UNIVERSIDADE DO VALE DO TAQUARI
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

PROTOCOLO DE COMUNICAÇÃO DE TOPOLOGIA *MESH* UTILIZANDO RADIOFREQUÊNCIA POR MODULAÇÃO LORA

Matheus George Abel

Lajeado, novembro de 2019

Matheus George Abel

PROTOCOLO DE COMUNICAÇÃO DE TOPOLOGIA *MESH* UTILIZANDO RADIOFREQUÊNCIA POR MODULAÇÃO LORA

Monografia apresentada na disciplina de Trabalho de Conclusão de Curso II, da Universidade do Vale do Taquari – Univates, como parte da exigência para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Me. Anderson Antônio Giacomolli.

Lajeado, novembro de 2019.

RESUMO

O avanço na quantidade de soluções IoT é inumerável e estes dispositivos demandam diversas áreas do desenvolvimento tecnológico, sendo o tipo de comunicação que oferecem um dos pilares de sua concepção. Conciliar o baixo custo energético e a distância de propagação são os desafios no desenvolvimento de novos dispositivos IoT e cada vez mais tecnologias que aliam estas características são alvo de pesquisas, como o LoRa. A Associação Brasileira de Internet das Coisas (ABINC) aponta que o setor de conectividade é o que mais deve crescer e receber incentivos financeiros nos próximos anos. A fim de aprimorar o alcance que um ponto de comunicação LoRa oferece, propõe-se o desenvolvimento de um protocolo de camada de rede de topologia lógica tipo *mesh*, que deve ser capaz de encaminhar mensagens através de saltos entre pontos conhecidos da rede, de forma com que todo e qualquer nó da rede torne-se visível em qualquer ponto. Com este protocolo pretende-se atingir distâncias da ordem de dezenas de quilômetros, com o intuito de tornar possível a realização do monitoramento de variáveis antes limitadas ao tipo de conexão disponível no local de medição.

Palavras-chave: LoRa. Rede Mesh. Protocolo de comunicação. IoT.

ABSTRACT

The grow of the quantity of IoT solutions is innumerable, these devices demands several areas of technological development, the kind of communication that they provide is one of their cores. Achieving low energy cost and long distances is a challenge in new IoT devices development and technologies that allies this, such as LoRa, are objects of researches. ABINC (Brazillian IoT Association) says the connectivity sector is the one which should grow up and receive more investments in the next years. In order to improve a point-to-point LoRa communication, it is proposed to develop a mesh logical topology protocol that must be able to forward messages through jumps between known network points so that any and every node of the network could be reached. With this protocol it is intended to reach distances of the order of tens of kilometers, in order to make possible the monitoring of variables previously limited to the type of connection available at the measurement site.

Keywords: LoRa. Mesh network. Communication protocol. IoT.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Um conceito importante tem ganhado força nos últimos anos na área tecnológica: a *Internet of Things* (IoT), em português Internet das Coisas, é um paradigma para objetos que são capazes de sentir, identificar, conectar e se comunicarem-se uns com os outros através da internet. Um objetivo da IoT é oferecer soluções capazes de transformar o modo com que muitos sistemas interagem e geram informações, entre eles, os sistemas de manufatura, transporte e infraestrutura. (WHITMORE; AGARWAL; DA XU, 2015; XU; HE; LI, 2014)

Inicialmente, o conceito de IoT foi utilizado apenas para se referir a objetos interoperáveis identificados por rádio frequência (RFID). Atualmente, Xu, He e Li (2014) definem IoT como:

> "Uma infraestrutura de rede global dinâmica com recursos de autoconfiguração baseados em protocolos de comunicação padronizados e interoperáveis onde as "coisas" físicas e virtuais têm identidades, atributos físicos e personalidades virtuais, usam interfaces inteligentes e são perfeitamente integrados à rede de informações".

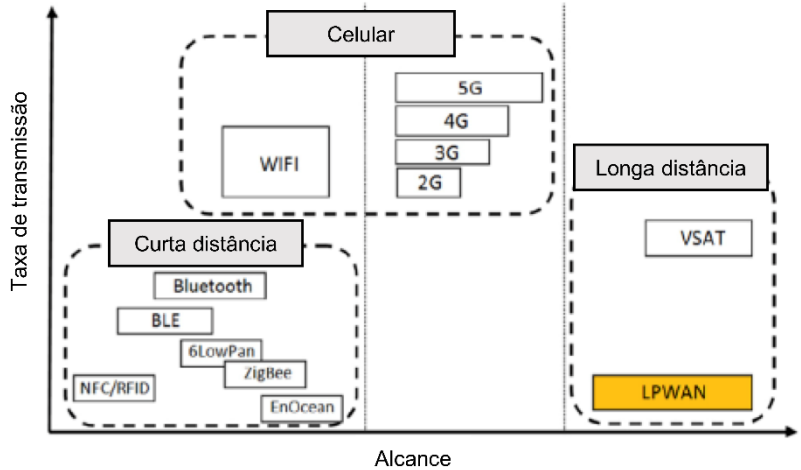
De uma forma menos conceitual, existem inúmeras possibilidades de aplicação, incluindo segurança, rastreamento de cargas ou objetos, agricultura, medições inteligentes, *smart cities* (cidades inteligentes) e *smart homes* (casas inteligentes) e até serviços de saúde que se tornaram possíveis graças ao surgimento e barateamento de sensores, atuadores e sistemas GPS (MEKKI et al., 2019; WHITMORE; AGARWAL; DA XU, 2015).

Estima-se que o mercado mundial de IoT, seja ele de *software*, de serviços, de conectividade ou de dispositivos, atingirá uma movimentação de 318 milhões de dólares até 2023 com crescimento a uma taxa anual de 20 %. A Associação Brasileira de Internet das Coisas (ABINC) ainda menciona que embora *software* e serviços liderem o setor, a conectividade tem a maior tendência de crescimento de receitas empregadas no mesmo período supracitado (ABINC, 2018).

Há diversos requisitos para uma aplicação IoT tais como conexão de longo alcance, baixo consumo de dados, baixo consumo de energia e, principalmente, custo-benefício em relação aos seus atributos. Diante disto, tecnologias como *Zigbee* e *Bluetooth* não são capazes de realizar comunicação de longo alcance, por exemplo. Por outro lado, as tecnologias celulares como 2G, 3G ou 4G atendem a demanda por uma maior cobertura, mas deixam a desejar no quesito consumo de energia (MEKKI et al., 2019).

