e Engenharia Elétrica da UFES.

Por último, à todos que colaboraram, não só com este trabalho, mas com minha formação como engenheiro.

*“A persistência é o caminho do êxito.
E a perseverança é a mãe da boa sorte.”
(Charles Chaplin; Miguel de Cervantes)*

Resumo

A Internet das Coisas (IoT) possui um princípio no qual os dispositivos do nosso dia a dia estão conectados à Internet por meio de redes de comunicação, que podem tanto enviar quanto receber dados. Geralmente, a conexão de dispositivos finais com a Internet ocorre através de uma rede sem fio, na qual existem diversas tecnologias, entre elas o LoRa. Em particular, a modulação LoRa e seu protocolo LoRaWAN têm amplo destaque por permitirem conectar dispositivos com um longo alcance e baixo consumo energético, caracterizando-se assim como uma rede de Longo Alcance e Baixa Potência (LPWAN). O LoRa foi idealizado para conectar os dispositivos a um gateway, em uma topologia estrela. Todavia, diante do alto custo deste, a utilização de uma rede com topologia em malha, denominada rede LoRa Mesh, na Universidade Federal do Espírito Santo se mostrou uma excelente alternativa, já que os próprios dispositivos se auto comunicam fazendo com que a cobertura aumente gradativamente a cada novo dispositivo inserido na rede. Este trabalho apresenta testes com a implantação de uma rede LoRa Mesh acadêmica, analisando todos os parâmetros essenciais para o seu funcionamento e estudando o protocolo de roteamento por inundação nessa rede Mesh. Ademais, os testes tiveram como objetivo mapear a área de cobertura dos dispositivos e verificar a qualidade do sinal em cada ponto da UFES. Por fim, através da rede LoRa mesh implantada foi possível cobrir, com apenas três dispositivos, cerca de 36 Hectares de área, o que representa 62% da área total da Universidade Federal do Espírito Santo, campus de Goiabeiras.

Palavras-chaves: LoRa. LoRaWAN. Rede Mesh. LPWAN. Internet das Coisas.

Lista de ilustrações

Figura 1 – Comparação entre as Tecnologias (Largura de Banda x Alcance)	15
Figura 2 – Setores com aplicações LoRa	16
Figura 3 – Arquitetura simples de rede sem fio no modo infraestrutura	19
Figura 4 – Arquitetura simples de rede sem fio no modo Ad Hoc	20
Figura 5 – Estrutura da pilha de protocolos LoRaWAN	23
Figura 6 – Topologia Estrela	24
Figura 7 – Arquitetura de rede LoRaWAN (modificado pelo autor)	25
Figura 8 – Classes dos dispositivos LoRaWAN (modificado pelo autor)	26
Figura 9 – Rede Mesh	28
Figura 10 – LoRa Heltec Wireless Stick ESP32	31
Figura 11 – Pinout Diagram LoRa Heltec Wireless Stick ESP32	37
Figura 12 – Criação de diretório e arquivos para a placa Heltec Wireless Stick	38
Figura 13 – Resultado esperado do Build da placa Heltec Wireless Stick	40
Figura 14 – Resultado esperado do Upload da placa Heltec Wireless Stick	40
Figura 15 – Configuração padrão do dispositivo	41
Figura 16 – Dispositivos configurados na rede LoRa mesh UFES	45
Figura 17 – Configuração final da rede LoRa mesh UFES	46
Figura 18 – Esquemático de desenvolvimento, configurações e testes	49
Figura 19 – Mensagens enviadas para o Canal UFES, pelo aplicativo Android	51
Figura 20 – Ícone de conexão, no aplicativo Android	52
Figura 21 – Mapa de Calor na UFES em relação ao Sinal-Ruído	54
Figura 22 – Nó Mesh 1 no Observatório da UFES, ponto 27 da Figura 21	55
Figura 23 – Nó Mesh 1 em frente a Biblioteca Central da UFES	56
Figura 24 – Mapa de Calor na UFES em relação ao Sinal-Ruído	57
Figura 25 – Mapa de cobertura da rede LoRa mesh UFES	59

Lista de tabelas

Tabela 1 – Relação do SNR com o Spreading Factor	23
Tabela 2 – Descrição do dispositivo LoRa	32
Tabela 3 – Ferramentas Utilizadas.	32
Tabela 4 – Softwares de Apoio ao Desenvolvimento do Projeto	33
Tabela 5 – Regiões Definidas.	42
Tabela 6 – Configurações modem_config.	43
Tabela 7 – Dados obtidos pelo nó Mesh 2	52
Tabela 8 – Dados obtidos pelos nós Mesh 2 e Mesh 1	56

Lista de abreviaturas e siglas

ABP	Activation by Personalization
AES	Advanced Encryption Standard
ANATEL	Administração Nacional de Telecomunicações
BLE	Bluetooth Low Energy
BW	Bandwidth
CRC	Cyclic Redundancy Check
CR	Code Rate
CSS	Chirp Spread Spectrum
FEC	Forward Error Correction
FCC	Federal Communications Commission
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPRM	Laboratório de Pesquisa em Redes e Multimídia
MAN	Metropolitan Area Network
PC	Personal Computer
PSK	Pre-Shared key
SF	Spreading Factor
SNR	Signal to Noise Ratio

TCP	Transmission Control Protocol
TI	Tecnologia da Informação
TIC	Information and Communications Technology
TTL	Time to Live
UFES	Universidade Federal do Espírito Santo
WAN	Wide Area Network
WMN	Wireless Mesh Network

Sumário

1	INTRODUÇÃO	14
1.1	Motivação e Justificativa	15
1.2	Objetivos	17
1.2.1	Objetivo Geral	17
1.2.2	Objetivos Específicos	17
1.3	Método de Desenvolvimento do Trabalho	18
1.4	Organização da Monografia	18
2	REFERENCIAL TEÓRICO E TECNOLOGIAS UTILIZADAS	19
2.1	Redes Sem Fio	19
2.2	LoRa	20
2.2.1	Spreading Factor	21
2.2.2	Code Rate	21
2.2.3	Bandwidth	21
2.2.4	Indicadores de Intensidade de Sinal	22
2.2.4.1	RSSI	22
2.2.4.2	Relação Sinal-Ruído	22
2.3	LoRaWAN	23
2.3.1	Classe dos Dispositivos	25
2.3.2	Banda ISM	26
2.3.3	Segurança	27
2.4	Redes Mesh	27
2.4.1	Algoritmos de Roteamento	28
2.4.1.1	Roteamento por Inundação	28
2.4.2	Rede LoRa Mesh	29
2.4.3	Meshtastic	30
2.5	Dispositivos LoRa Heltec Wireless Stick	31
2.6	Tecnologias Utilizadas	32
3	IMPLANTAÇÃO DE UMA REDE LORA MESH PARA A UNIVER- SIDADE FEDERAL DO ESPÍRITO SANTO	35
3.1	Ambiente de Desenvolvimento	35
3.1.1	Preparação do ambiente	35
3.2	Desenvolvimento	37
3.2.1	Ambiente de Compilação	37
3.2.2	Adicionando o próprio Hardware	37

3.2.2.1	platformio.ini	38
3.2.2.2	variant.h	39
3.2.3	Testando o próprio Hardware	39
3.3	Configurando os Nós na rede mesh	41
3.3.1	Definir o nome do Nó	41
3.3.2	Definir a Região do Nó	42
3.3.3	Definir o Canal do Nó	42
3.3.3.1	Criação do Canal	43
3.4	Funcionamento da rede LoRa mesh	45
3.4.1	Roteamento por Inundação (Flooding)	46
3.4.2	Aplicativo Android	47
3.4.3	Interface Web	48
3.4.4	Arquitetura de Funcionamento	48
4	AVALIAÇÃO DA REDE LORA MESH	50
4.1	Metodologia	50
4.2	Teste de alcance com dois nós	50
4.2.1	Nó M2	50
4.2.2	Nó M1	51
4.2.3	Resultados com dois nós	52
4.3	Teste de alcance com três nós	55
4.3.1	Resultados com três nós	55
4.4	Cobertura da rede LoRa mesh UFES	58
5	CONCLUSÃO	60
5.1	Considerações Finais	60
5.2	Trabalhos Futuros	62
	REFERÊNCIAS	63
	APÊNDICES	67
	APÊNDICE A – ARQUIVOS DE CONFIGURAÇÕES	68
	APÊNDICE B – CONFIGURAÇÕES DOS DISPOSITIVOS	70

1 Introdução

Nos últimos anos, a tecnologia de comunicação de dados sem fio se desenvolveu consideravelmente, possibilitando a troca de quantidades imensas de informações. No entanto, isso resulta na necessidade de equipamentos mais complexos para processar esses grandes volumes de dados. Diante disso, o aumento da complexidade exige melhores hardwares e com um consumo energético mais alto, além dos maiores custos financeiros. Todavia, essas barreiras dificultam a implantação da Internet das Coisas (IoT) em larga escala em áreas rurais ou mesmo em áreas urbanas de cidades, pois aumenta o custo financeiro de manter cada dispositivo conectado (SOUZA, 2019).

A Internet das Coisas está estimulando cada vez mais a descoberta e pesquisas sobre novas infraestruturas de rede para comunicação de dados de longa distância, principalmente para as chamadas cidades inteligentes (SCHAFFERS et al., 2011). Neste âmbito, as tecnologias e padrões de comunicação por radiofrequência têm sido amplamente utilizados. Essas tecnologias geralmente consistem em dispositivos que realizam detecção, coleta e transmissão de dados.

No campo da comunicação sem fio, pode-se diferir duas classificações de redes: LAN (Local Area Network) e WAN (Wide Area Network), onde a essencial diferença é o seu alcance. Para o WAN, a capacidade de alcance é maior, podendo abranger áreas maiores como cidades. Já o LAN, possui alcance limitado e sua aplicabilidade é voltada para redes com baixa dispersão. Por isso, a escolha de rede é feita conforme a aplicação.

Desse modo, diversas das aplicações IoT necessitam conectar dispositivos mais distantes uns dos outros (por exemplo, espalhados entre cidades). Nessa situação, a rede sem fio e dispositivos com baixo consumo de energia são oportunos, visto que podem funcionar com uso de baterias. Dito isso, são introduzidas as chamadas Redes de Longa Distância de Baixa Potência (LPWAN - *Low Power Wide Area Network*). Dentre as LPWANs, faz notar-se a Rede de Longa Distância LoRa (LoRaWAN - *LoRa Wide Area Network*) que utiliza a tecnologia LoRa em seus módulos transceptores (LoRa Alliance, 2022a).

LoRa é propriedade da empresa Semtech, e emprega um método de espalhamento espectral chamado de Chirp Spread Spectrum (CSS). O protocolo LoRaWAN, que possui código aberto e é mantido pela LoRa Alliance, detalha o funcionamento de uma rede de dispositivos que utilizam LoRa na camada física, englobando sua conexão a um servidor de rede, de modo que possa ser conectado à Internet. Além disso, LoRaWAN é um protocolo responsável pela segurança, confirmação de mensagens, gerenciamento, ativação de dispositivos, entre outras finalidades (CENTENARO et al., 2016).

Ao que se refere a rede mesh, neste tipo de topologia de rede sem fio, a grande vantagem é que os nós se comunicam entre si utilizando vários saltos para encaminhar mensagens para seus destinos finais. Estes, podem se conectar a um ponto de acesso da rede mesh usando Ethernet ou sem fio IEEE 802.11. Com esse modelo há possibilidade de adicionar novos pontos de conexão na rede, dessa forma, ela fornece uma área de cobertura maior fazendo a distribuição da conexão na rede de acordo com a necessidade do usuário (PASSOS et al., 2005).

Além disso, as redes mesh são formadas por grupos de dispositivos que se baseiam em algoritmos de roteamento cooperativo (PASSOS et al., 2005), e podem se comunicar sem a necessidade de um Gateway central, isto é, uma rede centralizada que possui um alto custo monetário de implantação.

1.1 Motivação e Justificativa

A tecnologia LoRa possui diversas vantagens, as principais são a longa distância de transmissão dos dados aliada ao baixo custo energético, e a essencial motivação desta proposta de estudo está fundamentada nas características inerentes da camada física de LoRa, através da grande cobertura de longo alcance -hipoteticamente 10 km para áreas sem obstáculos- e com um consumo energético muito baixo. Ademais, as redes com LoRa, tais quais usam o padrão LoRaWAN, preenchem uma lacuna tecnológica para redes celulares, Wi-Fi e Bluetooth Low Energy (BLE) que exigem alta largura de banda ou energia, ou têm alcance limitado ou não podem penetrar profundamente em ambientes internos, como pode-se observar na Figura 1. Assim, oferece uma solução eficiente, flexível e econômica para problemas do mundo real (Semtech, 2022).

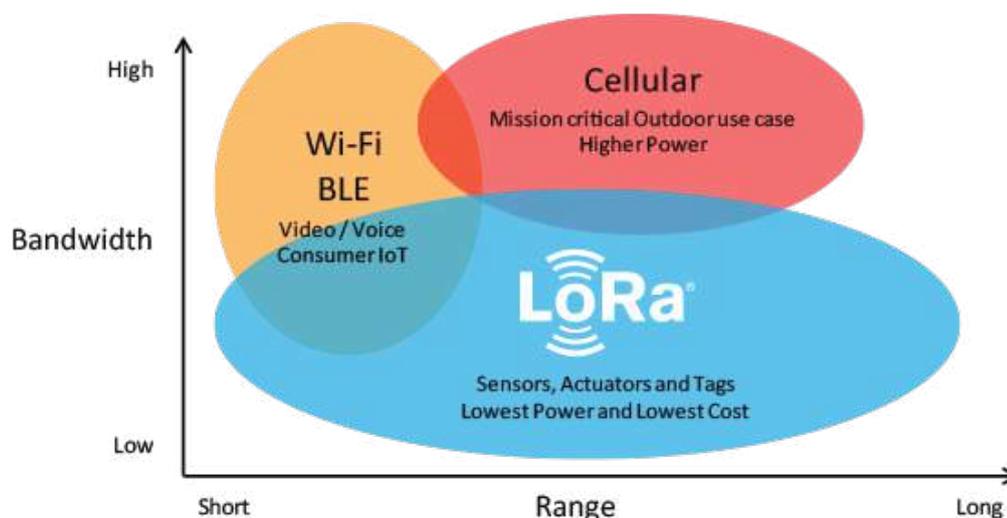


Figura 1 – Comparação entre as Tecnologias (Largura de Banda x Alcance)

Fonte: <https://www.semtech.com/lora>

Por esse motivo, a tecnologia LoRa está empolgando diversos setores, como agricultura inteligente, edifícios, cidades, meio ambiente, saúde, residências, controle industrial, cadeia de suprimentos e logística, serviços públicos e muito mais na Figura 2 (Semtech, 2022). Isso requer uma atenção especial para assuntos acadêmicos dessa categoria, já que possuem alguns desafios tais quais perdas de pacotes no caminho e da suscetibilidade à interferência, principalmente em áreas com densidade urbana, já que possuem maiores obstáculos ao sinal, o que tende a afetar o desempenho LoRa.



Figura 2 – Setores com aplicações LoRa
Fonte: <https://www.semtech.com/lora>

Além disso, o LoRa atua em bandas de radiofrequência ISM, ou seja, são porções do espectro de rádio reservadas internacionalmente para fins industriais, científicos e médicos (SONG; NUAYMI; LAGRANGE, 2016). No Brasil é utilizada as faixas de frequência AU915-928 (915 MHz a 928 MHz, o mesmo utilizado na Austrália) de acordo com a LoRa Alliance (2022b). Com isso, para pesquisas científicas não há necessidade de requerer uma certificação na FCC para usar essa faixa de radiofrequência. Além do mais, o protocolo LoRaWAN fornece segurança de ponta a ponta com algoritmos criptográficos AES.

Porém, a rede LoRa foi criada com intuito de ser utilizada com a topologia de rede estrela, onde o Gateway principal tem como objetivo receber os dados de todos

os end-nodes e após isso inserir os dados nos servidores de rede (CENTENARO et al., 2016). Também por se tratar de um componente mais complexo, o Gateway possui um preço bastante elevado, prejudicando assim, a implantação em muitos lugares. Dito isso, a topologia mesh se torna adequada e pode utilizar um de seus end-nodes que possui acesso aos servidores de rede para representar o Gateway, além de possuir diversas vantagens sobre outras tecnologias, incluindo o longo alcance, baixo custo, implantação fácil e incremental e tolerância a falhas (PASSOS et al., 2005).

As redes mesh sem fio comunitárias estão se tornando uma solução atraente para fornecer acesso e serviços para vários campos de aplicações. Assim, a utilização e avaliação de uma rede LoRa mesh acadêmica para a UFES se alinha com o conceito de cidades inteligentes, que integram as tecnologias de informação e comunicação (TICs) para auxiliar em pesquisas futuras e as demandas de seus pesquisadores.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem como objetivo geral avaliar uma rede LoRa utilizando a topologia Mesh, mapeando e aferindo a qualidade do sinal em diversos pontos na Universidade Federal do Espírito Santo.

1.2.2 Objetivos Específicos

- Aplicar os conceitos de IoT com comunicação sem fio de longo alcance para a execução de uma rede LoRa Mesh.
- Analisar a topologia Mesh em uma rede LoRaWAN.
- Conhecer a modulação LoRa para conseguir configurar corretamente os *end-nodes*.
- Estudar a configuração de diversos *end-nodes* na mesma rede e o protocolo de roteamento entre eles.
- Utilização de ferramentas auxiliares disponíveis na *web* para executar a rede LoRa Mesh.
- Mapeamento e testes de alcance entre os dispositivos sem e com a rede Mesh.
- Conhecer circunstanciadamente os componentes da rede, tanto a nível de Hardware como Software, consequentemente podendo avaliar o alcance, qualidade do sinal e perdas de pacote.

1.3 Método de Desenvolvimento do Trabalho

Para obter os resultados e conclusões acerca da problematização apresentada neste trabalho, o primeiro passo foi um estudo sobre redes de computadores e Internet, as camadas de protocolos e seus modelos, camada de aplicação, camada de transporte, camada de rede, camada de enlace, camada física, redes sem fio, redes mesh, algoritmos de roteamento, segurança e gerenciamento de rede.

Para o segundo passo, também foi necessário um conhecimento teórico em telecomunicações para compreender a modulação LoRa, o espalhamento espectral, Chirped-FM modulation, sensibilidade de recepção, relação sinal ruído, o alcance de longas distâncias e baixas taxas de transmissão alinhado com um baixo consumo energético. No âmbito da rede LoRaWAN, o conhecimento prévio sobre topologia estrela e mesh, segurança, classes de sensores, padrões internacionais e princípio de funcionamento foi imprescindível para antes da prática com os Hardwares.