A partir disso, tecnologias emergentes de comunicação como as redes de grande alcance com baixo consumo de energia (LPWAN) têm surgido e recebido reconhecimento pela comunidade industrial e pesquisadora. Pode-se alcançar distâncias na ordem de quilômetros de transmissão de dados com uma duração da bateria de vários anos, a depender das características do projeto. Outro aspecto positivo é que as tecnologias LPWAN existentes hoje, como LoRa, Sigfox e NB-IoT operam em frequências não licenciadas e, portanto, barateiam o projeto, pois dispensam homologação dos órgãos competentes (MEKKI et al., 2019).

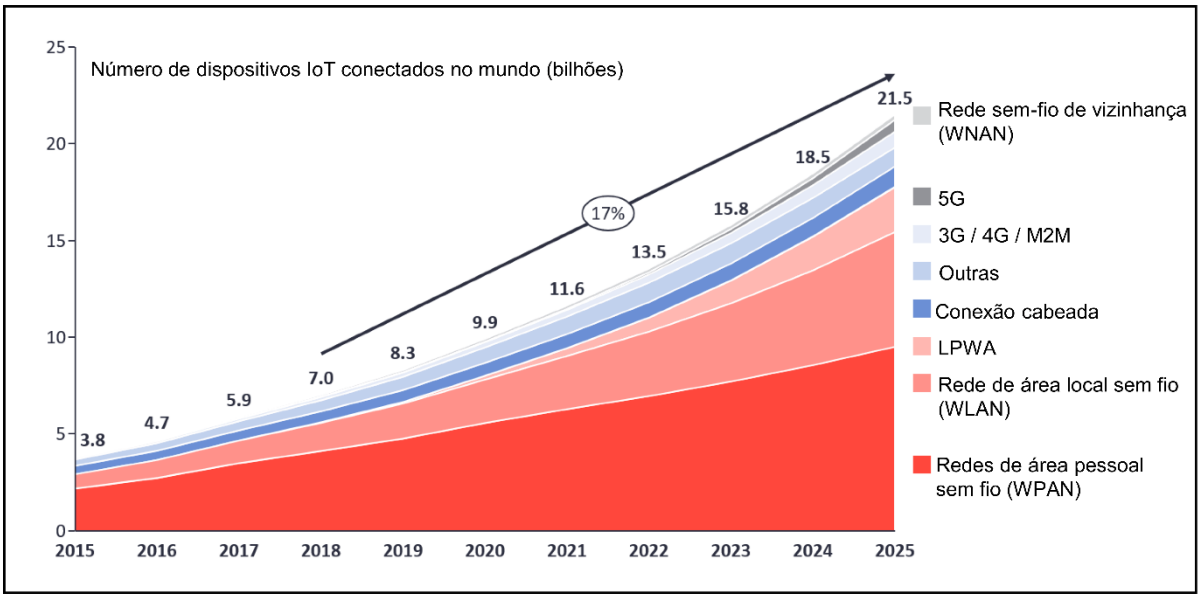
Figura 1 – Tecnologias de transmissão sem fio



Fonte: Adaptado pelo autor de MEKKI et al. (2019).

Espera-se que no ano de 2025 existam cerca de 2 bilhões de dispositivos no planeta operando em redes do tipo LPWAN (LUETH, 2018), desta forma, destaca-se a importância que este tipo de comunicação tem e terá ao longo dos anos na Figura 2.

Figura 2 - Número de dispositivos IoT conectados no mundo



Fonte: Adaptado pelo autor de LUETH (2018).

Diante dos cenários apresentados é nítida a necessidade e a possibilidade de avanço no quesito conectividade dos dispositivos e dos serviços IoT, principalmente em aplicações que demandem uma conectividade confiável e de longo alcance.

1.1 Tema

Projeto de um protocolo de comunicação de camada de rede baseado em topologia *mesh,* sob a camada física e de enlace do protocolo de comunicação LoRa.

1.1.1 Delimitação do tema

Desenvolvimento de um sistema que incluí uma placa de circuito impresso de testes e um *software* que seja capaz de construir *links* de comunicação entre dispositivos embarcados com comunicação LoRa numa mesma rede, a fim de encaminhar mensagens entre vários pontos de conexão da rede. Desta forma,

estabelecer comunicação sem fio sem que seja necessário o alcance direto entre a origem e o destino da mensagem.

1.2 Objetivo Geral

O propósito geral deste projeto será desenvolver um *software* rodando em um microcontrolador com comunicação LoRa, visando o encaminhamento de mensagens entre nós distintos a fim de estabelecer uma topologia de rede lógica do tipo *mesh* para que seja possível alcançar maiores distâncias de comunicação em relação a um enlace LoRa ponto a ponto. Para os testes, também faz-se necessário a prototipagem de uma placa de circuito impresso que integre os módulos para comunicação LoRa.

1.2.1 Objetivos Específicos

- Projetar uma placa de circuito impresso (PCB) que integre uma alimentação, um microcontrolador e um rádio do tipo LoRa;
- Definir o conjunto de regras de comunicação entre os dispositivos;
- Desenvolver uma rotina de criação de uma rede de dispositivos LoRa por meio de regras anteriormente definidas, desta forma, cada novo dispositivo que contenha esta rotina, deverá se incluir na rede já existente ou iniciar uma nova caso não exista rede conhecida;
- Desenvolver uma rotina para que mensagens sejam encaminhadas corretamente através dos pontos da rede criada no item anterior;
- Programar o microcontrolador para exibir informações relevantes da rede e de mensagens em um *display* acoplado, para que desta forma seja possível validar o funcionamento do projeto.

1.3 Justificativa

IoT é uma tendência mundial no desenvolvimento tecnológico e, atualmente, um dos maiores focos de estudos desta área está voltado para serviços e aplicações. Este tipo de pesquisa pode encontrar dificuldades devido aos protocolos de comunicação limitados no quesito distância de transferência de informações.

Uma vez que novas aplicações e serviços tendem a surgir nas mais diversas áreas, tais como agricultura e monitoramento em áreas de difícil acesso, é importante que haja comunicação de baixo custo e consumo energético, sem que seja necessária a implantação de itens de infraestrutura.

A partir disso, propõe-se o desenvolvimento de um protocolo acima da camada de enlace e física da comunicação LoRa para que os dispositivos possam comunicarem-se e também encaminhar mensagens que não sejam destinadas a eles mesmos. A escolha do LoRa está aliada, principalmente, ao longo alcance e ao baixo consumo de energia, apesar da comunicação não permitir o envio de dados muito grandes, o que atende as aplicações supracitadas.

2 REFERENCIAL BIBLIOGRÁFICO

2.1 Visão Geral

Dentre as diversas topologias de rede existentes, a Internet é, com certeza, a mais difundida e mais utilizada pela população (KUROSE; ROSS, 2013). Conforme Kurose (2012, p. 3), novos dispositivos estão fazendo o uso da internet para prover serviços para os consumidores, não mais apenas computadores e *laptops*, como no período de início de popularização da rede.