Com o que foi dito acima, os próximos passos foram referentes aos estudos relacionados ao objetivo do trabalho, sendo feitas as implementações e testes (parte prática), tanto em Hardware quanto em Software. A documentação e o código estão hospedados no link <https://github.com/mathbozzi/rede-LoRa-Mesh-UFES>, que é uma plataforma digital baseada em nuvem com um sistema de controle de versões, para possíveis contribuições, já que se trata de um estudo incremental que poderá ter futuras melhorias.

As implantações e testes foram feitos em pesquisa de campo, mais especificamente no campus da UFES de Goiabeiras, Vitória-ES, para a coleta de dados, assim, realizado os estudos de caso para verificação de melhorias na rede mesh. Por último, as soluções e respostas do problema foram discutidas e apresentadas, com embasamento teórico.

1.4 Organização da Monografia

Além desta introdução, este modelo de monografia é composto por outros quatro capítulos:

- O Capítulo 2 apresenta os aspectos relativos ao conteúdo teórico relevante para o trabalho e as tecnologias utilizadas;
- O Capítulo 3 apresenta a principal contribuição do trabalho;
- O Capítulo 4 apresenta os resultados e avaliações sobre os testes feitos em campo;
- O Capítulo 5 apresenta as considerações finais do trabalho;

2 Referencial Teórico e Tecnologias Utilizadas

2.1 Redes Sem Fio

As redes sem fio caracterizam-se pelo fato de não utilizarem cabos metálicos ou fibra óptica para criar a ligação entre os nós da rede sendo a ligação feita por ondas eletromagnéticas que se propagam pelo ar. Com esta característica, consequentemente, pode reduzir o custo de conexão à rede e garante a mobilidade dos nós da rede (BUNGART, 2017).

As classificações das redes sem fio são das mais diversas, porém a mais comum se dá através da área envolvida pela rede. Algumas mais conhecidas são a LAN, utilizada em rede doméstica, a MAN que é uma rede metropolitana, usada geralmente por provedores de serviço de internet, a WAN, que são utilizadas para interligar as redes MAN, podendo assim abranger grandes áreas (VENTURY, 2016).

Possui também outros métodos, voltados ao modo de operação da rede. Um modo no qual os dispositivos se conectam a um *Access Point* (AP), conhecidos também como estação-base, onde ficam encarregados pelo serviço de atribuição de endereço e roteamento de pacotes, é o modo infraestrutura, como mostra a Figura 3 (KUROSE; ROSS, 2013).



Figura 3 – Arquitetura simples de rede sem fio no modo infraestrutura
Desenvolvido pelo Autor

Além do modo infraestrutura, um modo que não demanda o uso de uma estação-base é conhecido como Ad Hoc. Neste, os nós se comunicam entre si (Figura 4) e estabelecem

a atribuição/tradução de endereços, roteamento, entre outros diversos serviços que uma rede possui (TANENBAUM, 2011).

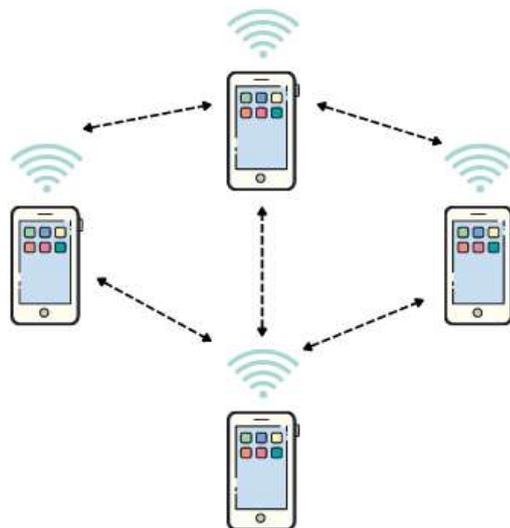


Figura 4 – Arquitetura simples de rede sem fio no modo Ad Hoc
Desenvolvido pelo Autor

Através desses conceitos vistos, pode-se adiantar algumas topologias de rede, que serão mostradas adiante nas seções 2.3 e 2.4. São as denominadas “Topologia Estrela” e “Topologia em Malha ou Mesh”. A topologia em malha, está associada a Figura 4, e possui uma maior confiabilidade e estabilidade, já que os dispositivos estão conectados entre si, uma falha particular ou até mesmo comunitária de alguns nós não é o suficiente para derrubar a conexão.

Ademais, a topologia em malha permite que os nós sempre possam enviar os pacotes de dados pela rota mais eficiente. Porém, conectar esses dispositivos nessa topologia requer um planejamento maior, sendo a implementação trabalhosa e cara. Enquanto a topologia estrela, associada com a Figura 3, é mais adequada ao gerenciamento de rede, já que a interrupção isolada de um nó não desarmoniza a rede, pois o controle acontece por um Hub central, entretanto isso causa uma dependência de que este esteja sempre em funcionamento. (PAAVOLA; LEIVISKA, 2010).

2.2 LoRa

A tecnologia LoRa é patenteada pela empresa Semtech (Semtech, 2022), e esta possui uma grande documentação para que terceiros auxiliem e aprimorem a tecnologia, a chamada LoRa Alliance (LoRa Alliance, 2022a), que foi fundada em 2015. É uma associação aberta e sem fins lucrativos dedicada à padronização de redes sem fio de longa distância de baixa potência (LPWAN) e à promoção global do padrão aberto LoRaWAN.

LoRa é a camada física da tecnologia responsável por modular o sinal utilizando uma técnica de *Chirp Spread Spectrum* (CSS), que espalha o sinal sobre uma larga faixa de frequência, aprimorada para aplicações de longo alcance. Chirps são sinais no qual a frequência pode aumentar ou diminuir ao decorrer do tempo e tem o propósito de codificar as informações. Além disso, proporciona que o sinal percorra longas distâncias e seja demodulado. Simultaneamente, também é utilizado a técnica de *Forward Error Correction* (FEC), aumentando a sensibilidade do receptor e tornando o sinal robusto ao que tange interferência a ruídos (MEKKI et al., 2019).

Uma mensagem LoRa pode ser de Uplink, no qual são enviadas pelos end-nodes para o servidor, e Downlink, que são enviados do servidor para os dispositivos end-nodes. A estrutura de mensagens é parecida em ambos os casos, sendo que é adicionado à mensagem um código de verificação CRC na camada física apenas quando for do tipo Uplink, isso garante a integridade do pacote (LoRa Alliance, 2022c).

Existem certos parâmetros que personalizam a modulação LoRa, influenciando diretamente na transmissão dos pacotes, esses fatores serão expostos abaixo.

2.2.1 Spreading Factor

O Fator de espalhamento (SF - *Spreading Factor*) é o parâmetro primordial da modulação LoRa. Ele possui a finalidade de melhorar a transmissão de informações entre os dispositivos, fazendo com que a mensagem seja enviada dividida em vários chirps, maximizando a sensibilidade do receptor. Este fator de espalhamento pode ser escolhido para cada dispositivo, de acordo com a sua finalidade. Quanto aos valores possíveis, são de 7 a 12, e quanto maior o SF, maior também o consumo energético usado por cada bit, fazendo com que maior seja o alcance entre o transmissor e receptor. (ADELANTADO et al., 2017).

2.2.2 Code Rate

A taxa de código (CR - *Code rate*) é utilizada pelo FEC e define o número de bits com finalidade de repetição da mensagem, isso tem a finalidade de recuperar erros eventuais nos pacotes (DELLALIBERA; ARTEN, 2018). Quanto ao número de CR, pode ser de 1 a 4, e quanto maior, maior a segurança de não haver perdas, todavia, aumenta o tempo de transmissão. (BOR; VIDLER; ROEDIG, 2016).

2.2.3 Bandwidth

A Largura de Banda (BW - *Bandwidth*) é o conceito que determina a medida da capacidade de transmissão, e pode assumir três valores: 125 kHz, 250 kHz ou 500kHz. Ele está relacionado ao chirp rate, isto é, a quantidade de vezes por segundo que um modulador

altera a frequência, definindo a largura de banda do sinal modulado. Em outras palavras, um BW alto possui uma maior taxa de dados e um menor tempo para ser transmitido, entretanto, propicia uma sensibilidade baixa, isso pelo ruído que surge. Já para BW baixo o que ocorre é o oposto, aumenta a sensibilidade mas prejudica a taxa de dados. (SILVA, 2021).

2.2.4 Indicadores de Intensidade de Sinal

Para a comunicação sem fio, um receptor precisa de uma boa intensidade de sinal e uma relação sinal-ruído. Abaixo será mostrado informações sobre dois indicadores de intensidade de sinal mais usados.

2.2.4.1 RSSI

A Indicação de Força do Sinal Recebido (RSSI - *Received Signal Strength Indicator*) é a potência do sinal recebido em miliwatts e é medida em dBm. Este valor pode ser usado para medir quão bem o receptor “escuta” o sinal do remetente.

Como o LoRaWAN suporta comunicação bidirecional, o RSSI é uma medida importante para gateways e dispositivos finais. RSSI está em dBm e seu valor é negativo. Quanto mais próximo o valor RSSI estiver de zero, mais forte será o sinal recebido. (TTN, 2022).

2.2.4.2 Relação Sinal-Ruído

De acordo com a TTN (2022), a relação sinal-ruído é a razão entre a potência do sinal e a potência do ruído que corrompe o sinal. Em outras palavras, SNR (*Signal-to-Noise Ratio*) é um valor que avalia o sinal com base no ruído que está sendo visto.

O ruído representa qualquer sinal externo que interfira no seu sinal, ele pode ser originado por outros dispositivos sem fio, como telefones sem fio, dispositivos de micro-ondas, etc.

O SNR pode ser calculado usando a seguinte fórmula:

$$SNR(dB) = P_{received_signal}(dBm) - P_{noise}(dBm)$$

Uma relação sinal-ruído positiva significa que a potência do sinal é maior que a potência do ruído, portanto, o receptor é capaz de demodular o sinal. Se o RSSI estiver abaixo do nível de ruído, o sinal não poderá ser demodulado. No entanto, LoRa pode demodular sinais abaixo do nível de ruído, e de acordo com a Tabela 1, pode-se ver alguns valores possíveis para o SNR. (TTN, 2022).

Tabela 1 – Relação do SNR com o Spreading Factor

Fator de Espalhamento	Chips/Símbolos	SNR
7	128	-7.5 dB
8	256	-10 dB
9	512	-12.5 dB
10	1024	-15 dB
11	2048	-17.5 dB
12	4096	-20 dB

Observe que o fator de espalhamento (Spreading Factor) deve ser conhecido antecipadamente nos lados de transmissão e recepção do link, pois diferentes fatores de espalhamento são ortogonais entre si.

Os valores característicos de operação para LoRa estão entre -20dB e +10 dB, e quanto maior esse valor, maior a chance de não estar corrompido e ser entregue. O LoRa consegue desmodular sinais que estão entre -7.5dB e -20dB abaixo do nível de ruído (Semtech, 2022) e esses valores estão relacionados com o Fator de Espalhamento que quanto maior, menor pode ser o SNR que o dispositivo é capaz de demodular, por isso esse indicador de intensidade é muito utilizado para verificar a qualidade do sinal.

2.3 LoRaWAN

O termo LoRaWAN caracteriza o protocolo baseado em um modelo de camadas e implementa uma pilha de protocolos em aplicações que utilizam a modulação LoRa, explicitadas acima (Figura 5)(OLIVEIRA et al., 2019). Esse protocolo, que é aberto à comunidade, foi desenvolvido inicialmente pela empresa Cycleo e é mantido pela LoRa Alliance, que é um grupo formado por várias empresas do ramo de TI como: Semtech, IBM, Microchip, etc.

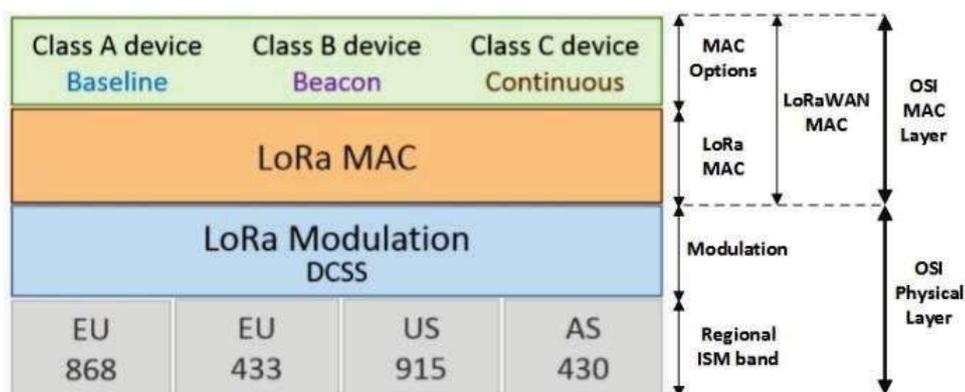


Figura 5 – Estrutura da pilha de protocolos LoRaWAN
Fonte: (OLIVEIRA et al., 2019)

LoRaWAN é um dos protocolos que utilizam o controle de acesso ao meio (MAC - *Media Access Control*), já que possui propriedade interessante: não existem endereços MAC iguais. Esse endereço faz com que vários dispositivos se comuniquem com um gateway (AUGUSTIN et al., 2016), baseando-se em uma rede com topologia do tipo estrela (Figura 6), no qual existe um nó principal (gateway) e os diversos dispositivos mandam e recebem informações a partir do gateway (ADELANTADO et al., 2017), portanto a comunicação ocorre em salto único.

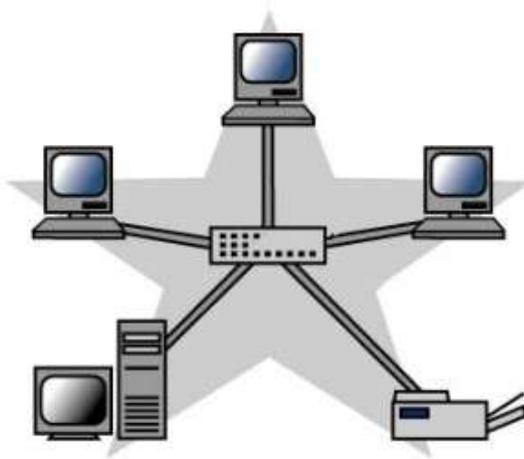


Figura 6 – Topologia Estrela

Fonte: <http://fabrica.ms.senac.br/2013/07/redes-de-computadores-parte-ii/>

Este tipo de topologia é coerente pelo fato de que se todos os dispositivos se conectarem com todos a complexidade aumentaria, o que não traria benefício ao tocante de economia de energia, que é uma das características dos dispositivos LoRa e um dos princípios fundamentais do IoT.

Portanto, a arquitetura completa possui os dispositivos finais (End Nodes), concentradores (gateways), servidor de rede (Network Server) e o servidor de aplicação (Application Server), como mostra a Figura 7 (LoRa Alliance, 2015). Esses serão explicitados abaixo.

1. Servidor de aplicação: São onde os dados coletados pelos dispositivos finais são vistos pelos usuários, podendo processá-los e armazenar os dados em algum banco de dados, por exemplo. O desenvolvedor é o principal responsável pela aplicação e seu objetivo.
2. Servidor de rede: É incumbido de realizar o controle do fluxo de pacotes, validando as informações e encaminhando para os servidores de aplicação adequados. Em outras palavras, fazem o gerenciamento da rede, recebendo as informações dos gateways e encaminhando para o usuário final. Neste estágio também é responsável pela segurança, decodificação e verificação de erros.

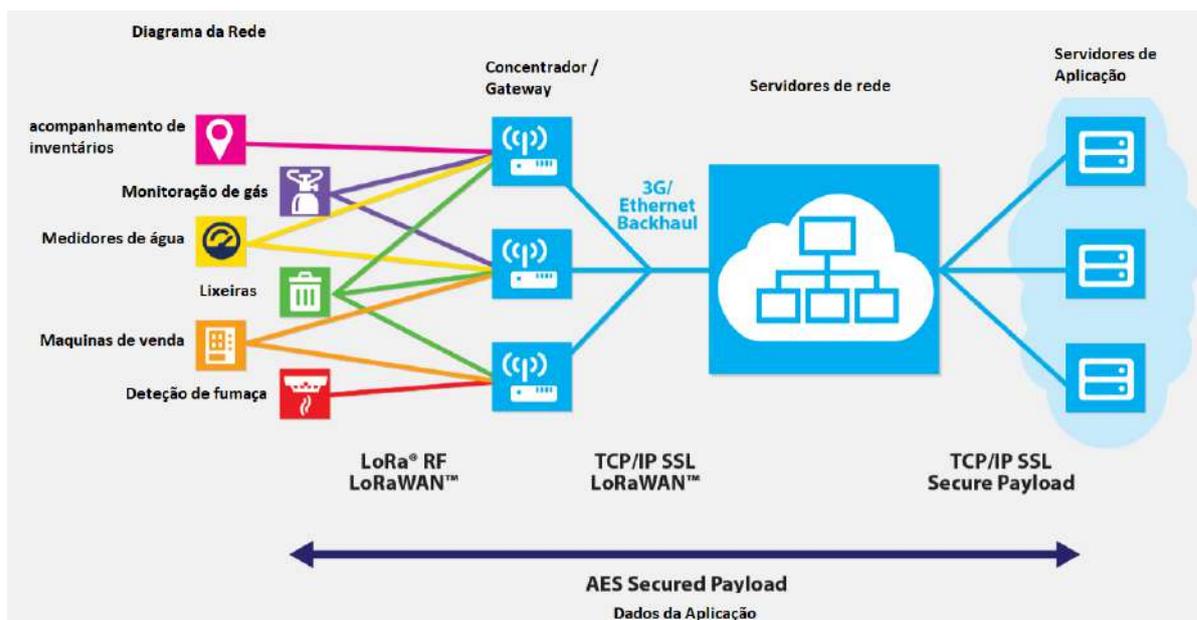


Figura 7 – Arquitetura de rede LoRaWAN (modificado pelo autor)

Fonte: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf#page=8>

3. Gateway: São os intermediadores que recebem os bits enviados pelos dispositivos finais, encaminhando para o servidor de rede por TCP/IP, ou seja, faz o papel entre a interface LoRa e o servidor de rede. Pode haver inúmeros gateways que recebam os pacotes dos dispositivos finais e enviem o mesmo pacote ao servidor de rede, já que este filtra as informações.
4. Dispositivos Finais: São os sensores/atuadores que utilizam a tecnologia LoRa com intuito de se comunicarem com os concentradores. Eles enviam as informações/medições referentes a aplicação que fora desenvolvida.