Para que a Internet funcione da maneira como a conhecemos, sua organização é fundamental e, portanto, a utilização de um protocolo único de comunicação é indispensável (KUROSE; ROSS, 2013). Segundo o autor, *Transmission Control Protocol* (TCP - Protocolo de Controle de Transmissão) e *Internet Protocol* (IP - Protocolo de Internet), são os dois protocolos mais importantes da internet. Em suma, o protocolo IP “especifica o formato dos pacotes que são enviados e recebidos entre roteadores e sistemas finais” (KUROSE; ROSS, 2013, p. 4).

Tamanha difusão da Internet só foi possível, conforme Kurose e Ross (2013, p. 4), em função da padronização redigida pela *Internet Engineering Task Force* (Força de Trabalho de Engenharia da Internet). São documentos extremamente técnicos e acurados que objetivam a definição de protocolos como TCP, IP e HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto). Ainda conforme o autor, outros órgãos também surgiram com a intenção de especificar padrões para componentes da rede, um exemplo é o *IEEE 802 LAN/MAN Standarts*

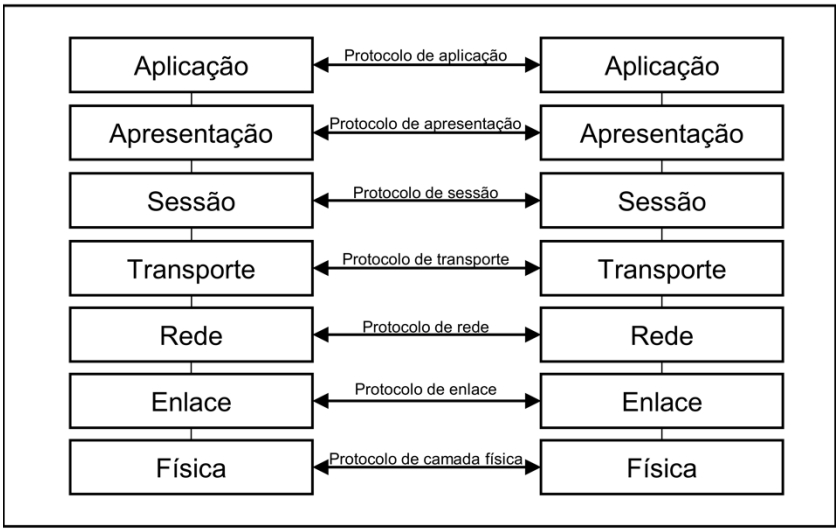
Committee (Comitê de Padrões LAN/MAN IEEE 802), que especifica os padrões Ethernet e Wi-Fi sem fio.

Kurose e Ross (2013, p. 6) comentam que um protocolo de rede pode ser comparado com um protocolo humano. Por exemplo, para perguntar a hora para outra pessoa, inicia-se o diálogo com uma saudação e dependendo da resposta a pergunta é feita ou não. Este procedimento é intrínseco a nossa comunicação e, no caso de comunicação entre dois dispositivos qualquer, o conjunto de regras deve ser definido previamente. Conforme Kurose e Ross (2013, p. 7), “um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento.”.

2.2 Modelo de Referência OSI

Lima Filho (2015, p. 16) explica que o modelo de referência OSI foi elaborado a partir de um projeto da *International Organization for Standarization* (ISO – Organização Internacional para Padronização) dos anos 70 e amplamente adotado pelas empresas de tecnologia nos anos 90, ao passo que o modelo TCP/IP também adquiriu boa aceitação. O modelo é composto por camadas e tem como objetivo fundamentar e organizar o melhor de cada padrão utilizado até então (LIMA FILHO,

Figura 3 – Camadas do modelo OSI



Fonte: Adaptado pelo autor com base em Lima Filho (2015, p. 16).

2015). A organização é feita de forma hierárquica em 7 camadas, demonstradas na Figura 3.

O autor destaca que o modelo OSI não é um protocolo, “[...] mas um conjunto de regras que classificam e normatizam o uso dos inúmeros protocolos existentes dentro das referidas camadas [...]”.

No entendimento de Lima Filho (2015, p. 18), as sete camadas podem ser resumidas como segue: A camada de aplicação, deve ser entendida como a camada que interage diretamente com o usuário. A camada de apresentação é definida pelo autor como a encarregada por adequar os dados recebidos, como por exemplo a conversão *Unicode Transformation Format* (UTF - Formato de Transformação Unicode) ou, caso haja criptografia, esta é a camada responsável pela quebra da mesma. A camada de sessão estabelece a conexão, transmite os dados e finaliza a conexão, mas também é responsável por separar dados de aplicações diferentes. A quarta camada, de transporte, é incumbida de estabelecer os fluxos de dados, sem que haja mistura entre as aplicações; Para isso, geralmente ocorre segmentação de dados juntamente com o gerenciamento da segmentação, no caso de existência, é neste nível que ocorre a verificação de entrega de mensagens. A camada de rede tem como tarefa a execução do transporte dos pacotes entre as extremidades e, para isso, utiliza-se de endereçamento (tanto físico quanto lógico), de encapsulamento (cabeçalhos), de roteamento (escolha do melhor caminho da mensagem) e de desencapsulamento (após a chegada correta de um pacote ao destino correto, dispõe a mensagem para as camadas superiores). A segunda camada, de enlace de dados, é compreendida na literatura como a camada que é responsável por promover a ligação entre dois pontos, detectar e corrigir erros, prover entrega confiável e controlar a compatibilidade entre origem e destino. A camada mais profunda do modelo OSI define o método que os bits envolvidos na comunicação serão manobrados fisicamente, ou seja, qual o tipo de modulação será utilizada, como a informação digital irá se propagar pelo meio físico (ar, cobre ou luz).

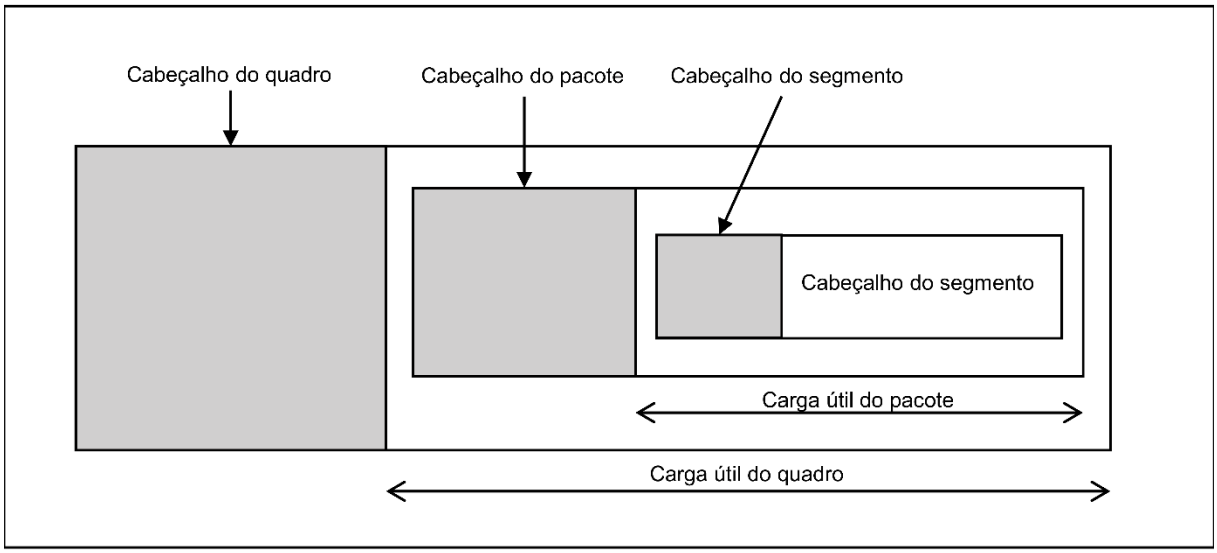
Por se tratar de um projeto substancialmente de camada de rede, as camadas superiores de sessão, apresentação e aplicação não serão revisadas, dando ênfase nas camadas de menor nível nas próximas seções.

2.2.1 Camada de transporte

A camada de transporte está localizada, no modelo OSI, entre a camada de sessão e rede. Este nível do modelo tem como principal característica fornecer um protocolo capaz de garantir a comunicação lógica entre diferentes aplicações. Garantia de comunicação lógica, neste contexto, significa organizar a comunicação entre dois processos de maneira que eles compreendam a comunicação como se estivessem conectados diretamente, assim explicam Kurose e Ross (2015).

> No lado remetente, a camada de transporte converte as mensagens que recebe de um processo de aplicação remetente em pacotes de camada de transporte, denominados segmentos de camada de transporte. [...] Isso é (possivelmente) feito fragmentando-se as mensagens da aplicação em pedaços menores e adicionando-se um cabeçalho de camada de transporte a cada pedaço para criar o segmento. [...] (KUROSE; ROSS, 2015, p. 136).

Figura 4 – Alinhamento de pacotes e quadros



Fonte: Adaptado pelo autor com base em Kurose e Ross (2013, p. 137).

Kurose e Ross (2013, p. 137) explicam que na Internet, por exemplo, existem dois protocolos de camada de transporte, o TCP e UDP, cada um oferecendo um conjunto diferente de serviços para as camadas superiores.

Há uma relação muito estreita entre a camada de transporte e a de rede, sendo que os autores definem que a primeira fornece comunicação lógica entre aplicações, enquanto que a segunda provê comunicação lógica entre hospedeiros (*hardware*). Para entender a diferença, pode-se utilizar uma analogia de um

condomínio. O porteiro é responsável por receber todas as correspondências do correio e distribuí-las para as unidades correspondentes. Pode-se dizer então que os correios realizam o trabalho do protocolo de camada de rede, entregando todas as correspondências a um hospedeiro (condomínio), e o porteiro realiza o trabalho do protocolo da camada de transporte, uma vez que seu redirecionamento se restringe aos apartamentos (processos) e não é necessário o conhecimento da infraestrutura dos correios (camada de rede) para tanto (KUROSE; ROSS, 2013).

Uma parte importante da camada de transporte é o processo de multiplexação e demultiplexação. O processo de demultiplexação ocorre no recebimento de um pacote e consiste em entregar os dados contidos a outra entidade, denominada *socket* (atua como um intermediário entre camada de transporte e processo, como uma porta de entrada e saída), através da interpretação do cabeçalho. Já a multiplexação reúne no hospedeiro de origem dados provenientes de diversos *sockets* e adiciona o cabeçalho criando, assim, um segmento com informações suficientes para ser encaminhado a camada de rede. Ainda utilizando a analogia apresentada no parágrafo anterior, o processo de demultiplexação é a identificação por parte do porteiro a qual processo pertence cada carta, a partir do número do apartamento (informação de cabeçalho) (KUROSE; ROSS, 2013).

2.2.2 Camada de rede

O papel da camada de rede é, de forma geral e simplória, transportar um pacote entre dois pontos. Porém, ao aprofundar a análise, torna-se uma das camadas mais complexas de todo o modelo RM-OSI e, conforme Kurose e Ross (2013, p. 224), tal complexidade dá-se por duas funções importantes do processo de comunicação, denominadas repasse e roteamento.

O repasse, de acordo com Kurose e Ross (2013, p. 225), é a condução de um pacote do enlace de entrada de um roteador até o enlace de saída do mesmo. Um pacote que chega a um roteador R proveniente de um hospedeiro A deve ser entregue a um hospedeiro B, mas, antes disso, deve ser repassado a outro roteador R2.

Roteamento é a determinação de um caminho pelo qual o pacote deve fluir do ponto de origem até o ponto de destino. Esta tarefa é realizada a partir de algoritmos que calculam os melhores caminhos. Define-se este componente da camada como algoritmos de roteamento (KUROSE; ROSS, 2013).

Os especialistas realizam em sua obra uma comparação interessante para melhor entendimento dos conceitos de repasse e roteamento:

> Para usar uma viagem como analogia, voltemos àquele nosso viajante [...] que vai da Pensilvânia à Flórida. Durante a viagem, nosso motorista passa por muitos cruzamentos de rodovias em sua rota. Podemos imaginar o repasse como o processo de passar por um único cruzamento: um carro chega ao cruzamento vindo de uma rodovia e determina qual rodovia ele deve pegar para sair do cruzamento. Podemos imaginar o roteamento como o processo de planejamento da viagem da Pensilvânia até a Flórida: antes de partir, o motorista consultou um mapa e escolheu um dos muitos caminhos possíveis. Cada um deles consiste em uma série de trechos de rodovias conectados por cruzamentos. (KUROSE; ROSS, 2013, p. 226)