Dessa maneira, nota-se que cada aplicação LoRa pode assumir diferentes objetivos, isso faz com que algumas classes de dispositivos se difiram entre si. Será visto a seguir as classes A, B e C e o que representam no contexto LoRaWAN.

2.3.1 Classe dos Dispositivos

Dentre as classes de dispositivos existentes chamadas de A, B e C, a principal peculiaridade que as diferem é relacionada ao tempo de latência das mensagens, ou seja, o tempo de espera que uma solicitação leva para ser transferida de um ponto para outro, o que interfere no tempo de vida útil da bateria dos dispositivos. A Figura 8 representa as classes dos dispositivos (LoRa Alliance, 2015).

- Classe A: Os dispositivos desta classe admitem comunicação bidirecional, ou seja, quando o dispositivo envia uma mensagem ao servidor (*UpLink*) ele espera por duas



Figura 8 – Classes dos dispositivos LoRaWAN (modificado pelo autor)

Fonte: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf#page=10>

mensagens enviadas do servidor (*Downlink*), em um breve período de tempo. Caso não seja recebida nenhuma mensagem, ele irá fazer o mesmo processo e aguardar uma mensagem. Pelo fato de não estar sempre disponível para comunicação, faz com que a economia energética seja alta. Essa é a classe básica dos dispositivos LoRa, diferente das B e C que são opcionais.

- Classe B: Os dispositivos desta classe são bidirecionais e disponibilizam janelas de recepção em tempos estipulados (agendados) para comunicação downlink. Essa prática acontece por meio de beacons, que são sinais enviados pelo gateway rotineiramente, provocando a sincronização dos dispositivos. Dessa forma o servidor de rede é capaz de enviar os pacotes quando o dispositivo está ativo.
- Classe C: Os dispositivos desta classe são bidirecionais e estão sempre disponíveis para receber dados do servidor. A recepção só permanece indisponível quando o dispositivo final está transmitindo algo. Isso causa um efeito de diminuição da latência, por outro lado um aumento de consumo energético.

2.3.2 Banda ISM

Para padronizar o uso das radiofrequências existem organizações internacionais e nacionais, responsáveis por administrar o espectro ao redor do mundo, evitando assim interferências. A ANATEL (Agência Nacional de Telecomunicações) é o órgão regulador do Brasil.

Com relação às bandas ISM, são bandas não licenciadas reservadas para dispositivos periféricos com finalidades industriais, científicas e médicas, é permitido no Brasil as bandas entre 902MHz e 928MHz, mas nem todas são liberadas já que as bandas entre de 907,5 MHz e 915 MHz são proibidas, sendo assim, no Brasil pode-se usar as bandas entre 902 MHz a 907,5 MHz e 915 MHz a 928 MHz (ANATEL, 2017; LoRa Alliance, 2022b).

2.3.3 Segurança

Com o uso da tecnologia sem fio crescendo, os incidentes cibernéticos relacionados à integridade, confidencialidade e autenticações devem ser aprimorados para que sejam suprimidas as ameaças e vulnerabilidade (RIZZETTI, 2018). Com isso, a rede LoRaWAN utiliza criptografia AES (*Advanced Encryption Standard*) em dois níveis para a segurança, uma para as aplicações e outra para a própria rede. Esta assegura a autenticação do nó na rede, já a de nível de aplicação certifica que o operador da rede não obtenha os dados do usuário final (MEIRELES, 2018).

Para realizar a conexão dos dispositivos finais, uma rede LoRaWAN possui dois métodos: o ABP (*Activation by Personalization*) no qual os dispositivos finais recebem uma chave fixa, única, a qual permanece ao longo de toda a vida útil do dispositivo. O outro se trata do método OTAA (*Over-The-Air Activation*) no qual os dispositivos possuem uma chave raiz e quando ingressam na rede LoRaWAN recebem chaves de sessão de maneira dinâmica, ou seja, a cada nova conexão é alternada a chave (RABUSKE, 2017).

2.4 Redes Mesh

Uma rede de malha sem fio (WMN - *Wireless Mesh Network*) é uma rede sem fio multi-hop que é composta de vários nós/roteadores, que passam a se comportar como uma única e grande rede, como mostra a Figura 9. Isto significa que os AP's (*Access point* - Ponto de acesso) de malha atuam como roteadores de malha que encaminham o tráfego para o destino através de outros roteadores mesh, saltando entre si e chegando ao dispositivo final que conecta a malha à Internet (DELY; KASSLER; BAYER, 2011).

De acordo com Peterson e Davie (2013), as malhas também são susceptíveis a falhas, fornecendo várias rotas para uma mensagem seguir de um ponto A a outro ponto B. Uma rede em malha pode ser ampliada de forma incremental, com custos incrementais. Apesar disso, a rede em malha exige que os nós não base tenham um nível de sofisticação em seu hardware e software, consequentemente aumentando os custos por unidade e o consumo de potência, sendo assim um ponto desfavorável para dispositivos alimentados por bateria.

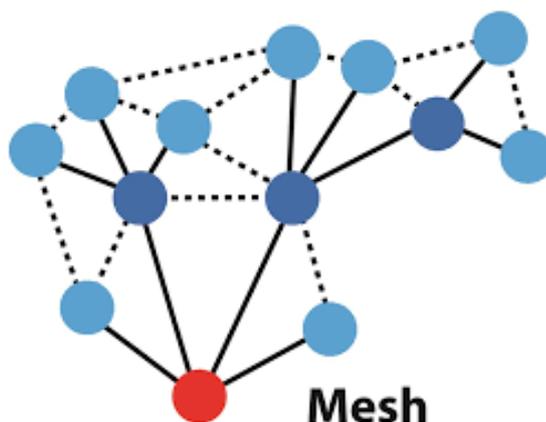


Figura 9 – Rede Mesh

<https://wiki.sj.ifsc.edu.br/index.php/PJI11101-2020-2>

2.4.1 Algoritmos de Roteamento

A camada de rede deve estabelecer a rota tomada pelos pacotes ao avançar de um remetente a um destinatário. Os algoritmos que calculam essas rotas são designados como algoritmos de roteamento. Em diversos casos, é possível transmitir mensagens de um nó a outro por diferentes caminhos e a tarefa do roteamento é determinar bons caminhos (KUROSE; ROSS, 2013). Desse modo, o roteamento em redes mesh tem sido um campo de pesquisa muito ativo e diferentes abordagens para roteamento foram e são investigadas. Várias abordagens têm um sabor de rede ad-hoc (por exemplo, AODV ou B.A.T.M.A.N.), enquanto outros são mais influenciado por protocolos de roteamento de LAN clássicos (por exemplo, OLSR) (DELY; KASSLER; BAYER, 2011).

A funcionalidade desses protocolos de roteamento é bastante limitada e as extensões são difíceis. Por exemplo, baseado em fluxo de roteamento, onde os fluxos tomam caminhos diferentes através da rede, chegar a um destino é difícil de implementar com os tradicionais protocolos de roteamento. Os protocolos de roteamento para essas redes consistem em cinco componentes: descoberta de rota, seleção de rota, manutenção de rota, encaminhamento de dados e representação e métrica de rota (LUNDELL et al., 2018). No entanto, esse roteamento baseado em fluxo permite casos de uso interessantes, como balanceamento de carga. Outra tarefa desafiadora é permitir a mobilidade do cliente na rede mesh (DELY; KASSLER; BAYER, 2011).

2.4.1.1 Roteamento por Inundação

O Roteamento por Inundação faz parte dos protocolos de roteamento por difusão (broadcast) e para um grupo (multicast). No roteamento por difusão a camada de rede fornece um serviço de entrega de pacote enviado de um nó de origem a todos os outros nós da rede ao seu alcance. O roteamento para um grupo habilita um único nó de origem a enviar a cópia de um pacote a um subconjunto de nós das outras redes (KUROSE; ROSS,

2013).

De acordo com Zhao et al. (2009), Flooding (Inundação) é um protocolo de roteamento fácil, no qual cada nó só precisa transmitir a mensagem recém-recebida para seu nós vizinhos, não exigindo registros de topologia ou grandes quantidades de análises. Este protocolo de roteamento é confiável. Independentemente de quaisquer alterações na rede, dados podem ser transmitidos para o nó destino na forma de inundação, desde que exista o caminho entre o nó de origem e o nó de destino. Mas a inundação consome muita energia, porque para cada pacote todos os nós no domínio de broadcast receberão e encaminharão para seus vizinhos. A grande quantidade de energia requerida por esse algoritmo causa uma vida útil menor da rede.

Além disso, esse algoritmo possui uma falha grave: se o grafo, que representa os dispositivos, tiver ciclos, então uma ou mais cópias de cada pacote de difusão permanecerão em ciclo indefinidamente, inundando assim toda a rede. Isso acabaria resultando na criação de uma quantidade tão grande de pacotes de difusão que a rede ficaria inutilizada (KUROSE; ROSS, 2013).

Para resolver essa situação, é necessário uma “inundação controlada”. Os nós remetentes colocam uma estrutura de dados utilizando o: identificador exclusivo do pacote (ID do pacote), o identificador do nó (ID do nó) e o horário recebido daquele pacote. Assim cada nó mantém uma lista com essa estrutura de cada pacote que já recebeu. Deste modo, quando um nó recebe um pacote, primeiro verifica se o pacote está nessa lista. Se estiver, é descartado; se não estiver, é duplicado e repassado para todos os vizinhos do nó, exceto para o nó de quem o pacote acabou de ser recebido (KUROSE; ROSS, 2013).

Outra solução simples é utilizar a “inundação de escopo limitado”. Um campo de tempo de vida (TTL - *Time To Live*) é usado para limitar o número de saltos que a mensagem será repassada. Quando um nó recebe a mensagem, ele decrementa o campo de TTL antes de repassá-la. Assim, uma mensagem inundada alcançará apenas dispositivos que estão dentro de um número estabelecido (TTL inicial) de saltos (KUROSE; ROSS, 2013).

2.4.2 Rede LoRa Mesh

Em cenários de aplicação agrícola em áreas rurais e diversas aplicações urbanas, a comunicação entre os gateways e o servidor de rede nem sempre pode ser garantida, pelo fato de ser difícil fornecer cobertura para grandes áreas. Como visto na seção 2.3, os gateways LoRa atuam como pontes transparentes de dispositivos finais (End-Nodes) para um servidor de rede. Os End-Nodes usam salto único para comunicação com o(s) gateway(s), que então se conecta ao servidor de rede via conexões IP, por exemplo Ethernet, 4G, ou Wi-Fi (LUNDELL et al., 2018). Um grande problema de colocar vários gateways

se dá pelo alto custo dos mesmos, o que se torna inviável para os clientes.

Diferentemente da topologia estrela, em uma rede LoRa mesh, cada nó encaminha informações de outros nós para aumentar o espectro de comunicação e o tamanho da área da rede. Embora isso aumente a largura de banda e a complexidade, reduz a capacidade da rede, além de um ponto crucial: A vida útil da bateria é reduzida quando os nós recebem e encaminham informações de outros nós que podem não ser relevantes para eles, o que é um ponto negativo para o uso dessa topologia de rede em dispositivos LoRa que tem como um de seus objetivos um baixo consumo de energia.

2.4.3 Meshtastic

O Meshtastic® (2022) é um projeto de Software de código aberto que permite usar dispositivos LoRa de baixo custo para fazer uma conexão criptografada como um comunicador fora da rede de longo alcance para áreas sem serviço celular confiável. Esses dispositivos retransmitem as mensagens que recebem para criar uma rede mesh, onde cada membro da malha pode enviar e visualizar mensagens de texto e habilitar alguns recursos opcionais, além de fazer configurações em cada dispositivo. O Meshtastic usa LoRa para comunicações de longo alcance e, dependendo das configurações usadas, o tamanho máximo do grupo teórico varia de 30 a 200 dispositivos.

Os dispositivos criam automaticamente uma malha para encaminhar pacotes conforme necessário, para que todos no grupo possam receber mensagens até mesmo do membro mais distante. Essa mensagem é então transmitida pelo nó três vezes em um determinado intervalo para criar redundância para pacotes perdidos. Quando um nó receptor captura um pacote, ele verifica se já ouviu essa mensagem antes. Se tiver, ignora a mensagem. Se não tiver ouvido a mensagem, ele a retransmitirá em um determinado intervalo três vezes. Para cada mensagem que um nó retransmite, ele decrementa o “limite de salto” em um, evitando assim que uma mensagem fique na malha em uma repetição infinita. Portanto, quando um nó recebe um pacote com um limite de salto (hop limit) igual a zero, ele não irá retransmitir a mensagem.

Atualmente, o software possui suporte a alguns tipos de Hardwares que usam os microcontroladores ESP32 e nRF52. Ambos oferecem baixo consumo de energia e longa duração da bateria. Para o caso do ESP32, que é compatível com Wi-Fi, possui interface Web para auxiliar com as configurações dos dispositivos e também a visualização dos dados. Além desta interface Web, possui o Meshtastic CLI, uma ferramenta que através da linha de comando do terminal auxilia nas configurações de qualquer um dos parâmetros dos dispositivos.

2.5 Dispositivos LoRa Heltec Wireless Stick

Os dispositivos LoRa destinados ao projeto são três módulos ESP32 Heltec Wireless Stick (Figura 10) que integram três formas distintas de comunicação: Wifi, Bluetooth e a rede de comunicação de longo alcance LoRa. Possui um microprocessador Xtensa 32-Bits LX6 que apresenta baixo consumo energético e pequena dimensão, também conta com dois núcleos de CPU (processador dual core), permitindo aplicação em conjunto com sensores, módulos e diferentes interfaces.

Entre as características únicas do ESP32 Heltec Wireless Stick encontra-se uma tela funcional OLED de 0,49” facilitando assim ao desenvolvedor observar informações em tempo real, além de contar com um controlador de carga e conector JST na parte inferior para uma bateria (Li-Ion ou Li-Po) de 3.7V e até 1000mAh para alimentação remota, sem a necessidade de fios.

Portanto, é uma placa de desenvolvimento ideal para desenvolver diversas aplicações de automação residencial, industrial, rural, sistemas de localização, infraestrutura de serviços, controle remoto de sensores e atuadores. Ela foi desenvolvida pela empresa Heltec Automation. (Heltec Automation, 2022).

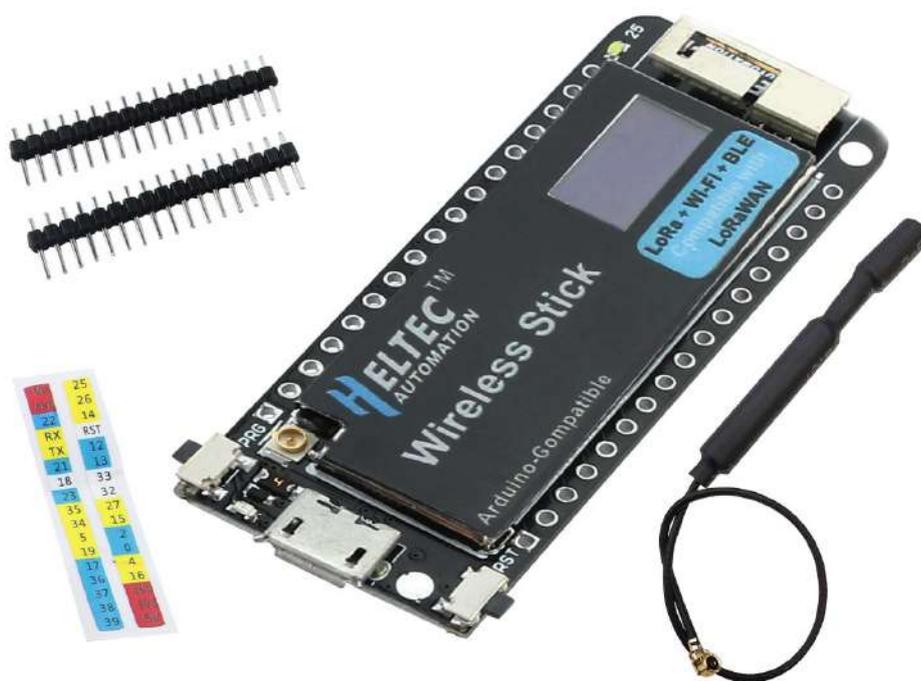


Figura 10 – LoRa Heltec Wireless Stick ESP32

Fonte: https://heltec-automation-docs.readthedocs.io/en/latest/esp32/wireless_stick/index.html

Tabela 2 – Descrição do dispositivo LoRa

Modelo	Heltec Wireless Stick
Chip Base	ESP32-D0WDQ6
Fonte de alimentação USB	5V DC
Fonte Bateria de lítio	3,7V a 4,2V
Processador	Xtensa 32-Bit LX6 Dual Core
Clock	80 a 240 MHz
Memória ROM	448KB
Memória RAM	520KB SRAM
Memória Flash Externa	32-Bit de acesso e 4Mb
WiFi 802.11 b/g/n	2,4 a 2,5 GHz
Criptografia WiFi	AES / RSA / ECC / SHA
Bluetooth	4.2 BR / EDR e BLE
Temperatura de trabalho	-40° à +80° C
Chip Base LoRa	SX1276
Sensibilidade	-139dBm
Potência Máxima	19dB ±1dB
Modo de modulação	LoRa, FSK, GFSK e OOK
Taxa de transferência	0,018 à 37,5 Kbps (LoRa)
Dimensões (CxLxE):	63x27x11mm
Peso	8,3g

2.6 Tecnologias Utilizadas

Como o projeto objetiva conhecimentos tanto de Hardware quanto de Software, foi necessária a utilização de alguns recursos para o desenvolvimento. O Sistema Operacional utilizado no computador é o Ubuntu 18.04.6 LTS. Na Tabela 3 são listadas as ferramentas utilizadas no desenvolvimento do trabalho, bem como o propósito de sua utilização.

Tabela 3 – Ferramentas Utilizadas.

Tecnologia	Versão	Descrição	Propósito
C++	7.5.0	Linguagem de programação compilada multi-paradigma e de uso geral.	Escrita do código-fonte das classes que compõem o sistema. Muito usada para o desenvolvimento de sistemas embarcados, IoT, etc.
C	-	Linguagem de programação compilada de propósito geral, estruturada, imperativa, procedural.	Escrita do código-fonte das classes que compõem o sistema. Muito usada para o desenvolvimento de sistemas embarcados, IoT, etc
Python3	3.6.9	Linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.	Pré requisito para instalação do meshtastic.