2.2.2.1 Repasse

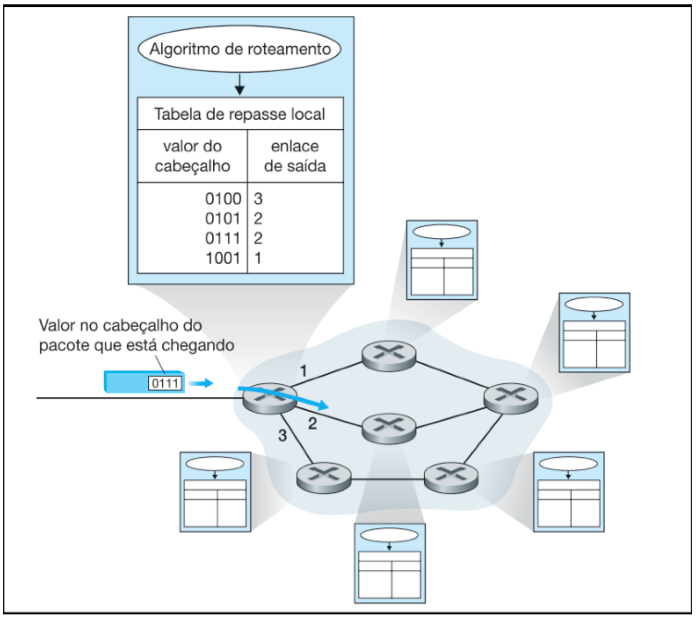
Levando em consideração a discussão do item anterior fica evidente a importância da tabela de repasse no contexto de camada de rede. Um pacote que chega a um roteador é repassado para o próximo ponto da rede baseado na interpretação do cabeçalho e na tabela de repasse daquele dispositivo.

Um ponto importante de um protocolo de rede é a definição do modelo de serviço de rede. Kurose e Ross (2013) indicam que o modelo de serviço “define as características do transporte de dados fim a fim entre uma borda da rede e a outra, isto é, entre sistemas finais remetente e destinatário”. Alguns modelos que podem ser empregados em uma rede são: (1) entrega garantida, assegurando que os pacotes são entregues; (2) entrega garantida com atraso limitado, igual ao anterior, porém com tempo limite de envio; (3) entrega de pacotes na ordem, garantia de que pacotes cheguem ordenadamente ao destino e (4) serviços de segurança, criptografia entre dois pontos através de *token* de sessão.

2.2.2.2 Roteamento

O roteamento é um dos pontos mais importantes de um protocolo de camada de rede e sua função é determinar um caminho entre origem e destino através dos pontos disponíveis em uma rede (KUROSE; ROSS, 2013). Para que um bom caminho seja encontrado o algoritmo de roteamento é responsável por determinar este caminho ao “menor custo”, geralmente associado ao caminho mais curto.

Figura 5 – Exemplo de roteamento



Fonte: Kurose e Ross (2013, p. 250).

Em algoritmos de roteamento, Kurose e Rosse (2013) definem um grafo como um termo central para formular problemas de roteamento. Um grafo é um conjunto de nós e uma coleção de arestas, sendo que cada uma destas arestas é um par de nós do conjunto de nós. Um roteador é entendido como um nó de um grafo, dispositivo em que são tomadas decisões de repasse, já uma aresta conectora é o enlace físico entre dois pontos. É de praxe eleger, apontam os autores, um custo para cada aresta do grafo e, geralmente, esta medida é essencial para o algoritmo de roteamento realizar o cálculo de uma rota.

Segundo Kurose e Ross (2013) é possível classificar algoritmos de roteamento em dois principais grupos: de roteamento global ou descentralizados. O primeiro deve ter uma visão geral da rede e, então, calcular o caminho de menor custo a partir do panorama previamente encontrado, ou seja, um algoritmo de roteamento global tem

informação completa a respeito de conectividade e custos. Em um algoritmo de roteamento descentralizado o cálculo de custo é realizado individualmente em cada um dos nós da rede. Neste caso nenhum nó tem informação completa sobre a rede, mas sim a respeito dos seus vizinhos de enlace. Um algoritmo descentralizado realiza um cálculo iterativo realizando troca de informações entre os vizinhos.

2.2.3 Camada de enlace

Previamente a uma explicação mais específica em relação a esta camada, é interessante dar significado a palavra enlace, no contexto de rede. Enlace pode ser definido como a ligação entre dois pontos (nós) da rede, independente do meio físico ser cabeado ou não. A camada de enlace é a ponte entre a camada de rede e a camada física propriamente dita e nela são formados envelopes de dados chamados de frames (KUROSE; ROSS, 2013).

Kurose e Ross (2013, p. 323) utilizam uma comparação que simplifica o entendimento desta camada:

> Imagine um agente de viagens que planeja uma viagem para um turista de Princeton, em Nova Jersey, até Lausanne, na Suíça. O agente decide que é mais conveniente para o turista pegar uma limusine de Princeton até o aeroporto JFK, em seguida um avião até o aeroporto de Genebra e, por fim, um trem até a estação ferroviária de Lausanne. Assim que o agente fizer as três reservas, é responsabilidade da empresa de limusines conduzir o turista de Princeton ao aeroporto JFK; é responsabilidade da companhia aérea transportar o turista do aeroporto JFK a Genebra; e é responsabilidade do trem suíço levar o turista de Genebra a Lausanne.

Pode-se interpretar o trecho do texto reproduzido afirmando que cada segmento da viagem é direto entre dois pontos próximos, outro ponto importante é perceber que cada trecho da viagem não é operado pela mesma empresa ou pelo mesmo meio (estrada, trilho ou ar). Então, neste caso compara-se o turista com um *frame*, cada trecho com um enlace, o meio de transporte com o protocolo da camada de enlace e o agente de viagens com o protocolo de roteamento (KUROSE; ROSS, 2013).

Dentre os serviços que a camada de enlace fornece à comunicação, quatro são de maior relevância. O enquadramento de dados é um deles e, nada mais é do que a

adição de um cabeçalho especificado pelo protocolo de enlace; Este processo cria o *frame* propriamente dito, que conterá informações relevantes do encaminhamento pelo enlace em conjunto com todo o resto da mensagem. O acesso ao enlace também é oferecido e nele é implementado um protocolo de controle de acesso ao meio (MAC - *Medium Access Control*) que determina regras de transmissão e recepção dos *frames*. Em conjunto, a garantia de entrega também é um componente de responsabilidade da camada de enlace e, geralmente, é utilizado apenas quando o enlace tem alta taxa de erro, como por exemplo nas redes sem fio; no caso de fibras ópticas ou cabos coaxiais este serviço pode ser custoso para a rede. É possível também que a camada de enlace preste o serviço de detecção e correção de erros, executada geralmente por *hardware,* e consiste na obrigação do transmissor em enviar *bits* de detecção de erro e na verificação destes pelo receptor (KUROSE; ROSS, 2013).