Tecnologia	Versão	Descrição	Propósito
Meshtastic Firmware	1.2.64	Uma classe específica de software que fornece controle de baixo nível para o hardware específico do dispositivo.	Inicializar as rotinas e preservar o funcionamento do dispositivo LoRa.
Pip	21.3.1	Sistema de gerenciamento de pacotes padrão de facto usado para instalar e gerenciar pacotes de software escritos em Python.	Pré requisito para instalação do Meshtastic Cli.
Esptool	3.3.1	Um utilitário baseado em Python, de código aberto, para se comunicar com o carregador de inicialização ROM nos chips das séries Espressif ESP8266 e ESP32.	Pré requisito para o projeto que possui dispositivos Esp32, visto na seção 2.5.

Na Tabela 4 nota-se os softwares que apoiaram o desenvolvimento da documentação e também do código fonte.

Tabela 4 – Softwares de Apoio ao Desenvolvimento do Projeto

Tecnologia	Versão	Descrição	Propósito
Visual Studio Code	1.62.2	Ambiente de desenvolvimento (IDE) com suporte às linguagens utilizadas.	Desenvolvimento, Build, Upload e testes.
PlatformIO	2.4.3	Ambiente de desenvolvimento profissional para IoT, Arduino, CMSIS, ESP-IDF, FreeRTOS, libOpenCM3, mbed OS, Pulp OS, etc.	Construtor de código multiplataforma sem dependências externas; Mais de 1000 placas incorporadas, mais de 40 plataformas de desenvolvimento, mais de 20 estruturas; Depuração; Teste de unidade; Análise de código estático; Desenvolvimento Remoto; Conclusão de código inteligente C/C++; Gerenciador de bibliotecas para milhares de bibliotecas populares;
GIT	2.17.1	Sistema de controle de versões distribuído, pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo	Código será hospedado em uma plataforma digital baseada em nuvem com um sistema de controle de versões, para possíveis contribuições, já que será um projeto incremental.
Meshtastic CLI	1.2.95	A CLI (Commandline Interface) é uma ferramenta que através da linha de comando do terminal auxilia nas configurações de qualquer um dos parâmetros dos dispositivos	Auxiliar e definir as preferências de usuário para o dispositivo.

Tecnologia	Versão	Descrição	Propósito
Meshtastic Web	-	Meshtastic Web é um PWA (Progressive Web App) que roda diretamente no seu navegador.	Auxiliar com as configurações dos dispositivos, obter dados de testes também a visualização dos dados.
Meshtastic Android	1.2.64	É um aplicativo Android mobile que lida com a comunicação entre os dispositivos da rede mesh.	Auxiliar com a localização e mensagens de texto enviadas ao seu canal.
OverLeaf	2022	Editor de LaTeX.	Escrita da documentação do sistema, sendo usado o template abn-TeX. < http://www.abntex.net.br >.

Além dos recursos de Software, foram utilizados três dispositivos Heltec Wireless Stick Esp32 (seção 2.5), protoboards, antenas, cabos USB para transmissão de dados e carregadores portáteis.

3 Implantação de uma rede LoRa Mesh para a Universidade Federal do Espírito Santo

Nesta parte do trabalho são apresentadas as formas em que o projeto foi desenvolvido, as etapas das atividades realizadas com os Hardwares (Seção 2.5) e as tecnologias utilizadas (Seção 2.6). O desenvolvimento foi dividido em quatro fases principais: configuração do ambiente de desenvolvimento (Seção 3.1), desenvolvimento (Seção 3.2), configuração dos nós na rede (Seção 3.3) e por fim o funcionamento da rede LoRa mesh (Seção 3.4).

Esse trabalho consiste no desenvolvimento, implantação e validação experimental de uma rede mesh através de dispositivos IoT. Como padrão para comunicação de dados, escolheu-se a tecnologia LoRa, que é uma tecnologia de rádios de longa distância e uma das mais utilizadas atualmente em aplicações de cidades inteligentes. Além disso, com a tecnologia LoRa, poucos nós abrangem uma área extensa, diminuindo assim o custo para cobrir uma grande área.

3.1 Ambiente de Desenvolvimento

Como mostrado na seção 2.6, o sistema operacional Linux Ubuntu 18.04.6 LTS foi utilizado, porque possui características que facilitam para o desenvolvedor, além de muitas ferramentas que já vêm instaladas ou possuem um gerenciamento de pacotes que ajudam na instalação das mesmas por linha de comando.

3.1.1 Preparação do ambiente

Para que não houvessem problemas no desenvolvimento foram instalados alguns softwares, seguindo os passos:

1. Em um terminal aberto;
2. Instalar o g++ que é o compilador C++:
`apt-get install g++`
3. Instalar o gcc que é o compilador C:
`apt-get install gcc`
4. Instalar o gerenciador de versão GIT:
`apt-get install git`

5. Instalar o Python3 (versão necessária > 3.6):

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
sudo apt-get install python3.6, ou
sudo apt-get install python3.X, onde X = {7,8,9,10}
```

6. Instalar o pip:

```
sudo apt-get install python3-pip
```

7. Instalar o esptool:

```
pip3 install --upgrade esptool
```

Ao instalar o esptool e conectar o dispositivo LoRa Heltec Wireless Stick com um cabo USB, usou-se o seguinte comando: `esptool chip_id` para assim conferir o(s) dispositivo(s) conectado(s):

```
esptool.py v3.3.1
Found 1 serial ports
Serial port /dev/ttyUSB0
Connecting...
Detecting chip type... ESP32
Chip is ESP32-D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 84:cc:a8:4a:e0:1c
Uploading stub...
Running stub...
Stub running...
Warning: ESP32 has no Chip ID. Reading MAC instead.
MAC: 84:cc:a8:4a:e0:1c
Hard resetting via RTS pin...
```

Além destes passos, foram instalados o Visual Studio Code (MICROSOFT®, 2022), o PlatformIO para VsCode (PLATFORMIO, 2022) e o navegador Google Chrome (GOOGLE, 2022), onde foi possível obter os dados dos testes pelo Meshtastic Web.

Posteriormente, foi necessário instalar o meshtastic CLI, que é parte do projeto Meshtastic-python (2022) e é atualmente a melhor maneira de gerenciar suas configurações. É rico em recursos e bem documentado. As páginas de configurações têm exemplos para cada preferência de usuário para o dispositivo. Para a instalação, possui diferentes formas, uma é baixando o executável disponível em Releases <<https://github.com/meshtastic/Meshtastic-python/releases>>. A versão utilizada foi a 1.2.95, que se mostrou estável.

Após fazer o download, utilizou-se os comandos:

```
chmod +x meshtastic_ubuntu && mv meshtastic_ubuntu meshtastic
./meshtastic
```

A outra forma é baixando pelo pip (Em alguns casos substituir o “pip” por “pip3”):

```
pip install meshtastic
pip install --upgrade meshtastic
```

Feito isso, começou-se enfim o desenvolvimento.

3.2 Desenvolvimento

Com todas as tecnologias de Software e Hardware ajustadas, foi pesquisado os hardwares suportados pelo projeto Meshtastic e como não havia a placa Heltec Wireless Stick, foi adicionado esta como uma nova variante de placa Esp32. Para isso foi essencial a criação de um ambiente de compilação.

3.2.1 Ambiente de Compilação

Para o ambiente de compilação, foi utilizado o PlatformIO e o Visual Studio Code, foi clonado o projeto meshtastic-device, mas por motivos de estabilidade foi necessário utilizar a Release 1.2.64 <<https://github.com/meshtastic/Meshtastic-device/releases>> e na pasta usar o comando: (para este comando é necessário o Python3 e o Git instalados) `git submodule update --init`

3.2.2 Adicionando o próprio Hardware

Como o meshtastic é um projeto incremental, foi possível adicionar um modelo específico ao projeto. Para isso foi necessário conhecer o diagrama de Pinagem da placa, mostrado na Figura 11.

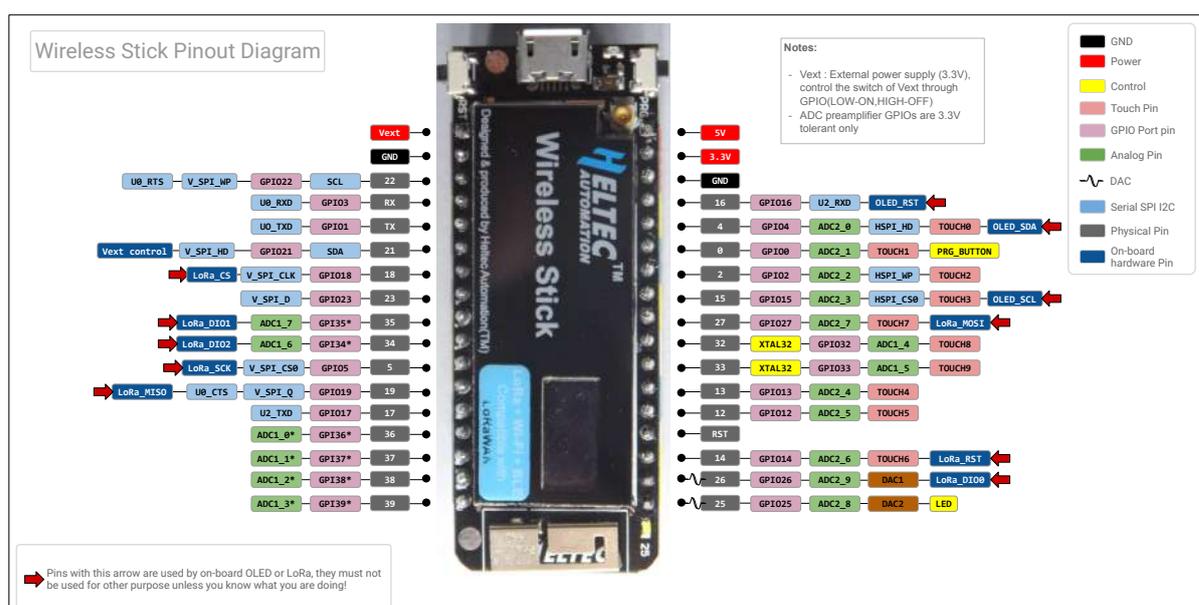


Figura 11 – Pinout Diagram LoRa Heltec Wireless Stick ESP32
 Fonte: https://resource.heltec.cn/download/Wireless_Stick/Wireless_Stick.pdf

Em seguida dirigiu-se ao ambiente de compilação e na pasta “variants” foi criado um novo diretório com o nome da nova placa e criado dois arquivos fundamentais: platformIO.ini e variant.h que serão explicados a seguir. (Figura 12).

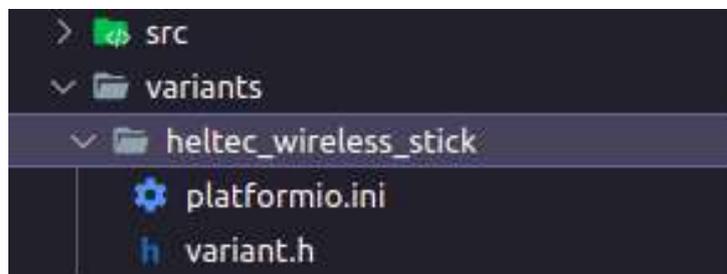


Figura 12 – Criação de diretório e arquivos para a placa Heltec Wireless Stick
Fonte: Autor, 2022.

3.2.2.1 platformio.ini

Cada projeto PlatformIO deve ter um arquivo de configuração denominado “platformio.ini”. O platformio.ini possui seções (cada uma indicada por um [cabeçalho]) e pares de chave/valor dentro das seções. (Project Configuration File platformio.ini (2022)).

Uma seção com um prefixo “env:” define um ambiente de trabalho para pio run, pio test, pio check, pio debug e outros comandos. Vários ambientes [env:NOME] com NOME diferentes são permitidos. Além disso, todo projeto deve definir pelo menos um ambiente de trabalho que deve ter um NOME exclusivo. Como o meshtastic oferece suporte para algumas placas, possui diferentes ambientes de trabalho. (Project Configuration File Section: Working (2022)).

O PlatformIO tem configurações pré-configuradas para as placas mais populares: configuração de compilação, configuração de upload, configuração de depuração, informações de conectividade, etc. Então foi pesquisado através do catálogo de placas a Heltec Wireless Stick. (Boards » Heltec Wireless Stick (2022)). Este site contém características do hardware, configurações, Uploading, Debugging, Frameworks.

Através do Código A.1 disponível no Apêndice A, pôde-se notar que foi criado um novo ambiente com NOME = heltec_wireless_stick. As linhas que começam com ; são comentadas, e nota-se também que não se utilizou a “plataform = espressif32” como na documentação do platformIO indica, porque este não representa o ambiente raiz, que é o esp32_base e está localizado no arquivo platformio.ini do diretório raiz do projeto.

Também foi necessário no arquivo “platformio.ini” usar uma definição canônica da variante de hardware para PRIVATE_HW e fazer o -I no build_flags apontar para o diretório recém-criado.

3.2.2.2 variant.h

Um arquivo salvo com extensão de arquivo h é um arquivo de cabeçalho usado em arquivos C/C++ para incluir a declaração de variáveis, constantes e funções. O arquivo “variant.h” possui as declarações dos pinos que serão usados como constantes no projeto meshtastic. As constantes tem duas vantagens principais:

- Facilitam a modificação do programa
- Tornam o programa mais legível.

Para a definição dos pinos corretamente, foi feito um estudo profundo nos códigos do projeto e como eles foram utilizados em outras placas. A partir daí foram definidos corretamente no arquivo variant.h, disponível no Apêndice A - Código A.2.

3.2.3 Testando o próprio Hardware

Elaborada a parte descrita nas seções 3.2.2.1 e 3.2.2.2, testou-se o hardware. Foi necessário fazer um build do projeto, seguindo os seguintes passos:

1. Com o Visual Studio aberto, foi acessado o arquivo platformio.ini no diretório raiz e adicionada a nova placa:

Na seção [platformio], “default_envs” permite definir quais ambientes devem ser processados por padrão. Além disso, na depuração verifica-se essa opção ao procurar o ambiente de depuração. Esta opção também pode ser configurada pela variável de ambiente global PLATFORMIO_DEFAULT_ENVS.

No Código A.3 do Apêndice A, pôde-se notar que as demais placas suportadas pelo meshtastic estão comentadas e foi inserido na linha 5 a nova placa Heltec Wireless Stick.

2. Pressionou-se “Ctrl + Shift + P” e selecionou-se a opção “platformio: Switch Project Environment”;
3. Dentre as opções que surgem, escolheu-se a opção Default, já que foi colocada a nova placa como padrão no passo 2;
4. Para compilar o firmware, bastou executar PlatformIO (“Ctrl + Shift + P”) e selecionar a opção “Build”;

Com isso começou um processo de compilação/instalação das bibliotecas e dependências, que demorou alguns segundos. O resultado que se obteve foi similar à Figura 13, que contém alguns dados sobre a memória utilizada pelo firmware, e a sua versão.

```

Using meshtastic platform-custom.py, firmware version 1.2.64
Retrieving maximum program size .pio/build/heltec_wireless_stick/firmware.elf
Checking size .pio/build/heltec_wireless_stick/firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [==          ] 24.1% (used 79008 bytes from 327680 bytes)
Flash: [=====   ] 91.8% (used 1685103 bytes from 1835008 bytes)
===== [SUCCESS] Took 25.45 seconds =====

Environment      Status      Duration
-----
heltec_wireless_stick SUCCESS    00:00:25.452
===== 1 succeeded in 00:00:25.452 =====

0 terminal será reutilizado por tarefas, pressione qualquer tecla para fechá-lo.
    
```

Figura 13 – Resultado esperado do Build da placa Heltec Wireless Stick
 Fonte: Autor, 2022.

6. Por fim, atualizar o firmware para o novo dispositivo basta executar o PlatformIO: e selecionar a opção “Upload”.

Com um cabo USB conectado ao dispositivo, com o upload foi possível utilizar as funções do meshtastic. O resultado que se obteve foi similar à Figura 14.

```

Wrote 1685344 bytes (987936 compressed) at 0x00010000 in 16.3 seconds (effective 825.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 46.34 seconds =====

Environment      Status      Duration
-----
heltec_wireless_stick SUCCESS    00:00:46.341
===== 1 succeeded in 00:00:46.341 =====

0 terminal será reutilizado por tarefas, pressione qualquer tecla para fechá-lo.
    
```

Figura 14 – Resultado esperado do Upload da placa Heltec Wireless Stick
 Fonte: Autor, 2022.

Todas essas configurações acima são para o desenvolvedor fazer o upload para a placa Heltec. O desenvolvedor também pode criar uma nova release e disponibilizar o Firmware para outras pessoas instalarem em seus dispositivos Heltec Wireless Stick, sendo possível instalar através do esptool com os comandos:

`./device-install.sh -f firmware-BOARD-VERSION.bin`

E atualizar:

`./device-update.sh -f firmware-BOARD-VERSION.bin`

E caso não consiga, baixar o ESPHome-Flasher (ESPHome (2022)), e utilizar para fazer o upload do Firmware disponibilizado.

3.3 Configurando os Nós na rede mesh

Após todo o desenvolvimento visto na seção 3.2, foi utilizado o meshtastic Cli para algumas configurações iniciais essenciais para o funcionamento dos nós. Com o cabo USB conectado, utilizou-se o comando:

```
meshtastic --info
```

E se obteve a seguinte configuração inicial, disponível no Apêndice B - Registro B.1 e na Figura 15.

```

Registro de Saída B.1 – Dispositivo com configurações padrões
1 $ meshtastic --info
2 Connected to radio
3
4 Owner: Unknown e01c (?1C)
5 My info: { 'myNodeNum': 2823479324, 'numBands': 13, 'firmwareVersion': '1.2.64',
  'rebootCount': 251, 'bitrate': 17.08847, 'messageTimeoutMsec': 300000, '
  minAppVersion': 20200, 'maxChannels': 8, 'hasWifi': true, 'channelUtilization
  ': 7.7133336, 'airUtilTx': 0.7772222 }
6 Nodes in mesh: { 'num': 2823479324, 'user': { 'id': '!a84ae01c', 'longName': '
  Unknown e01c', 'shortName': '?1C', 'macaddr': '84:cc:a8:4a:e0:1c', 'hwModel':
  'PRIVATE_HW'}, 'position': {} }
7
8 Preferences: { 'phoneTimeoutSecs': 900, 'lsSecs': 300 }
9
10 Channels:
11 PRIMARY psk=default { 'modemConfig': 'Bw125Cr48Sf4096', 'psk': 'AQ==' }
12
13 Primary channel URL: https://www.meshtastic.org/d/#CgUYAyIBAQ

```

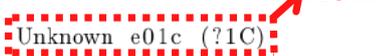





Figura 15 – Configuração padrão do dispositivo

Fonte: Autor, 2022.

Pode-se observar que o dispositivo vem com algumas configurações padrões essenciais que precisam ser alteradas, tais quais: Nome, Região, URL do canal e configurações do canal. A seguir estão os passos importantes que o desenvolvedor deve tomar antes de transmitir qualquer mensagem pela rede.

3.3.1 Definir o nome do Nó

Definir o nome do nó é um passo de suma importância para a identificação do dispositivo na malha por parte do usuário. Pois cada dispositivo recebe um ID exclusivo, que são os 8 últimos dígitos do endereço MAC, acrescentado de ! ao início. Além disso, o nó também recebe um número (`myNodeNum`) que é um número que o identifica na rede. Então, por parte dos próprios Hardwares, o nome não é utilizado, mas para visualização do usuário é essencial. Portanto, com esse comando pôde-se dar nome ao Nó:

```
meshtastic --dest '!a84ae01c' --set-owner 'mesh 3'
```

O `--dest` indica o ID do nó que deve ser definido e com o `--set-owner` o novo nome.

3.3.2 Definir a Região do Nó

A variável `Region` define em qual região seu rádio está configurado para funcionar. É importante garantir que foi configurada a região correta para não transmitir em uma frequência incorreta. Como viu-se na seção 2.3.2, a frequência permitida no Brasil está entre 902 MHz a 907,5 MHz e 915 MHz a 928 MHz, mesma utilizada na Austrália. O `meshtastic` possui algumas regiões definidas, como mostrado na Tabela 5. A que foi utilizada é a ANZ (Austrália e Nova Zelândia) que é a mesma utilizada no Brasil. Se não for definido, o padrão será as configurações `Unset`, equivalente aos US.

Tabela 5 – Regiões Definidas.

Nome	Frequência	Espaçamento	Nº de Canais
US	903.08	2.16	13
EU433	433.175	0.2	8
EU868	865.2	0.3	10
CN	470.0	2.0	20
JP	920.0	0.5	10
ANZ	916.0	0.5	20
KR	921.9	0.2	8
TW	923.0	0.2	10
RU	868.9	0.2	2
Unset	903.08	2.16	13

Portanto, para selecionar a região correta utilizou-se o seguinte comando no `meshtastic` CLI:

```
meshtastic --set region ANZ
```

3.3.3 Definir o Canal do Nó

Um canal é definido por uma URL, no qual todos os dispositivos que contém a URL do canal podem se comunicar pela rede mesh, além disso um dispositivo pode possuir mais que um canal, ou seja, participar de várias redes mesh distintas. E caso tenha configurações do canal incompatíveis, os nós não poderão se comunicar uns com os outros.

A forma de funcionamento é que cada nó mantém uma lista de canais que conhece. Um desses canais é rotulado como o canal “primário”. Esse é o único canal usado para definir os parâmetros de rádio, nele é controlado o fator de espalhamento, taxa de codificação,

largura de banda e também é selecionada a frequência específica que todos os membros desta malha estão “falando”.

Este canal pode ou não ter um PSK (criptografia). O PSK padrão é tecnicamente criptografado, mas a chave está incluída no código-fonte do GitHub e deve-se assumir que qualquer pessoa maliciosa a teria e usando o meshtastic conseguiria decifrar.

Diante disso, deve-se definir um canal de comunicação exclusivo para a nossa Rede Mesh. A princípio, a rede mesh começa com um canal padrão chamado LongSlow-V (este nome tem a ver com o alcance e velocidade de propagação), mas deve-se criar um para que fique com maior segurança.

3.3.3.1 Criação do Canal

Para a criação do canal, tem-se que conhecer alguns parâmetros. Cada canal possui um parâmetro chamado de ID, que é um número inteiro. E por padrão o canal tem ID = 0, Mas é possível mudar utilizando o comando:

```
meshtastic --ch-set id 1234 --ch-index 0
```

Outro parâmetro de grande importância em um canal é o `modem_config` que define o alcance e a velocidade. Na Tabela 6 nota-se alguns valores dos mais utilizados e que funcionam bem.

Tabela 6 – Configurações `modem_config`.

Contexto	Nome (alcance/velocidade)	Largura de Banda	Taxa de Codificação	Fator de espalhamento	Taxa
Bw500Cr45Sf128	Short/Fast	500 kHz	4/5	7 = 128chips/symbol	18.89 kbps
Bw125Cr45Sf128	Short/Slow	125 kHz	4/5	7 = 128chips/symbol	4.69 kbps
Bw250Cr47Sf1024	Medium/Fast	250 kHz	4/7	10 = 1024chips/symbol	1.2 kbps
Bw250Cr46Sf2048	Medium/Slow	250 kHz	4/6	11 = 2048chips/symbol	0.75 kbps
Bw31_25Cr48Sf512	Long/Fast	31.25 kHz	4/8	9 = 512chips/symbol	0.19 kbps
Bw125Cr48Sf4096	Long/Slow	125 kHz	4/8	12 = 4096chips/symbol	0.13 kbps

Para definir algum desses parâmetros no canal, deve-se indicar o contexto e o ID do canal: (devido ao objetivo do trabalho ser o mapeamento da rede na UFES, foi escolhido a configuração com característica de longo alcance, baixa velocidade e baixa taxa de bits/s).

```
meshtastic --ch-set modem_config Bw125Cr48Sf4096 --ch-index 0
```

Uma observação importante é que cada um desses parâmetros do `modem_config` pode ser definido, por exemplo, a Largura de Banda [1], Taxa de Codificação[2], Fator de Espalhamento[3] e a Potência de transmissão configurada do rádio LoRa[4]:

```
meshtastic --ch-set bandwidth 125 --ch-index 0 [1]
meshtastic --ch-set coding_rate 8 --ch-index 0 [2]
meshtastic --ch-set spread_factor 12 --ch-index 0 [3]
meshtastic --ch-set tx_power 0 --ch-index 0 [4]
```

Outro parâmetro importante para os usuários é o NOME do Canal, que utilizou-se o seguinte comando:

```
meshtastic --ch-set name "UFES" --ch-index 0
```

O canal criado para o projeto possui o nome “UFES”. Um ponto a destacar é que a cada mudança em alguma configuração do canal, a URL também muda, o que torna um pouco mais difícil a manutenção da rede mesh, por isso o ideal foi definir uma boa configuração inicial.

Além dessas configurações, possui a criptografia que pode ser definida como:

- none: Desabilita a criptografia;
- default: Usa uma chave fraca de criptografia;
- random: Gera uma chave de criptografia segura de 256 bits. Usada para canais secundários;
- simple0 / simple254: Utiliza uma codificação de byte único para criptografia.

E por fim, o canal também deve ter definido a região, da mesma forma que o nó. O comando para a definição é:

```
meshtastic --set region ANZ
```

Durante todas essas configurações, foi utilizado para os testes o ID = 0, a criptografia padrão, o `modem_config = Bw125Cr48Sf4096`, a região = ANZ e o Nome = UFES. Essas configurações geraram uma URL para o canal, que foi carregado pelo cabo USB em cada dispositivo através do comando:

```
meshtastic --seturl https://www.meshtastic.org/d/#CgsYAyIBASoEVUZFUw
```

O resultado de todas essas configurações realizadas nas seções 3.3.1, 3.3.2, 3.3.3 é mostrado no Apêndice B - Registro B.2.

Ao chegar nesse ponto do trabalho, pode-se notar que já estão definidas as configurações essenciais do dispositivo Heltec Wireless Stick para com a rede LoRa mesh. Os próximos passos foram configurar os outros dois dispositivos (foram utilizados três para os testes).

3.4 Funcionamento da rede LoRa mesh

Mediante as configurações vistas na seção 3.3, utilizou-se três placas Heltec Wireless Stick e foi realizado o mesmo procedimento para cada uma delas. Também foi utilizado um display Oled para o acompanhamento das mensagens pelo canal UFES. A Figura 16 mostra o protótipo montado com os dispositivos.

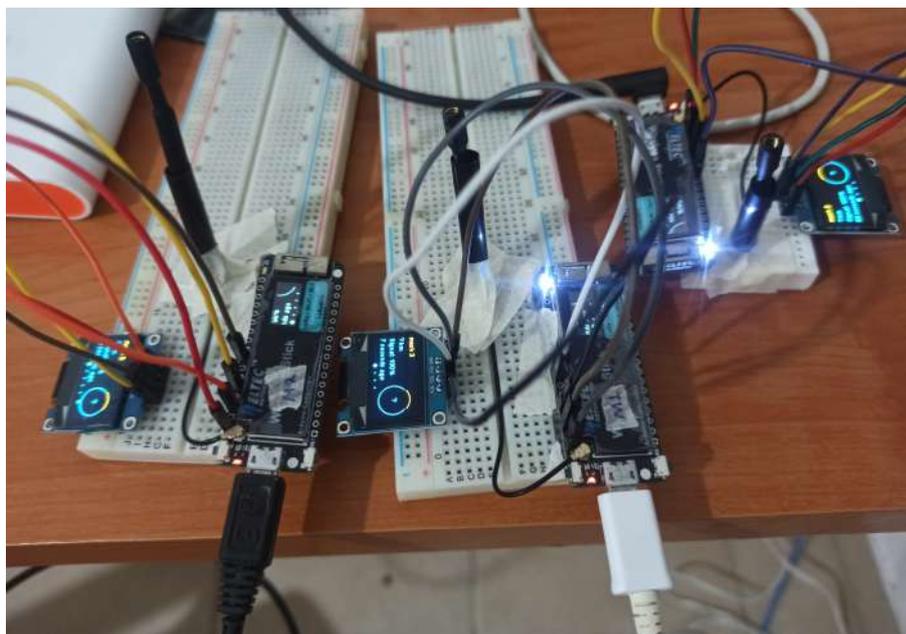


Figura 16 – Dispositivos configurados na rede LoRa mesh UFES
Fonte: Autor, 2022.

Com os dispositivos ligados, esperou-se um tempo curto até que a rede fosse formada, e o registro de saída está disponível no Apêndice B - Registro B.3. Também é mostrado na Figura 17.

Diante da Figura 17, pode-se observar, neste momento, que “Nodes in Mesh” possui os dispositivos Mesh 1, Mesh 2 e Mesh 3 e algumas informações extras sobre cada um deles na rede, mostrando assim o funcionamento da rede LoRa mesh. Além de outras mudanças em *Preferences* e na característica do canal, que gerou uma URL que foi configurada em todos os dispositivos para que comunicassem entre si.

Registro de Saída B.3 – Rede LoRa mesh UFES configurada

```

1 $ meshtastic --port /dev/ttyUSB0 --info
2 Connected to radio
3
4 Owner: mesh 3 (m3) Nome
5 My info: { 'myNodeNum': 2823479324, 'numBands': 20, 'firmwareVersion': '1.2.64',
  'rebootCount': 254, 'bitrate': 17.08847, 'messageTimeoutMsec': 300000, '
  minAppVersion': 20200, 'maxChannels': 8, 'hasWifi': true, 'airUtilTx': Nós
  0.190666666 }
6 Nodes in mesh: { 'num': 2823479324, 'user': { 'id': '!a84ae01c', 'longName': 'mesh
  3', 'shortName': 'm3', 'macaddr': '84:cc:a8:4a:e0:1c', 'hwModel': 'PRIVATE_HW'
  }, 'position': {} } { 'num': 2823479488, 'user': { 'id': '!a84ae0c0', 'longName
  ': 'mesh 2', 'shortName': 'm2', 'macaddr': '84:cc:a8:4a:e0:c0', 'hwModel': '
  PRIVATE_HW' }, 'position': { 'latitudeI': -202736021, 'longitudeI': -403060778,
  'latitude': -20.273602099999998, 'longitude': -40.3060778}, 'snr': 11.0, '
  lastReceived': { 'from': 2823479488, 'to': 2823479324, 'id': 658023066, 'rxSnr
  ': 11.0, 'hopLimit': 3, 'fromId': '!a84ae0c0', 'toId': '!a84ae01c' }, '
  lastHeard': None, 'hopLimit': 3 } { 'num': 2823478980, 'user': { 'id': '!
  a84adec4', 'longName': 'mesh 1', 'shortName': 'm1', 'macaddr': '84:cc:a8:4a:de
  :c4', 'hwModel': 'PRIVATE_HW' }, 'position': {}, 'snr': 14.5, 'lastReceived':
  { 'from': 2823478980, 'to': 2823479324, 'id': 1087849284, 'rxSnr': 14.5, '
  hopLimit': 3, 'fromId': '!a84adec4', 'toId': '!a84ae01c' }, 'lastHeard': None,
  'hopLimit': 3 }
7
8 Preferences: { 'phoneTimeoutSecs': 900, 'lsSecs': 300, 'region': 'ANZ' Região
9
10 Channels:
11 PRIMARY_psk=default { 'modemConfig': 'Bw125Cr48Sf4096', 'psk': 'AQ==', 'name':
  'UFES' Canal
12
13 Primary channel URL: https://www.meshtastic.org/d7#CgsYAvlBASoEVUZFUw URL do canal

```

Figura 17 – Configuração final da rede LoRa mesh UFES

Fonte: Autor, 2022.

Além disso, para firmar o funcionamento e enviar mensagem entre os dispositivos do canal UFES, existem algumas formas diferentes, a mais simples foi pelo meshtastic CLI, com o seguinte comando:

```
meshtastic --sendtext "Hello Canal UFES sou M3"
```

Dessa forma com o auxílio do display OLED, nota-se que todos os dispositivos recebem a mensagem.

3.4.1 Roteamento por Inundação (Flooding)

Contudo ainda não se pode afirmar se a rede mesh está funcionando através do protocolo de roteamento, isso por que o protocolo utilizado funciona no princípio de inundação conceituado na seção 2.4.1.1. Ao ligar um nó, seu número de salto será inicializado para o máximo de saltos (MaxHop) que por padrão é 3, mas podendo ser alterado com o comando: (o valor máximo é 7, por limite de protocolo).

```
meshtastic --set hop_limit 7
```

Em seguida, a cada novo nó da rede que a mensagem transitar, o número de saltos para a mensagem é decrementado em uma unidade e conseqüentemente transmite a mensagem para seus vizinhos até que ela chegue em seu destino final ou número de saltos esteja em zero. Além disso, quando outros nós recebem esta mensagem, eles primeiro verificam em seu histórico se a mensagem já foi recebida recentemente (Essa tabela tem tamanho N e quando chega ao limite vai retirando as mensagens dos pacotes mais antigos). Se sim, então a descarta. Se não, a adiciona à tabela e assim reenvia essa mensagem aos nós vizinhos.

Ao reenviar, utiliza-se o ID do remetente original, e não o ID do nó atual. Além de ao reenviar, define-se um atraso aleatório entre 0 e 10 segundos para diminuir a chance de colisões entre os transmissores e quaisquer entradas com mais do que o tempo máximo que uma inundação pode levar será descartada. Isso é feito para que a mensagem não fique transitando pela rede infinitamente dada a simplicidade do algoritmo. É importante salientar também que os dispositivos enviam um ACK, que indica a confirmação ou recebimento da mensagem para o remetente.

Desse modo, como os dispositivos após as configurações iniciais estão próximos um dos outros e o algoritmo por inundação faz com que um nó envie para todos os seus vizinhos, todas as mensagens alcançam seu destino final com apenas um salto e visualmente não é possível notar o funcionamento da rede. Porém na seção 4 será notado os saltos entre os nós até que a mensagem chegue no destino final.

3.4.2 Aplicativo Android

Diante da instalação dos firmware no dispositivo Heltec Wireless Stick, a comunicação entre os dispositivos estão restritas ao meshtastic CLI, o que dificulta a acessibilidade em testes de campo, que serão apresentados na seção 4. Para isso foi utilizado um aplicativo Android que possibilita enviar mensagens pelo canal UFES.

O princípio de funcionamento deste aplicativo é por Bluetooth no qual os dispositivos são listados e pareados, de acordo com um código PIN que aparece na tela OLED do dispositivo. A contar deste momento o dispositivo é conectado e o aplicativo lida com a comunicação e pode mostrar a localização de todos em seu canal, se estes contiverem módulos GPS ou estiverem pareados com outro dispositivo móvel.

Sendo assim, cada membro da rede pode ver a localização e a distância dos membros e mensagens de texto enviadas ao seu canal. Vale ressaltar que cada dispositivo móvel pode apenas conectar com um único dispositivo.

Para a obtenção do aplicativo, foi necessário baixar o Android Application Pack

(.APK), que é um arquivo de aplicação destinado ao sistema operacional móvel Android com a versão Release 1.2.64 <<https://github.com/meshtastic/Meshtastic-Android/releases>>, que se mostrou compatível com o firmware instalado nos dispositivos.

3.4.3 Interface Web

Como o dispositivo Heltec Wireless Stick possui conexão WiFi, foi utilizada a Interface Web no navegador Google Chrome para os testes de alcances, que serão apresentados na seção 4. Ao se utilizar a interface Web, todo o tráfego é servido pelo protocolo HTTPS (protocolo HTTP sobre uma camada adicional de segurança), pois os nós geram certificados autoassinados.

Para a utilização Web, primeiramente deve-se configurar para que o dispositivo tenha acesso ao WiFi, que por padrão vem desabilitado. Para habilitar existem duas opções: Se conectar ao WiFi como cliente ou criar um ponto de acesso baseado em software (SoftAP). Os comandos são mostrados a seguir:

```
meshtastic --set wifi_ssid "nome da rede"
meshtastic --set wifi_password "senha da rede"
meshtastic --set wifi_ap_mode true
meshtastic --set wifi_ap_mode false
```

Com esses comandos consegue-se conectar a rede, sendo necessário definir o ssid e o password e podendo escolher se conectar como cliente (`wifi_ap_mode false`) ou ser um ponto de acesso (`wifi_ap_mode true`).

Ao habilitar o WiFi faz com que desative o Bluetooth porque apenas um método de conexão funcionará por vez. A seguir, pode ser feito o acesso do nó diretamente por meio do navegador: `https://ENDEREÇO_IP_DISPOSITIVO/` substituindo o `ENDEREÇO_IP_DISPOSITIVO` pelo endereço IP do nó, que pode ser encontrado na tela OLED do dispositivo Heltec.

3.4.4 Arquitetura de Funcionamento

Diante do desenvolvimento foram utilizadas várias ferramentas como o VsCode e PlataformaIO, que carregando o código disponível no Github <<https://github.com/mathbozzi/rede-LoRa-Mesh-UFES>> nos dispositivos Heltec Wireless Stick, foi possível implantar a rede LoRa Mesh. Foi necessário utilizar o Meshtastic CLI para definir as configurações de rede em cada dispositivo. O Meshtastic Web foi necessário para fazer o download dos dados .csv gerados nos testes de campo que serão vistos adiante na seção 4. O Aplicativo Android foi necessário para os testes de campo, onde via Bluetooth Low Energy (BLE) poderia se conectar ao dispositivo e enviar os dados pelo canal. Esse esquemático é mostrado na

Figura 18.

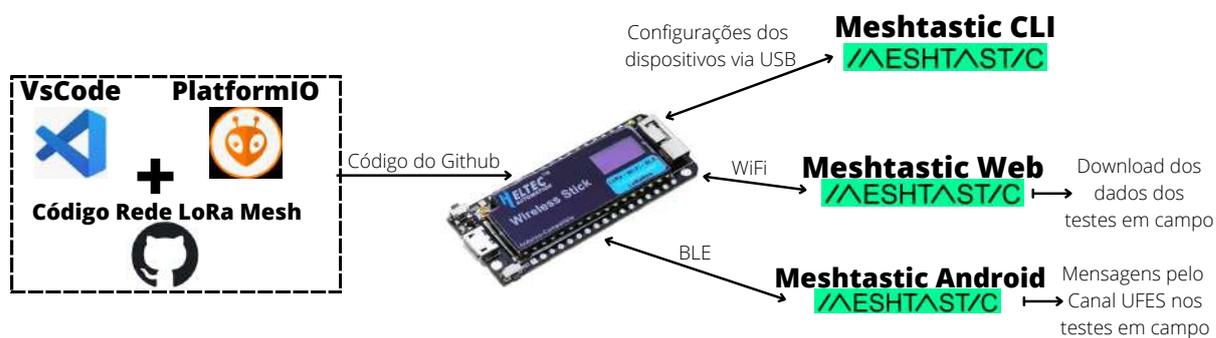


Figura 18 – Esquemático de desenvolvimento, configurações e testes
Fonte: Autor, 2022.

4 Avaliação da Rede LoRa Mesh

4.1 Metodologia

Para realizar uma avaliação sobre o objetivo do trabalho, foi implantada uma rede LoRa mesh na Universidade Federal do Espírito Santo, realizado em área urbana universitária, no bairro de Goiabeiras em Vitória no Espírito Santo.

Para avaliar a cobertura da rede LoRa mesh, foram feitos testes com dois (seção 4.2) e três (seção 4.3) nós, sendo feitas as configurações de cada um deles em ambos os testes. Também foram mensurados a SNR em diversos pontos da UFES para verificar a qualidade do sinal. Além disso, os experimentos foram realizados em três dias diferentes, mas com condições atmosféricas similares para que os resultados fossem equivalentes. Por fim, será mostrada a cobertura da rede LoRa Mesh UFES (seção 4.4).

4.2 Teste de alcance com dois nós

O primeiro teste realizado foi com apenas dois dispositivos com o intuito de avaliar o desempenho com os parâmetros configurados. Foi escolhido um ponto fixo para um nó, localizado no prédio CT13, no Laboratório de Pesquisa em Redes e Multimídia (LPRM), de modo que o dispositivo ficasse no alto próximo a janela ao lado de fora. Já o outro nó teve como finalidade mapear a área em diferentes pontos da UFES.

4.2.1 Nó M2

O dispositivo chamado de “Mesh 2” (M2) foi o escolhido para ficar no LPRM, ele funciona como o nó principal dos testes, onde os dados devem chegar nele para serem armazenados. Em uma analogia, ele funciona como um gateway, já que é o nó final e tem acesso à rede WiFi.

As configurações ajustadas para o M2 foram as descritas na seção 3.3, utilizando a região ANZ, o canal UFES, Largura de Banda de 125 kHz, Taxa de Codificação 4/8 e Fator de Espalhamento 12, em outras palavras, esse canal terá uma baixa taxa de bits/segundo mas terá um longo alcance.

Além dessas características gerais, foi ativado o WiFi para o M2 para que fosse possível obter os dados salvos que fora recebido pelo outro dispositivo, pelo sistema Web. Para ativar o WiFi, foi necessário utilizar um hotspot móvel, que é um ponto de WiFi criado pela rede 4G do celular, que pôde ser acessado mediante login e senha pelo M2. O comando utilizado para ativar o WiFi foi o seguinte:

```
meshtastic --set wifi_ap_mode false
meshtastic --set wifi_ssid "matheus_bozzi"
meshtastic --set wifi_password "matheusb"
```

Além do WiFi, outra característica específica para esse nó foi a posição fixa em latitude, longitude e altura do LPRM, que foi definida da seguinte forma:

```
meshtastic --setlat -20.27360218680015
meshtastic --setlon -40.30607780865459
meshtastic --setalt 10
```

Por fim, a última configuração específica para esse nó foi ativar um método de salvar as mensagens, para que fosse possível analisar os dados de forma mais eficaz. Para isso, foi utilizado os seguintes comandos:

```
meshtastic --set range_test_plugin_enabled true
meshtastic --set range_test_plugin_save true
```

Após essas configurações, as informações do M2 estavam dispostas como mostra o Apêndice B - Registro B.4. Diante disso, pode-se observar que a rede LoRa mesh contém os três dispositivos, bem como observar em “Preferences” as alterações feitas.

4.2.2 Nó M1

O nó “Mesh 1” (M1) foi o nó “andarelo”, isto é, foi utilizado para percorrer a área, para assim definir o alcance e analisar os dados. O M1 foi utilizado para enviar as mensagens para o M2 com a localização em latitude e longitude.

Para isso, foi necessário o uso do aplicativo visto na seção 3.4.2, no qual facilitou o envio das mensagens, como mostra a Figura 19 ao lado.

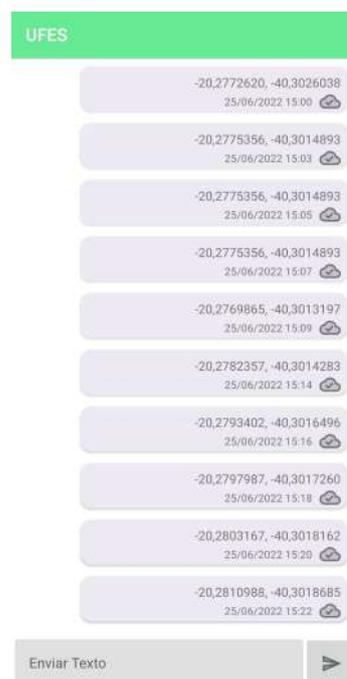


Figura 19 – Mensagens enviadas para o Canal UFES, pelo aplicativo Android

Fonte: Autor, 2022.

Como pôde-se notar, a mensagem enviada é dada pela (latitude,longitude) e além do mais, conta com um ícone de nuvem no canto superior direito que indica se o dispositivo está conectado a outro.



Figura 20 – Ícone de conexão, no aplicativo Android

Fonte: Autor, 2022.

Na Figura 20, a nuvem com uma barra indica que sua mensagem não chegou a nenhum dispositivo, ou seja, não recebeu nenhum ACK de confirmação e o pacote foi dado como perdido. Já a nuvem com uma marca de “correto” indica que a mensagem chegou a um nó da malha.

Nenhuma configuração além das realizadas na seção 3.3 foi necessária para no M1, logo sua configuração é mostrada no Apêndice B - Registro B.5.

4.2.3 Resultados com dois nós

Feito todas as configurações dos nós, e utilizando o Mesh 1 para mapear a área. O Indicador de Intensidade de Sinal visto na seção 2.2.4 foi levado em consideração para verificar a qualidade do sinal. O Valor do SNR com o Fator de Espalhamento 12 (que foi o valor configurado no canal utilizado pelos nós), demodula sinais com até -20dB, ou seja, pode demodular sinais abaixo do nível de ruído, como visto na seção 2.2.4.2.

A Tabela 7 mostra os dados obtidos a partir dos testes, onde a primeira coluna indica a numeração dos pontos em que as mensagens foram enviadas, em sequência. A segunda coluna representa o horário da mensagem recebida. A terceira representa o nome do Nó remetente da mensagem. A quarta e quinta coluna representa a localização do Mesh 2, em latitude e longitude, que é fixa no LPRM. A sexta coluna indica o SNR. A sétima coluna indica a quantidade de saltos restantes que a mensagem possui dentro da rede até parar de ser transmitida. A oitava coluna representa os dados recebidos pelo canal UFES.

Tabela 7 – Dados obtidos pelo nó Mesh 2

ID	time	sender name	rx lat	rx long	rx snr	hop limit	payload
M2	13:10:35	mesh 1	-20.273.602	-40.306.078	9,50	3	-20,2736242, -40,3061225
1	13:14:15	mesh 1	-20.273.602	-40.306.078	11,00	3	-20,2732433, -40,3065936
2	13:17:34	mesh 1	-20.273.602	-40.306.078	4,50	3	-20,2720563, -40,3065104
3	13:19:59	mesh 1	-20.273.602	-40.306.078	-4,00	3	-20,2715638, -40,3060538
4	13:25:30	mesh 1	-20.273.602	-40.306.078	-10,00	3	-20,2712820, -40,3053520

ID	time	sender name	rx lat	rx long	rx snr	hop limit	payload
5	13:29:13	mesh 1	-20.273.602	-40.306.078	-18,75	3	-20,2718994, -40,3046392
6	13:33:45	mesh 1	-20.273.602	-40.306.078	-14,75	3	-20,2730036, -40,3037702
7	13:37:09	mesh 1	-20.273.602	-40.306.078	-19,00	3	-20,2737600, -40,3031711
8	13:40:33	mesh 1	-20.273.602	-40.306.078	-19,75	3	-20,2742400, -40,3028770
9	13:43:57	mesh 1	-20.273.602	-40.306.078	-19,00	3	-20,2748837, -40,3023339
10	13:47:44	mesh 1	-20.273.602	-40.306.078	-11,50	3	-20,2752039, -40,3012955
11	13:49:55	mesh 1	-20.273.602	-40.306.078	-13,75	3	-20,2759832, -40,3012231
12	14:06:10	mesh 1	-20.273.602	-40.306.078	-15,00	3	-20,2775296, -40,3032552
13	14:09:30	mesh 1	-20.273.602	-40.306.078	-14,25	3	-20,2772541, -40,3034386
14	14:20:44	mesh 1	-20.273.602	-40.306.078	-18,25	3	-20,2751347, -40,3034433
15	14:23:02	mesh 1	-20.273.602	-40.306.078	-16,50	3	-20,2746416, -40,3038909
16	14:25:45	mesh 1	-20.273.602	-40.306.078	-14,50	3	-20,2744337, -40,3039556
17	14:28:25	mesh 1	-20.273.602	-40.306.078	-21,00	3	-20,2748340, -40,3044759
18	14:41:36	mesh 1	-20.273.602	-40.306.078	-6,25	3	-20,2742749, -40,3047251
19	14:43:30	mesh 1	-20.273.602	-40.306.078	-9,50	3	-20,2735770, -40,3047452
20	14:45:09	mesh 1	-20.273.602	-40.306.078	-12,00	3	-20,2729338, -40,3050395
21	14:47:28	mesh 1	-20.273.602	-40.306.078	-2,75	3	-20,2722652, -40,3057537
22	16:56:39	mesh 1	-20.273.602	-40.306.078	8,75	3	-20,274097, -40,305754
23	17:00:50	mesh 1	-20.273.602	-40.306.078	-6,75	3	-20,274783, -40,305383
24	17:03:35	mesh 1	-20.273.602	-40.306.078	-11,25	3	-20,2757737, -40,3058400
25	17:09:05	mesh 1	-20.273.602	-40.306.078	-18,50	3	-20,2764080, -40,3061672
26	17:26:22	mesh 1	-20.273.602	-40.306.078	-17,25	3	-20,2775432, -40,3069693
27	17:29:06	mesh 1	-20.273.602	-40.306.078	-16,50	3	-20,2781747, -40,3074306
28	17:56:41	mesh 1	-20.273.602	-40.306.078	-18,00	3	-20,2774464, -40,3040740
29	18:06:26	mesh 1	-20.273.602	-40.306.078	-13,75	3	-20,276813, -40,304387
30	18:08:34	mesh 1	-20.273.602	-40.306.078	3,25	3	-20,275734, -40,304903
31	18:13:03	mesh 1	-20.273.602	-40.306.078	6,25	3	-20,274877, -40,305145
32	18:19:12	mesh 1	-20.273.602	-40.306.078	7,25	3	-20,274054, -40,305746

Diante dos dados mostrados na Tabela 7 acima, foi desenvolvido um mapa geográfico de calor representando todos os pontos mapeados.

A paleta de cores ajuda com a visualização dos pontos em que o sinal foi enviado pelo remetente, indicando um melhor ou pior sinal. A cor vermelha representa um valor alto, próximo de 10dB. A cor amarela representa o SNR próximo de 4dB. A cor verde representa um SNR de -2dB. A cor azul clara, SRN de -8dB. O lilás/azul-escuro representa um SNR próximo de -14dB. Por fim, o rosa representa um sinal bem fraco, próximo ao limite inferior de -20dB.

Assim, pode-se verificar a representação dos dados na Figura 21.

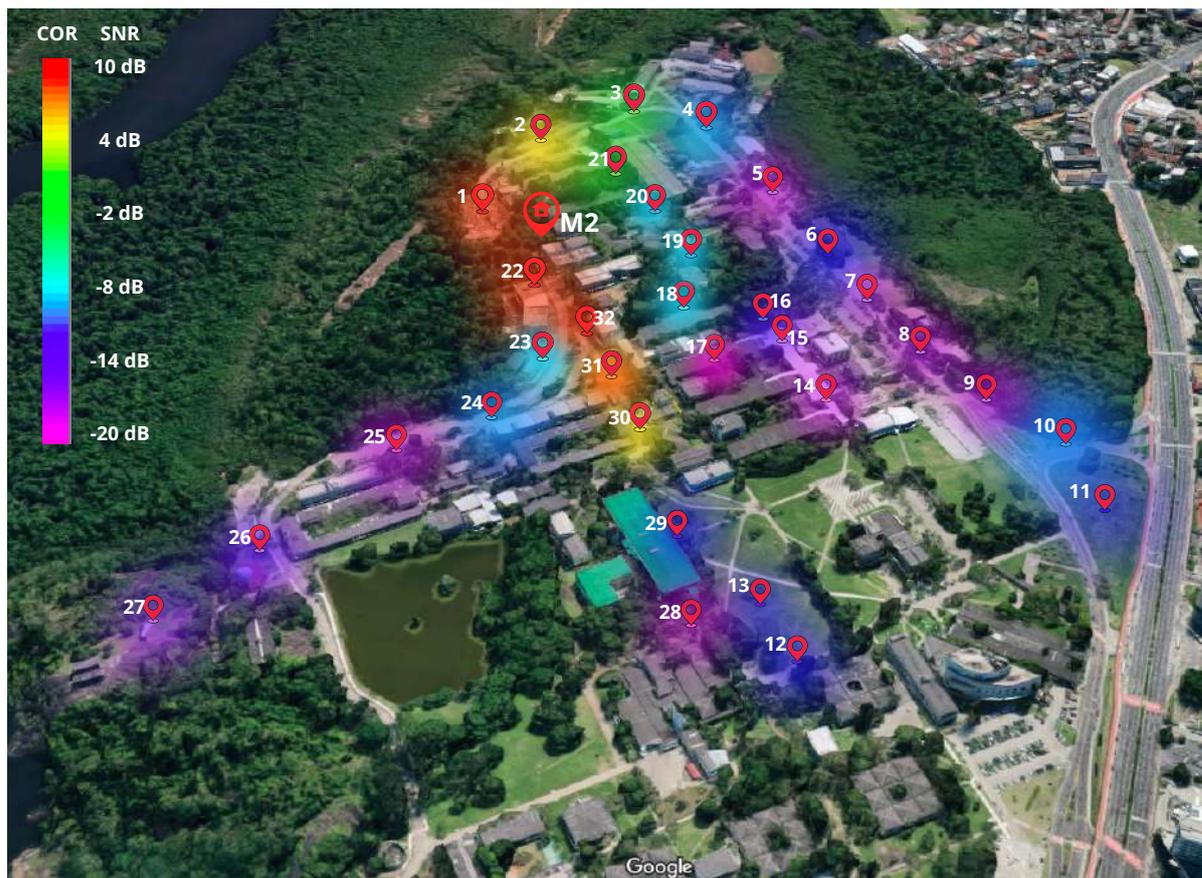


Figura 21 – Mapa de Calor na UFES em relação ao Sinal-Ruído
Fonte: Autor, 2022.

Diante do que foi apresentado de forma visual, pode-se assumir alguns resultados consistentes. Próximo ao M2, que representa o Mesh 2, o sinal é bem forte, isto é, a chance dos dados conseguirem ser transmitidos é muito alta. Esses pontos próximos não possuem muitos obstáculos e por isso estão entre 11dB e 7.25dB.

Ao passo que o M1 foi se afastando, o sinal gradativamente piorava. Além disso, os testes foram feitos em solo e muitas construções e vegetações eram um obstáculo que prejudicava muito a qualidade e alcance do sinal. Os pontos 26 e 27 estão a grandes distâncias, já que são lugares com maior altitude com menores obstáculos, como mostra a Figura 22. Também os pontos 11 e 12 por não terem muitos obstáculos.

Por outro lado, os lugares em meio aos prédios e vegetação intensa faziam com que a mensagem enviada não chegasse ao destino, nos pontos 11 e 12, 13 e 14, 25 e 26 e entre 26 e 28.

Como consequência desses resultados, pode-se afirmar que com as configurações feitas nas seções 4.2.1 e 4.2.2, a área aproximada de cobertura útil com dois nós foi de 23 Hectares, o que significa 230 mil metros quadrados. E as distâncias em linha reta foram de 542 metros de M2 até o ponto 27, 550 metros de M2 até o ponto 12 e a distância máxima foi de 588 metros de M2 ao ponto 11.



Figura 22 – Nó Mesh 1 no Observatório da UFES, ponto 27 da Figura 21
Fonte: Autor, 2022.

4.3 Teste de alcance com três nós

Após a realização do primeiro teste com dois nós, optou-se por escolher um lugar “central” para que o nó M1 ficasse localizado, e assim ser possível avançar com o Nó “Mesh 3” mapeando uma área maior. Em especial, verificar como a rede LoRa mesh se comportaria com o protocolo de roteamento.

As configurações dos nós para essa etapa dos testes foram as seguintes:

- O nó M2 ficou com a mesma configuração realizada na seção 4.2.1;
- O nó M1 ficou posicionado em frente a biblioteca central, que foi o ponto “central” escolhido. A alteração feita nesse nó foi a capacidade de salvar um arquivo com os dados advindos do Mesh 3, assim como o Mesh 2.
- O nó Mesh 3 (M3) por sua vez, não teve nenhuma configuração além das feitas na seção 3.3. E suas configuração ficaram como mostra o Apêndice B - Registro B.6.

4.3.1 Resultados com três nós

Feito todas as configurações, o Mesh 2 ficou localizado no LPRM, o Mesh 1 ficou em frente a Biblioteca Central da UFES (Figura 23) e o Mesh 3 foi se afastando gradativamente de forma a mapear a área.



Figura 23 – Nó Mesh 1 em frente a Biblioteca Central da UFES
Fonte: Autor, 2022.

A Tabela 8 mostra os dados obtidos a partir dos testes, de forma análoga ao teste feito com dois nós (seção 4.2.3).

Tabela 8 – Dados obtidos pelos nós Mesh 2 e Mesh 1

ID	time	sender name	rx lat	rx long	rx snr	hop limit	payload
33	14:18:17	mesh 1	-20.273.602	-40.306.078	8,50	3	-20,2736160, -40,3061724
34	14:18:50	mesh 3	-20.273.602	-40.306.078	10,75	3	-20,2736219, -40,3061614
35	14:43:32	mesh 3	-20.273.602	-40.306.078	3,75	3	-20,2746576, -40,3054124
36	14:45:02	mesh 1	-20.273.602	-40.306.078	6,50	3	-20,2746523, -40,3054130
37	14:50:01	mesh 3	-20.273.602	-40.306.078	-17,25	3	-20,2765305, -40,3041370
38	14:51:20	mesh 1	-20.273.602	-40.306.078	-17,50	3	-20,2765195, -40,3040344
39	14:56:56	mesh 3	-20.273.602	-40.306.078	8,25	2	-20,2761461, -40,3031104
40	14:58:53	mesh 3	-20.273.602	-40.306.078	7,75	2	-20,2766695, -40,3031871
41	15:01:00	mesh 3	-20.273.602	-40.306.078	3,25	2	-20,2772620, -40,3026038
42	15:07:44	mesh 3	-20.273.602	-40.306.078	-5,5	2	-20,2775356, -40,3014893
43	15:10:33	mesh 3	-20.273.602	-40.306.078	-2,00	2	-20,2769865, -40,3013197
44	15:14:18	mesh 3	-20.273.602	-40.306.078	-7,75	2	-20,2782357, -40,3014283
45	15:16:42	mesh 3	-20.273.602	-40.306.078	-11,50	2	-20,2793402, -40,3016496
46	15:18:48	mesh 3	-20.273.602	-40.306.078	-15,75	2	-20,2797987, -40,3017260
47	15:20:49	mesh 3	-20.273.602	-40.306.078	-19,00	3	-20,2803167, -40,3018162
48	15:23:07	mesh 3	-20.273.602	-40.306.078	-20,25	2	-20,2810988, -40,3018685
49	15:45:19	mesh 3	-20.273.602	-40.306.078	-15,50	2	-20,2789826, -40,3026702
50	15:48:38	mesh 3	-20.273.602	-40.306.078	-1,75	2	-20,2778161, -40,3035637
51	15:51:12	mesh 3	-20.273.602	-40.306.078	-7,5	2	-20,2777083, -40,3042744
52	15:54:18	mesh 3	-20.273.602	-40.306.078	-15,50	2	-20,2782199, -40,3046513
53	15:55:42	mesh 3	-20.273.602	-40.306.078	-19,00	2	-20,2789301, -40,3049302

ID	time	sender name	rx lat	rx long	rx snr	hop limit	payload
54	15:57:27	mesh 3	-20.273.602	-40.306.078	-21,75	3	-20,2791788, -40,3052977
54	15:57:21	mesh 3	-20.273.602	-40.306.078	-18,75	2	-20,2791788, -40,3052977
55	16:05:05	mesh 3	-20.273.602	-40.306.078	-19,75	3	-20,2784703, -40,3057540
56	16:11:14	mesh 3	-20.273.602	-40.306.078	-17,25	2	-20,2775007, -40,3069976
57	16:15:17	mesh 3	-20.273.602	-40.306.078	-18,25	3	-20,2784272, -40,3074894
57	16:15:29	mesh 3	-20.273.602	-40.306.078	-20,25	2	-20,2784272, -40,3074894
58	16:32:46	mesh 3	-20.273.602	-40.306.078	-8,50	2	-20,2763081, -40,3053396
58	16:32:54	mesh 3	-20.273.602	-40.306.078	-13,25	3	-20,2763081, -40,3053396
59	16:35:54	mesh 3	-20.273.602	-40.306.078	-17,25	2	-20,2767132, -40,3048940

Diante dos dados mostrados na Tabela 8 acima, foi desenvolvido um mapa geográfico de calor representando todos os pontos mapeados, de forma análoga ao teste com dois nós (seção 4.2.3). Pode-se verificar os dados na Figura 24.



Figura 24 – Mapa de Calor na UFES em relação ao Sinal-Ruído
Fonte: Autor, 2022.

Em consideração ao que é apresentado de forma visual na Figura 24, pode-se assumir alguns resultados relevantes. O percurso inicial foi ir em direção ao ponto “central” que ficaria o Mesh 1, portanto aos pontos 33, 34, 35, 36, 37 e 38 foram dados enviados alternadamente entre os nós M1 e M3 para o M2. Nota-se que próximo ao M2 o sinal é bem forte, isto é, a chance dos dados conseguirem ser transmitidos é muito alta em ambos os nós remetentes.

Ao passo que chegou-se ao ponto 37 e 38 foi deixado o M1 em frente a Biblioteca (Figura 23) e foi se afastando com o M3, o sinal gradativamente piorava já que os testes foram feitos em solo e muitas construções e vegetações eram um obstáculo que prejudicava muito a qualidade e alcance do sinal.

Como pôde-se ver na Tabela 8, muitos pontos possuem Hop Limit como 2. Isso significa que o protocolo de roteamento estava em funcionamento, já que a mensagem não conseguia chegar em M2 sem antes passar por M1, utilizando assim um salto e portanto podendo ter apenas mais dois saltos. Esse resultado foi um dos pontos destaques desse trabalho.

Prosseguindo com o mapeamento, foi possível alcançar o ponto 48, que está a 554 metros de distância M1 e 960 metros de distância de M2. A dificuldade do sinal entre os prédios foi constatada entre os pontos 48, 50 e 54, sendo perdidos em sua totalidade.

Outro ponto a se destacar foram que os pontos 54, 55 e 58 tiveram as mensagens chegando diretamente ao M2 e chegando também ao M1, e as mensagens duplicadas são descartadas ao chegar no Mesh 2. Esse resultado também é de grande importância para a rede LoRa mesh, já que não permite a redundância de mensagens.

4.4 Cobertura da rede LoRa mesh UFES

Como consequência dos testes realizados mostrados nas seções 4.2 e 4.3 e diante dos resultados neles apresentados, pode-se afirmar que com a utilização do Mesh 1 como dispositivo central foi possível obter um alcance em área expandido de forma significativa. A maior distância em linha reta foi próxima de 1Km e a área útil total aumentou de 23 Hectares para aproximadamente 36 Hectares, isto é, 360 mil metros quadrados. Além disso, com três dispositivos notou-se que o funcionamento da rede LoRa mesh se mostrou eficiente e cobriu grande parte da Universidade.

A Figura 25, abaixo, mostra uma interseção entre os testes com dois e três nós, o que indica a área aproximada de cobertura da rede Lora mesh UFES.



Figura 25 – Mapa de cobertura da rede LoRa mesh UFES
Fonte: Autor, 2022.

5 Conclusão

5.1 Considerações Finais

A maior utilização de serviços que necessitam de coleta e transmissão de dados faz com que o termo “IoT e Cidades Inteligentes” seja colocado cada vez mais em evidência. Este trabalho discutiu sobre a tecnologia LoRa. Todavia, o LoRa foi desenvolvido para ser usado com um Gateway, e este possui alto custo. Por isso, optou-se por utilizar uma rede LoRa mesh onde todos os dispositivos formam uma rede se comunicando entre si.

Diante desse trabalho, foi necessário obter um maior conhecimento sobre as redes sem fios, o protocolo LoRaWAN e as possibilidades de configurações dos parâmetros como Spreading Factor, Code Rate e Bandwidth, que personalizam a modulação LoRa e influenciam diretamente na transmissão dos pacotes e por conseguinte podem aumentar ou diminuir o alcance e a taxa de bits/segundos, interferindo assim no consumo energético. O conhecimento acerca dos Indicadores de Intensidade de Sinal como RSSI e SNR foram fundamentais para verificar a qualidade do sinal. Dito isso, as configurações corretas dos dispositivos foram imprescindíveis para atingir a finalidade principal do trabalho.

A rede LoRa mesh por sua vez, aumenta progressivamente ao passo que são inseridos novos dispositivos e por isso há uma escalabilidade da área de cobertura, porém isso aumenta o custo para ampliar a infraestrutura, sendo necessário um protocolo de roteamento eficiente de tal forma que não sobrecarregue a banda da rede e assim cause interferências prejudicando a transmissão dos dados. Uma limitação do trabalho desenvolvido foi justamente essa, já que o protocolo de roteamento utilizado foi o Flooding (Inundação) que é um protocolo de roteamento de fácil implementação, visto que cada nó envia a mensagem pela rede de forma “broadcast”, ou seja, para todos os seus vizinhos. Desse modo, existe um uso excessivo da rede e os dispositivos devem descartar as mensagens já recebidas, não exigindo registros de topologia ou grandes quantidades de análises.

Este protocolo de roteamento se mostrou confiável para o trabalho, visto que os testes foram feitos com apenas três dispositivos. Os dados enviados pela rede Lora mesh utilizavam os saltos entre os dispositivos para chegar ao dispositivo final que armazenava os dados e foi possível notar também a recusa das mensagens que passavam por rotas alternativas para chegar dispositivo final. Dessa forma foi possível concluir o funcionamento da rede LoRa mesh de forma adequada.

Um outro ponto que deve-se destacar se trata do Hardware e suas configurações. Foi um desafio muito grande adaptar o Heltec Wireless Stick Esp32 ao Meshtastic, visto que foi necessário encontrar uma versão em que o firmware satisfizesse todos os recursos

utilizados. Como o projeto está em andamento, possui alguns bugs que ainda não foram resolvidos. Além disso, foi necessário também certificar todas as pinagens, antena e tela OLED antes de fazer o envio do código para o dispositivo, uma vez que ligar o dispositivo sem antes checar pode danificá-lo. Para isso, foi necessário um estudo cuidadoso sobre as configurações do Hardware e os demais recursos utilizados. As adaptações e estudo de forma acentuada nos códigos foram essenciais para o sucesso da rede LoRa mesh UFES.

Com o auxílio do projeto *meshtastic* que possui diversas funcionalidades para a implantação de uma rede mesh, foi muito utilizado o *meshtastic CLI*. Este recurso facilita na definição de parâmetros essenciais em uma rede LoRa por meio da linha de comando, pois para esse projeto se fez necessário definir o nome dos dispositivos, a frequência do rádio, configurações do canal UFES definida por uma URL, entre outras coisas como criptografia, velocidade e taxa de bits. Portanto, essa ferramenta de código aberto foi extremamente importante no desenvolvimento do projeto.

Com os dados dos resultados obtidos nos testes, a construção e utilização da rede LoRa mesh UFES comprovou-se viável. A robustez e alcance da transmissão foi comprovada através dos testes em campo, mesmo diante da pouca altura das antenas, os resultados dos testes com apenas dois dispositivos puderam servir na decisão para definir um local central estratégico para o teste com três dispositivos. Esses testes precisaram ser refeitos para alterar algumas configurações e, portanto, serviu como aprendizado de como funciona na prática a rede LoRa mesh vista em teoria. Outrossim, deixou claro que não é apenas a distância que influencia a qualidade da comunicação, mas também o relevo do solo e os obstáculos como prédios e vegetação.

Através da rede LoRa mesh implantada foi possível cobrir, com apenas três dispositivos, cerca de 36 Hectares de área, o que representa 62% da área total da Universidade Federal do Espírito Santo, portanto sendo satisfatória pelo fato dos testes terem sido feitos em solo e com muitos obstáculos como prédios e vegetação densa. A maior distância entre os dispositivos LoRa foi de aproximadamente 1 Km, confirmando o longo alcance pretendido.

A partir disso, pode-se concluir que a tecnologia LoRaWAN traz enormes benefícios para as soluções IoT principalmente na cobertura urbana e rural e de longa distância. E a rede LoRa mesh apresentada neste trabalho poderá ser usada em mais pesquisas, sendo assim um trabalho incremental. Além do mais, poderá servir como referência no desenvolvimento de soluções no ramo de “IoT e Cidades Inteligentes” não só para Universidade Federal do Espírito, mas também para a cidade de Vitória.

5.2 Trabalhos Futuros

Este estudo teve como principal objetivo introduzir a tecnologia LoRaWAN e mapear a área acadêmica da UFES e para isto optou-se por utilizar a rede mesh, onde os dispositivos utilizam uns dos outros para o transporte das informações. Assim, o protocolo de roteamento é de suma importância. Um possível trabalho futuro seria no âmbito da melhoria do protocolo utilizado, que por sua vez é muito simples. Este novo protocolo poderá usar os valores SNR para calcular as melhores rotas, e para testar e analisar os resultados seriam necessários mais dispositivos.

Um outro estudo interessante seria a realização de mais testes para apurar melhor a perda de pacotes, utilizando diversas configurações que influenciam na modulação do sinal LoRa, e também a utilização de antenas com maior capacidade. Para isso, seria necessário fazer alguns protótipos para que os dispositivos possam ficar em lugares externos e altos de maneira que o sinal tenha menos obstáculos. Dessa forma, seria fundamental verificar a disponibilidade dos lugares onde esses dispositivos ficariam e com isso poderia aumentar a área de alcance da rede LoRa mesh.

Um outro trabalho poderia utilizar o conceito de telemetria que permite a medição e comunicação de informações de interesses, assim sendo essencial adicionar sensores aos dispositivos. Os dados podem ser enviados pela rede até chegar ao dispositivo final, e este usaria algum protocolo, por exemplo o MQTT, e enviaria os dados para plataformas IoT.

Por fim, um outro trabalho importante seria manter essa rede em funcionamento constante. Dessa forma, seria possível que novos nós entrem e usufruam da rede, provendo de fato uma rede comunitária.

Referências

ADELANTADO, F. et al. Understanding the limits of lorawan. *IEEE Communications Magazine*, v. 55, n. 9, p. 34–40, 2017. Citado 2 vezes nas páginas 21 e 24.

ANATEL. *Agência Nacional de Telecomunicações - Ato Nº 14448, de 04 de dezembro de 2017*. 2017. Disponível em: <https://sei.anatel.gov.br/sei/publicacoes/controlador_publicacoes.php?acao=publicacao_visualizar&id_documento=2549681&id_orgao_publicacao=0>. Citado na página 27.

AUGUSTIN, A. et al. A study of lora: Long range amp; low power networks for the internet of things. *Sensors*, v. 16, n. 9, 2016. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/16/9/1466>>. Citado na página 24.

BOARDS » HELTEC WIRELESS STICK. 2022. Disponível em: <https://docs.platformio.org/en/latest/boards/espressif32/heltec_wireless_stick.html>. Citado na página 38.

BOR, M.; VIDLER, J.; ROEDIG, U. Lora for the internet of things. In: *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. USA: Junction Publishing, 2016. (EWSN '16), p. 361–366. ISBN 9780994988607. Citado na página 21.

BUNGART, J. *Redes de computadores: Fundamentos e protocolos*. SENAI-SP Editora, 2017. (Tecnologia da Informação). ISBN 9788583937647. Disponível em: <<https://books.google.com.br/books?id=S6w2DwAAQBAJ>>. Citado na página 19.

CENTENARO, M. et al. Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, v. 23, n. 5, p. 60–67, 2016. Citado 2 vezes nas páginas 14 e 17.

DELLALIBERA; ARTEN, L. Técnicas de múltiplo acesso para redes lorawan. 2018. 48 f. trabalho de conclusão de curso (bacharelado - engenharia de telecomunicações) - universidade estadual paulista julio de mesquita filho, câmpus experimental de são joão da boa vista. Disponível em: <<http://hdl.handle.net/11449/156791>>, 2018. Citado na página 21.

DELY, P.; KASSLER, A.; BAYER, N. Openflow for wireless mesh networks. In: *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*. [S.l.: s.n.], 2011. p. 1–6. Citado 2 vezes nas páginas 27 e 28.

ESPHOME. *ESPHome-Flasher Download for Ubuntu*. 2022. Disponível em: <<https://github.com/esphome/esphome-flasher/releases>>. Citado na página 40.

GOOGLE. *Download Google Chrome for Ubuntu*. 2022. Disponível em: <https://www.google.com/chrome/?brand=BNSD&gclid=Cj0KCQjwntCVBhDdARIsAMEwACkMZs5NR7D7Hnm_2fsL9xnSy2zoABm1u1F6P8-UzHUBf8dnqWCxnpwaAuRMEALw_wcB&gclsrc=aw.ds>. Citado na página 36.

Heltec Automation. *Wireless Stick*. 2022. Disponível em: <<https://heltec-automation-docs>>.

readthedocs.io/en/latest/esp32/wireless_stick/index.html>. Citado na página 31.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a Internet: uma abordagem top-down*. Brazil: Pearson, 2013. 896 p. (Redes de Computadores). ISBN 978-8581436777. Disponível em: <https://www.google.com.br/books/edition/Redes_de_Computadores/57BIAgAAQBAJ?hl=pt-BR&gbpv=0&kptab=getbook>. Citado 3 vezes nas páginas 19, 28 e 29.

LoRa Alliance. *A technical overview of LoRa® and LoRaWAN™*. 2015. Disponível em: <<https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>>. Citado 2 vezes nas páginas 24 e 25.

LoRa Alliance. 2022. Disponível em: <<https://lora-alliance.org>>. Citado 2 vezes nas páginas 14 e 20.

LoRa Alliance. *ANATEL (BRAZIL)*. 2022. Disponível em: <https://lora-alliance.org/regulatory_certifications/anatel-brazil/>. Citado 2 vezes nas páginas 16 e 27.

LoRa Alliance. *LoRaWAN 1.1 Specification*. 2022. 25 p. Disponível em: <https://loraalliance.org/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf>. Citado na página 21.

LUNDELL, D. et al. A routing protocol for lora mesh networks. In: *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. [S.l.: s.n.], 2018. p. 14–19. Citado 2 vezes nas páginas 28 e 29.

MEIRELES, L. V. Projeto de uma rede de internet das coisas para monitoramento e alerta de emergência em Áreas de risco. f. 136, 2018. Citado na página 27.

MEKKI, K. et al. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, Elsevier, v. 5, n. 1, p. 1–7, mar. 2019. Disponível em: <<https://hal.archives-ouvertes.fr/hal-01670379>>. Citado na página 21.

Meshtastic-Android/Releases. *Meshtastic-android project - Releases*. 2022. Disponível em: <<https://github.com/meshtastic/Meshtastic-Android/releases>>.

Meshtastic-devices. *Meshtastic-devices project*. 2022. Disponível em: <<https://github.com/meshtastic/meshtastic-device>>.

Meshtastic-python. *Meshtastic-python project*. 2022. Disponível em: <<https://github.com/meshtastic/Meshtastic-python>>. Citado na página 36.

Meshtastic-python/Releases. *Meshtastic-python project - Releases*. 2022. Disponível em: <<https://github.com/meshtastic/Meshtastic-python/releases>>.

Meshtastic® . *Meshtastic - Geeksville Industries LLC*. 2022. Disponível em: <<https://meshtastic.org>>. Citado na página 30.

MICROSOFT©. *Download VScode Debian, Ubuntu*. 2022. Disponível em: <<https://code.visualstudio.com/download>>. Citado na página 36.

OLIVEIRA, L. et al. Mac layer protocols for internet of things: A survey. *Future Internet*, v. 11, n. 1, 2019. ISSN 1999-5903. Disponível em: <<https://www.mdpi.com/1999-5903/11/1/16>>. Citado na página 23.

- PAAVOLA, M.; LEIVISKA, K. Wireless sensor networks in industrial automation. In: SILVESTRE-BLANES, J. (Ed.). *Factory Automation*. Rijeka: IntechOpen, 2010. cap. 10. Disponível em: <<https://doi.org/10.5772/9532>>. Citado na página 20.
- PASSOS, D. et al. Mesh network performance measurements. 01 2005. Citado 2 vezes nas páginas 15 e 17.
- PETERSON, L. L.; DAVIE, B. S. *Redes de computadores : uma abordagem de sistemas*. Rio de Janeiro : Elsevier, 2013. 545 p. (Redes de Computadores). ISBN 978-8535248975. Disponível em: <<https://docplayer.com.br/11441530-Redes-de-computadores-uma-abordagem-de-sistemas.html>>. Citado na página 27.
- PLATFORMIO. *Download PlatformIO for VSCode*. 2022. Disponível em: <<https://platformio.org/install/ide?install=vscode>>. Citado na página 36.
- PROJECT CONFIGURATION FILE PLATFORMIO.INI. 2022. Disponível em: <<https://docs.platformio.org/en/latest/projectconf/index.html>>. Citado na página 38.
- PROJECT CONFIGURATION FILE SECTION: WORKING. 2022. Disponível em: <https://docs.platformio.org/en/latest/projectconf/section_env.html#working-env-name>. Citado na página 38.
- RABUSKE, M. H. B. Estudo do protocolo lorawan e implementação de uma rede privada com nós distribuídos. f. 70, 2017. Citado na página 27.
- RIZZETTI, T. A. Novos métodos para prover segurança à comunicação no âmbito de redes elétricas inteligentes. f. 198, 2018. Citado na página 27.
- SCHAFFERS, H. et al. Smart cities and the future internet: Towards cooperation frameworks for open innovation. In: DOMINGUE, J. et al. (Ed.). *The Future Internet*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 431–446. ISBN 978-3-642-20898-0. Citado na página 14.
- Semtech. *LoRa Technology Overview*. 2022. Acesso em: 02/03/2022. Disponível em: <<https://www.semtech.com/lora/>>. Citado 4 vezes nas páginas 15, 16, 20 e 23.
- SILVA, R. V. Simulação da integração de uma rede lorawan com satélites de baixa Órbita. f. 53, 2021. Citado na página 22.
- SONG, Q.; NUAYMI, L.; LAGRANGE, X. Survey of radio resource management issues and proposals for energy-efficient cellular networks that will cover billions of machines. *EURASIP Journal on Wireless Communications and Networking*, v. 2016, 06 2016. Citado na página 16.
- SOUZA, F. V. M. Uma arquitetura lpwan de custo acessível para cidades inteligentes. f. 68, 2019. Citado na página 14.
- TANENBAUM, A. S. *Rede de computadores*. Brasil: Pearson, 2011. 604 p. (Redes de Computadores). ISBN 9788576059240. Disponível em: <<https://www.bvirtual.com.br/NossoAcervo/Publicacao/2610>>. Citado na página 20.
- TTN. *RSSI and SNR*. 2022. Disponível em: <<https://www.thethingsnetwork.org/docs/lorawan/rssi-and-snr/>>. Citado na página 22.

VENTURY, S. N. *Fundamentos de redes de computadores*. ESTACIO - POD, 2016. 176 p. (Tecnologia da Informação). ISBN 978-8555482571. Disponível em: <<https://www.martinsfontespaulista.com.br/fundamentos-de-redes-de-computadores-937607/p>>. Citado na página 19.

ZHAO, L. et al. Flooding and directed diffusion routing algorithm in wireless sensor networks. In: *2009 Ninth International Conference on Hybrid Intelligent Systems*. [S.l.: s.n.], 2009. v. 2, p. 235–239. Citado na página 29.

Apêndices

APÊNDICE A – Arquivos de Configurações

Código A.1 – Código platformio.ini no diretório variants