2.3 Topologias de rede

Quanto a organização das redes, pode-se dividir a topologia em lógica e física. Organizar a rede utilizando topologias conhecidas têm como principal meta a facilidade na visualização do conjunto da rede, visando seu melhor projeto (Intersaberes, 2014).

2.3.1 Topologias lógicas

Topologia lógica diz respeito à maneira na qual os dados fluem ao longo da rede entre origem e destino. Duas topologias lógicas são relevantes e descritas a seguir.

O padrão *token* de passagem pode ser relacionado diretamente com uma corrida de revezamento, na qual somente o atleta que possui o bastão pode correr. Na topologia lógica *token*, o dispositivo somente transmite nos momentos em que possui o *token* da rede, caso contrário apenas “escuta” a rede e permanece no aguardo (Intersaberes, 2014). A International Business Machines (IBM) desenvolveu

a arquitetura *token ring* que, em meados dos anos 80, alcançava 16 Mbps através de cabos coaxiais ou par trançado. Em resumo, *token* deve ser entendido como a permissão para um nó da rede acessar o meio físico e transmitir informações na rede; Essa permissão vai sendo passada adiante e somente um nó da rede a possui, o que faz esta topologia ser livre de colisões (Intersaberes, 2014).

O padrão lógico *broadcast* (transmissão, em tradução livre) é, hoje, o padrão mais utilizado no mundo. Seu princípio é muito simples: o envio da informação é feito a todos os nós da rede de uma só vez (Intersaberes, 2014).

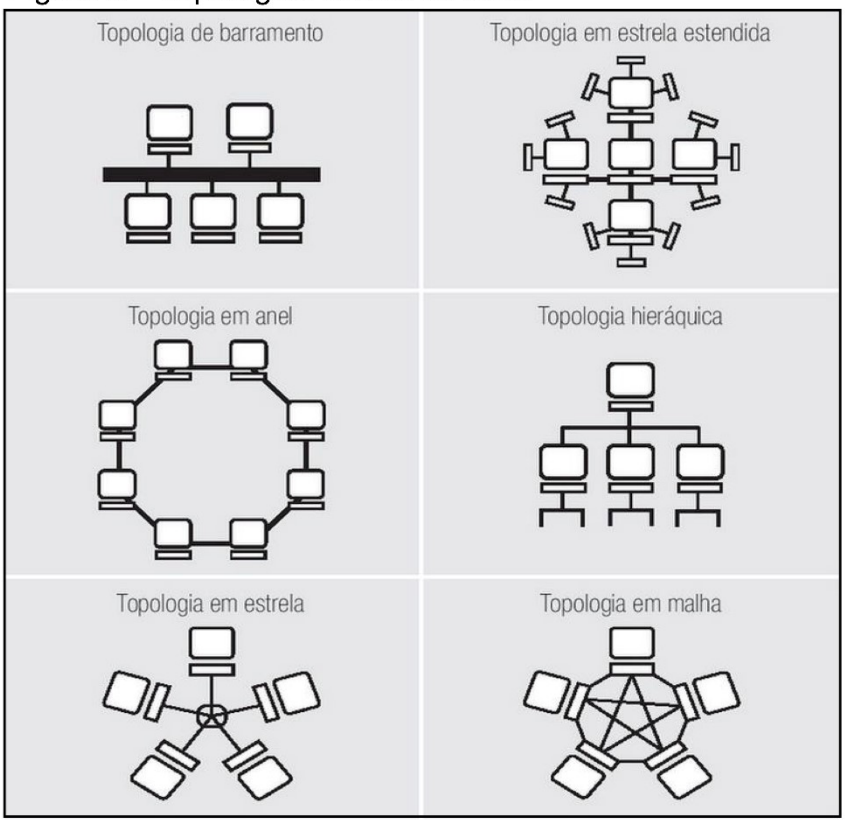
2.3.2 Topologias físicas

Como o próprio nome remete, a topologia física é como os pontos da rede são conectados entre si. Pode-se classificar a topologia de rede em 4 principais grupos: (1) O barramento é um deles e trata-se de um único caminho que chega a todos os nós da rede e, dessa forma, apenas uma informação pode ser enviada por vez; (2) A topologia em anel liga todos os nós em um *loop*, dessa forma, a informação vai passando por todos os dispositivos da rede; (3) A organização estrela é amplamente utilizada e sua característica majoritaria é a existência de um dispositivo concentrador, como um *switch*, no qual todos os nós são conectados. É a topologia usada nas redes LANs; (4) A topologia física em malha[1] é pouco usada comercialmente por ser complexa, já que todos os pontos têm conexão com todos os outros. A forma desta rede pode ser considerada muito confiável se implementada corretamente (Intersaberes, 2014).

A Figura 6 representa, ainda, as topologia estrela estendida que também pode ser entendida como uma “estrela de estrelas” ou várias estrelas e a hierárquica, que são minimamente aplicadas (Intersaberes, 2014).

[1] Ao longo do texto a expressão “malha” deve ser interpretata também como “*mesh*”.

Figura 6 – Topologias físicas de rede



Fonte: Intersaberes (2014, p. 174).

2.4 Padrão IEEE 802

Este conjunto de especificações abertas têm como objetivo normatizar as redes locais e metropolitanas, principalmente nas camadas físicas e de enlace do modelo OSI. A norma IEEE (*Institute of Electrical and Electronics Engineers* - Instituto de Engenheiros Elétricos e Eletrônicos) 802 surgiu a partir de um comitê chamado LMSC (*Lan/Man Standarts Committee* - Comitê dos Padrões LAN/MAN), que dividiu o trabalho a ser feito em diversos grupos de trabalho. A IEEE 802 é, hoje, utilizada nos mais diversos protocolos de comunicação sendo o principal deles o Ethernet em conjunto com a *Wireless LAN*. (KUROSE; ROSS, 2013)

Existem mais de 20 variações deste padrão, cada um com um objetivo específico: 802.15.1 especifica o *Bluetooth*, 802.15.4 especifica *ZigBee* e *WirelessHART*, 802.11 específica a *Wi-Fi*, 802.3 é a especificação do *Ethernet*. Pode-se notar que a família IEEE 802 não se limita ao meio físico guiado (par trançado, fibra óptica e cabo coaxial), mas também à comunicações sem fio, não guiadas, como

tecnologias de transmissão de dados por radiofrequência em diversos níveis de frequência, na base de sub até *gigahertz*, tornando-a uma norma bastante abrangente (KUROSE; ROSS, 2013).