```

1 ; https://docs.platformio.org/en/latest/boards/espressif32/heltec_wireless_stick.html?utm_source=platformio&utm_medium=piohome
2 [env:heltec_wireless_stick]
3 extends = esp32_base
4 ; platform = espressif32
5 board = heltec_wireless_stick
6 build_flags =
7   \${esp32_base.build_flags} -D PRIVATE_HW -I variants/heltec_wireless_stick

```

Código A.2 – Código variant.h

```

1 #define BUTTON_PIN 0 // sera usado para o pressionamento do botão
2
3 #ifndef USE_JTAG // verificacao caso nao tenha tela Oled
4 #define I2C_SDA 4 // I2C oled sda
5 #define I2C_SCL 15 // I2C oled scl
6 #endif
7
8 #define EXT_NOTIFY_OUT 13 // Pino para o Ext Notify Plugin.
9
10 #ifndef USE_JTAG
11 #define LORA_RESET 14
12 #endif
13
14 #define RESET_OLED 16 // reset Oled
15
16 #define VEXT_ENABLE 21 // vnext control
17 #define LED_PIN 25 // piscar o led
18
19 #define USE_RF95
20 #define LORA_DIO0 26 // sem conexão module SX1262
21
22 #define LORA_DIO1 35 // não usado
23 #define LORA_DIO2 34 // não usado
24
25 #undef GPS_RX_PIN
26 #undef GPS_TX_PIN
27 #define GPS_RX_PIN 36 // modulo gps
28 #define GPS_TX_PIN 37 // modulo gps
29
30 #define ADC_MULTIPLIER 3.8
31 #define BATTERY_PIN 37 //medicao de tensao da bateria

```

Código A.3 – Código platformio.ini no diretório raiz

```

1
2 [platformio]
3 ; default_envs = tbeam

```

```
4 ;default_envs = tbeam0.7
5 default_envs = heltec_wireless_stick
6 ;default_envs = tlorav1
7 ;default_envs = tlorav1
8 ;default_envs = tlorav1_3
9 ;default_envs = tlorav2
10 ;default_envs = tlorav2-1-1.6
11 ;default_envs = lorarelayv1 # nrf board
12 ;default_envs = t-echo
13
14 ; board specific config can be moved to the respective 'variants' file.
15 ; See https://docs.platformio.org/en/latest/projectconf/section\_platformio.html#
    extra-configs
16 extra_configs = variants/*/platformio.ini
17
18 ...
```

APÊNDICE B – Configurações dos dispositivos

Registro de Saída B.1 – Dispositivo com configurações padrões

```

1 $ meshtastic —info
2 Connected to radio
3
4 Owner: Unknown e01c (?1C)
5 My info: { "myNodeNum": 2823479324, "numBands": 13, "firmwareVersion": "1.2.64",
  "rebootCount": 251, "bitrate": 17.08847, "messageTimeoutMsec": 300000, "
  minAppVersion": 20200, "maxChannels": 8, "hasWifi": true, "channelUtilization
  ": 7.7133336, "airUtilTx": 0.7772222 }
6 Nodes in mesh: { 'num': 2823479324, 'user': { 'id': '!a84ae01c', 'longName': '
  Unknown e01c', 'shortName': '?1C', 'macaddr': '84:cc:a8:4a:e0:1c', 'hwModel':
  'PRIVATE_HW'}, 'position': {} }
7
8 Preferences: { "phoneTimeoutSecs": 900, "lsSecs": 300 }
9
10 Channels:
11   PRIMARY psk=default { "modemConfig": "Bw125Cr48Sf4096", "psk": "AQ==" }
12
13 Primary channel URL: https://www.meshtastic.org/d/#CgUYAyIBAQ

```

Registro de Saída B.2 – Resultado das configurações

```

1 $ meshtastic —port /dev/ttyUSB0 —info
2 Connected to radio
3
4 Owner: mesh 3 (m3)
5 My info: { "myNodeNum": 2823479324, "numBands": 20, "firmwareVersion": "1.2.64",
  "rebootCount": 253, "bitrate": 17.08847, "messageTimeoutMsec": 300000, "
  minAppVersion": 20200, "maxChannels": 8, "hasWifi": true, "channelUtilization
  ": 13.363333, "airUtilTx": 0.42066666 }
6 Nodes in mesh: { 'num': 2823479324, 'user': { 'id': '!a84ae01c', 'longName': 'mesh
  3', 'shortName': 'm3', 'macaddr': '84:cc:a8:4a:e0:1c', 'hwModel': 'PRIVATE_HW
  '}, 'position': {}, 'lastHeard': 1656139494 }
7
8 Preferences: { "phoneTimeoutSecs": 900, "lsSecs": 300, "region": "ANZ" }
9
10 Channels:
11   PRIMARY psk=default { "modemConfig": "Bw125Cr48Sf4096", "psk": "AQ==", "name":
  "UFES" }
12
13 Primary channel URL: https://www.meshtastic.org/d/#CgsYAyIBASoEVUZFUw

```

Registro de Saída B.3 – Rede LoRa mesh UFES configurada

```

1 $ meshtastic —port /dev/ttyUSB0 —info
2 Connected to radio

```

```

3
4 Owner: mesh 3 (m3)
5 My info: { "myNodeNum": 2823479324, "numBands": 20, "firmwareVersion": "1.2.64",
  "rebootCount": 254, "bitrate": 17.08847, "messageTimeoutMsec": 300000, "
  minAppVersion": 20200, "maxChannels": 8, "hasWifi": true, "airUtilTx":
  0.19066666 }
6 Nodes in mesh: { 'num': 2823479324, 'user': { 'id': '!a84ae01c', 'longName': 'mesh
  3', 'shortName': 'm3', 'macaddr': '84:cc:a8:4a:e0:1c', 'hwModel': 'PRIVATE_HW
  '}, 'position': {} } { 'num': 2823479488, 'user': { 'id': '!a84ae0c0', 'longName
  ': 'mesh 2', 'shortName': 'm2', 'macaddr': '84:cc:a8:4a:e0:c0', 'hwModel': '
  PRIVATE_HW'}, 'position': { 'latitudeI': -202736021, 'longitudeI': -403060778,
  'latitude': -20.273602099999998, 'longitude': -40.3060778}, 'snr': 11.0, '
  lastReceived': { 'from': 2823479488, 'to': 2823479324, 'id': 658023066, 'rxSnr
  ': 11.0, 'hopLimit': 3, 'fromId': '!a84ae0c0', 'toId': '!a84ae01c'}, '
  lastHeard': None, 'hopLimit': 3} { 'num': 2823478980, 'user': { 'id': '!
  a84adec4', 'longName': 'mesh 1', 'shortName': 'm1', 'macaddr': '84:cc:a8:4a:de
  :c4', 'hwModel': 'PRIVATE_HW'}, 'position': {}, 'snr': 14.5, 'lastReceived':
  { 'from': 2823478980, 'to': 2823479324, 'id': 1087849284, 'rxSnr': 14.5, '
  hopLimit': 3, 'fromId': '!a84adec4', 'toId': '!a84ae01c'}, 'lastHeard': None,
  'hopLimit': 3}
7
8 Preferences: { "phoneTimeoutSecs": 900, "lsSecs": 300, "region": "ANZ" }
9
10 Channels:
11 PRIMARY psk=default { "modemConfig": "Bw125Cr48Sf4096", "psk": "AQ==", "name":
  "UFES" }
12
13 Primary channel URL: https://www.meshtastic.org/d/#CgsYAyIBASoEVUZFUw

```

Registro de Saída B.4 – Configuração final do Mesh 2

```

1 $ meshtastic --port /dev/ttyUSB0 --info
2 Connected to radio
3
4 Owner: mesh 2 (m2)
5 My info: { "myNodeNum": 2823479488, "numBands": 20, "firmwareVersion": "1.2.64",
  "rebootCount": 186, "bitrate": 17.08847, "messageTimeoutMsec": 300000, "
  minAppVersion": 20200, "maxChannels": 8, "hasWifi": true, "airUtilTx":
  0.20797223 }
6 Nodes in mesh: { 'num': 2823479488, 'user': { 'id': '!a84ae0c0', 'longName': 'mesh
  2', 'shortName': 'm2', 'macaddr': '84:cc:a8:4a:e0:c0', 'hwModel': 'PRIVATE_HW
  '}, 'position': { 'latitudeI': -202736021, 'longitudeI': -403060778, 'latitude
  ': -20.273602099999998, 'longitude': -40.3060778}, 'snr': 11.75} { 'num':
  2823479324, 'user': { 'id': '!a84ae01c', 'longName': 'mesh 3', 'shortName': 'm3
  ', 'macaddr': '84:cc:a8:4a:e0:1c', 'hwModel': 'PRIVATE_HW'}, 'position': {}, '
  lastHeard': 1654227905, 'snr': 10.0} { 'num': 2823478980, 'user': { 'id': '!
  a84adec4', 'longName': 'mesh 1', 'shortName': 'm1', 'macaddr': '84:cc:a8:4a:de
  :c4', 'hwModel': 'PRIVATE_HW'}, 'position': {}, 'lastHeard': 1655095012, 'snr
  ': 9.25}
7
8 Preferences: { "phoneTimeoutSecs": 900, "lsSecs": 300, "wifiSsid": "matheus_bozzi
  ", "wifiPassword": "sekrit", "region": "ANZ", "fixedPosition": true, "
  rangeTestPluginEnabled": true, "rangeTestPluginSave": true }
9
10 Channels:
11 PRIMARY psk=default { "modemConfig": "Bw125Cr48Sf4096", "psk": "AQ==", "name":

```

```

12         "UFES" }
13 Primary channel URL: https://www.meshtastic.org/d/#CgsYAYIBASoEVUZFUw

```

Registro de Saída B.5 – Configuração final do Mesh 1

```

1 $ meshtastic --port /dev/ttyUSB0 --info
2 Connected to radio
3
4 Owner: mesh 1 (m1)
5 My info: { "myNodeNum": 2823478980, "numBands": 20, "firmwareVersion": "1.2.64",
  "rebootCount": 196, "bitrate": 17.08847, "messageTimeoutMsec": 300000, "
  minAppVersion": 20200, "maxChannels": 8, "hasWifi": true }
6 Nodes in mesh: { 'num': 2823478980, 'user': { 'id': '!a84adec4', 'longName': 'mesh
  1', 'shortName': 'm1', 'macaddr': '84:cc:a8:4a:de:c4', 'hwModel': 'PRIVATE_HW
  '}, 'position': {}, 'snr': 10.75} { 'num': 2823479324, 'user': { 'id': '!
  a84ae01c', 'longName': 'mesh 3', 'shortName': 'm3', 'macaddr': '84:cc:a8:4a:e0
  :1c', 'hwModel': 'PRIVATE_HW'}, 'position': {}, 'lastHeard': 1654224268, 'snr
  ': 8.75} { 'num': 2823479488, 'user': { 'id': '!a84ae0c0', 'longName': 'mesh
  2', 'shortName': 'm2', 'macaddr': '84:cc:a8:4a:e0:c0', 'hwModel': 'PRIVATE_HW
  '}, 'position': { 'latitudeI': -202736021, 'longitudeI': -403060778, 'latitude
  ': -20.273602099999998, 'longitude': -40.3060778}, 'lastHeard': 1656138951, '
  snr': 11.25}
7
8 Preferences: { "phoneTimeoutSecs": 900, "lsSecs": 300, "region": "ANZ"}
9
10 Channels:
11 PRIMARY psk=default { "modemConfig": "Bw125Cr48Sf4096", "psk": "AQ==", "name":
  "UFES" }
12
13 Primary channel URL: https://www.meshtastic.org/d/#CgsYAYIBASoEVUZFUw

```

Registro de Saída B.6 – Configuração final do Mesh 3

```

1 $ meshtastic --port /dev/ttyUSB0 --info
2 Connected to radio
3
4 Owner: mesh 3 (m3)
5 My info: { "myNodeNum": 2823479324, "numBands": 20, "firmwareVersion": "1.2.64",
  "rebootCount": 314, "bitrate": 17.08847, "messageTimeoutMsec": 300000, "
  minAppVersion": 20200, "maxChannels": 8, "hasWifi": true }
6 Nodes in mesh: { 'num': 2823479324, 'user': { 'id': '!a84ae01c', 'longName': 'mesh
  3', 'shortName': 'm3', 'macaddr': '84:cc:a8:4a:e0:1c', 'hwModel': 'PRIVATE_HW
  '}, 'position': {}, 'lastHeard': 1654227905, 'snr': 10.0} { 'num': 2823479488,
  'user': { 'id': '!a84ae0c0', 'longName': 'mesh 2', 'shortName': 'm2', 'macaddr
  ': '84:cc:a8:4a:e0:c0', 'hwModel': 'PRIVATE_HW'}, 'position': { 'latitudeI':
  -202736021, 'longitudeI': -403060778, 'latitude': -20.273602099999998, '
  longitude': -40.3060778}, 'snr': 11.75} { 'num': 2823478980, 'user': { 'id': '!
  a84adec4', 'longName': 'mesh 1', 'shortName': 'm1', 'macaddr': '84:cc:a8:4a:de
  :c4', 'hwModel': 'PRIVATE_HW'}, 'position': {}, 'lastHeard': 1655095012, 'snr
  ': 9.25}
7
8 Preferences: { "phoneTimeoutSecs": 900, "lsSecs": 300, "region": "ANZ"}
9
10 Channels:

```

-
- 11 PRIMARY psk=default { "modemConfig": "Bw125Cr48Sf4096", "psk": "AQ==", "name":
"UFES" }
 - 12
 - 13 Primary channel URL: <https://www.meshtastic.org/d/#CgsYAyIBASoEVUZFUw>