2.4.1 Redes de área pessoal sem fio (WPAN)

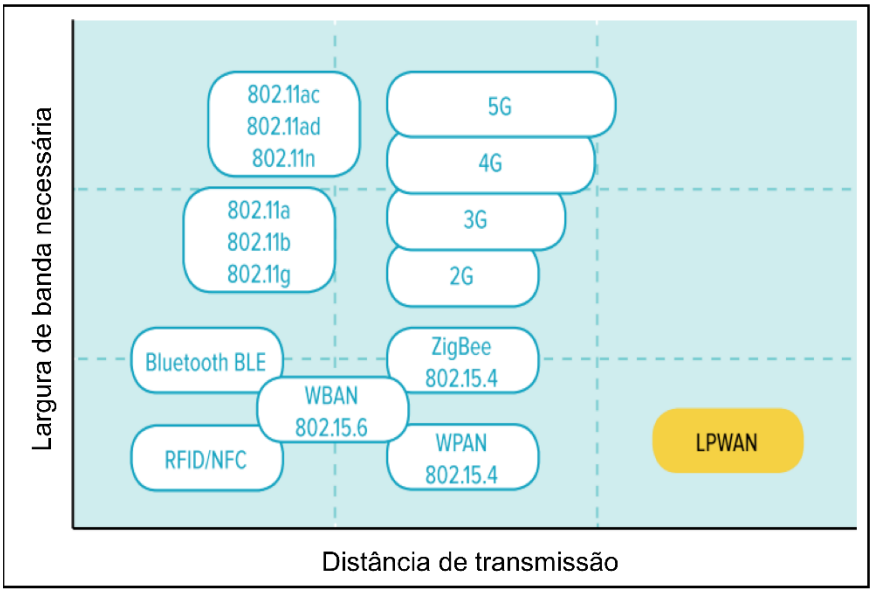
Personal Area Network (PAN) pode ser entendida como uma rede para conectar dispositivos em uma área pessoal, ou seja, limitado a alguns metros. Uma rede de área pessoal sem fio busca estabelecer esta mesma rede, obviamente, sem cabos, e exemplos destas redes são o *Bluetooth* ou *Zigbee* (KUROSE; ROSS, 2013).

Estas redes são especificadas nas diversas divisões da IEEE 802.15. Por exemplo, a 802.15.5 é um projeto de arquitetura de rede que teve como objetivo tornar dispositivos WPAN capazes de promover redes *mesh* interoperáveis, estáveis e escaláveis. Para diferenciar alguns itens existem dois ramos, um para redes de baixa taxa de transmissão e outra para altas taxas de transmissão. As características comuns aos dois ramos englobam a inicialização da rede e endereçamento. No caso das redes *mesh* de baixa taxa de transmissão, a arquitetura suporta *broadcast*, *multicast*, traço de rotas e meios específicos para economia de energia (IEEE, 2009).

2.4.2 Redes de grande alcance com baixo consumo de energia (LPWAN)

Rede LPWAN (*Low Power Wide Area Network*) é um padrão de rede utilizado amplamente em soluções IoT (*Internet of Things* - Internet das Coisas) e seu principal atributo é o envio de poucas informações a longas distâncias. Exemplos de técnicas de modulação que se encaixam no quadro de LPWANs são o LoRa, Sigfox e até alguns tipos de *Long Term Evolution* (LTE). A Figura 7 posiciona as LPWANs de acordo com sua capacidade de envio e banda.

Figura 7 – Tecnologias de transmissão de dados



Fonte: Linklabs (2017, p.4).

Conforme LinkLabs (2017), são três as principais características das LPWAN: longo alcance (dependendo da tecnologia dos rádios pode alcançar 10 km de distância), taxa de transmissão baixa (algo em torno de 5 kbps) e, por final, baixo consumo de energia.

LinkLabs (2017), ressalta ainda que, são duas as aplicações alvo deste tipo de tecnologia: Uma delas são as cidades, com grande concentração de conexões e disponibilidade de energia, onde as LPWAN podem ser boas alternativas às redes de celulares geralmente já congestionadas para projetos diversos, como iluminação inteligente ou GPS. Por outro lado, em ambientes geralmente de difícil acesso e que necessitem de soluções a bateria também são propícios à este tipo de rede, como cita LinkLabs (2017), detectores de vazamento em gasodutos, medições para agricultura (*Smart Agriculture*) e os controles de acesso são exemplos.

Tecnologias como o *ZigBee* que implementam redes *mesh* nativas tem enfrentado problemas uma vez que sua capacidade de transmissão cai muito rápido em distâncias relativamente baixas como 30 metros. Por isso, as LPWANs tendem a ser organizadas na topologia estrela, pois sua configuração é mais fácil (LINKLABS, 2017).

Do ponto de vista técnico, as LPWANs atingem longas distâncias devido aos receptores que tem sensibilidade de, no mínimo, -130 dBm. Se comparadas com outras tecnologias *wireless*, a *Wi-Fi*, por exemplo, utiliza sensibilidade de cerca de -90 dBm, que se comparada com os -130 dBm das LPWANs, a tornam 10000 vezes mais fraca em termos de recebimento de sinais (LINKLABS, 2017).

Uma das características fundamentais, conforme LinkLabs (2017), é que a “maioria das tecnologias LPWAN usam uma banda sem licença”. Nas américas, a faixa dos 915 MHz tem esta característica, já na Europa pode-se utilizar a frequência dos 868 MHz. A justificativa para estas frequências está na quantidade de espectro que pode ser alocado; Em bandas licenciadas é possível conseguir algo em torno de 1 MHz de espectro, enquanto que nas faixas mencionadas pode-se utilizar até 25 MHz. Em consequência, tratando-se de uma tecnologia de baixa taxa de transmissão, quanto mais espectro disponível, maior será a capacidade de transmissão da LPWAN. Uma das desvantagens apontadas pela LinkLabs a respeito da faixa de frequência é de que não há um padrão de licenciamento para redes sub-giga, em geral segue-se o padrão americano de 915 MHz ou o europeu de 868 MHz e, com isso, ainda não há uma disponibilidade global para as LPWANs em comparação com o *Bluetooth* ou a Wi-fi (LINKLABS, 2017).

2.5 Redes *Mesh*

As redes de malha devem ser redes organizadas de tal maneira que torna-se possível a comunicação de qualquer par de dispositivos presentes. *Wireless Mesh Networks* (WMN), em português redes de malha sem fio, consistem em dois elementos principais, clientes *mesh* e roteadores *mesh*, e seu intuito é prover acesso tanto para clientes do tipo *mesh* quanto para clientes comuns. Esta é uma das principais vantagens da topologia *mesh*, uma vez que clientes comuns de *Wi-Fi* (por exemplo) conectam-se a uma única rede de abrangência muito mais ampla do que as redes organizadas em estrela (AKYILDIZ; WANG; WANG, 2016)

Faccin et al. (2008), descreve em sua pesquisa que as redes *mesh* possuem a topologia de criação, o roteamento, o MAC, a segurança, a qualidade do serviço (QoS)

e a eficiência energética como características essenciais. Os mesmos autores destacam ainda a vantagem da escalabilidade da rede de malha sem fio, ainda que a ideia de posicionar diversos pontos de acesso a rede seja tentadora pelo baixo custo destes equipamentos, ainda é necessário um *link* cabeado para que haja sinal com qualidade aceitável. Em compensação, as redes de malha são estruturadas e configuradas a ponto de não ocorrer perda de pacotes ainda que o *link* seja sem fio.

A concepção de uma rede de malha inicia-se com dois processos, explicam Faccin et al. (2008): Uma estação associa-se com um *gateway* e nós associam-se com o *gateway*. A partir deste tipo de interação entre os dispositivos, pode-se entender uma rede *mesh*, como uma composição de diversas conexões do tipo *ad hoc* (conexões ponto a ponto) e o roteamento geralmente ocorre baseado no endereço físico de cada dispositivo na segunda camada do modelo OSI (FACCIN et al., 2008). Akyildiz, Wang e Wang (2016), indicam, ainda, que uma rede *mesh* pode ser composta apenas de nós do tipo cliente. “Neste tipo de arquitetura, nós clientes constituem a própria rede para executar as funcionalidades de configuração e roteamento além de prover aos usuários finais as aplicações” (AKYILDIZ; WANG; WANG, 2016, tradução do autor).

2.5.1 Características gerais das redes *Mesh*

Desenvolver redes de malha sem fio tem como intuito expandir a área de cobertura das redes *wireless* sem que haja perdas de pacotes ou saturação de dados (AKYILDIZ; WANG; WANG, 2016). Outro atributo deste tipo de rede é fornecer conectividade ponto a ponto sem que hajam *links* “*line-of-sight*”, ou seja, comunicar dois pontos da rede diretamente, ainda que não haja visada física direta. Para que sejam obtidos estes resultados, Akyildiz, Wang e Wang (2016), complementam que é indispensável que haja redirecionamento de mensagens entre nós da rede, conceito conhecido como *multi-hop*, desta forma *links* de menor distância são utilizados, poluindo menos o canal e diminuindo os riscos de interferências devido a sinal com baixa potência de recepção.

As redes *mesh* essencialmente devem ser capazes de se auto formar, organizar e curar. Uma vez configuradas, este tipo de rede deve ser capaz de lidar com aparições e sumiços de nós, criando novas rotas e organizando-se a fim de garantir a entrega dos pacotes. Desta forma, as redes *mesh* tornam-se relativamente baratas inicialmente e além disso podem ser ampliadas gradativamente sem necessidade de alteração em todo sistema (AKYILDIZ; WANG; WANG, 2016).

Também é indispensável que seja possível trocar informações com a Internet. Isso faz da rede *mesh*, uma rede de múltiplo acesso, na qual nós da rede comunicam-se entre si e devem ser capazes de receber ou enviar informações a um servidor na nuvem, por exemplo (AKYILDIZ; WANG; WANG, 2016).

Existem alguns fatores para o desenvolvimento e operação destes tipos de rede que são cruciais e devem ser considerados. Tecnologias de rádio têm avançado rapidamente e por isso vem tornando-se um problema menor no desenvolvimento de redes *mesh*, uma vez que sistemas *Multiple Inputs Multiple Outputs* (MiMo) em português Múltiplas Entradas Múltiplas Saídas, tem sido tendência entre as fabricantes, além de sistemas multicanais e antenas mais eficientes, dois exemplos claros são as tecnologias já mencionadas ZigBee e LoRa (AKYILDIZ; WANG; WANG, 2016).

Escalabilidade de rede também é um fator que determina a boa qualidade do serviço. É convencionado que redes com conexões *multi-hop* sofrem deste mal e sua performance cai rapidamente quando o número de conexões de um único nó sobe. A IEEE 802.11 convenciona que a partir de 4 conexões não é possível atingir uma taxa de transmissão aceitável para os padrões da norma (HUANG; LAI, 2002).

Segurança da rede também é indispensável e “sem uma solução de segurança convincente, WMNs não serão bem sucedidas devido a baixa tendência de clientes utilizarem um serviço não confiável” (AKYILDIZ; WANG; WANG, 2016, tradução do autor). Os autores ainda destacam que muito embora já hajam diversos sistemas de segurança para LANs e *Wi-Fi*, a topologia de malha faz com que estas soluções não sejam aplicáveis devido a arquitetura distribuída do sistema e com isto a segurança de redes *mesh* tende a ser objeto de estudos futuros.

2.5.2 Aplicações e IoT

Conforme o conceito de rede *mesh* é apresentado, diversos desafios cotidianos de comunicação de dados podem ser classificados como solucionáveis. Este tipo de rede possui particularidades interessantes que podem resolver tais desafios, ou mitigá-los.

Uma das aplicações comentadas por Akyildiz, Wang e Wang (2016) é a utilização da topologia para fornecimento de conexão a internet para áreas metropolitanas, uma vez que a rede é capaz de entregar conexões mais rápidas do que as de celular a um menor custo. Sistemas de transporte públicos também podem ser beneficiados com esta topologia, no caso de um trem, os vagões podem comunicar-se com intuito de gerar informações tanto para operação quanto para entretenimento e informação ao usuário.

A automação residencial, de acordo com Akyildiz, Wang e Wang (2016) é uma das principais aplicações para este tipo de rede, uma vez que uma série de dispositivos cada vez mais tendem a ter conexão com a internet. Eletricidade, iluminação, elevadores e ar condicionado são alguns exemplos mais comuns que são de essencial controle e monitoramento. Fathany e Adiono (2016) também relatam a importância desta aplicação da topologia *mesh*, que se aplicada corretamente entrega resultados como economia de energia, iluminação e temperaturas confortáveis, além de segurança e confiabilidade no sistema em geral. A principal vantagem, destacada tanto por Akyildiz, Wang e Wang (2016) quanto por Fathany e Adiono (2016) é que a infraestrutura necessária de redes *mesh* limita-se basicamente a alimentação de energia do dispositivo, tornando-a muito mais barata e rápida de se implementar. Ainda no âmbito residencial e até empresarial, destaca-se a aplicabilidade de redes *mesh* para sistemas de vigilância e segurança, pois não há dependência de sistemas centralizados como ocorre hoje, inibindo o funcionamento completo e portanto deixando vulnerável a área em caso de falha da central de alarmes (AKYILDIZ; WANG; WANG, 2016).

2.5.3 Redes *mesh* em diversas camadas do modelo RM-OSI

Dentre as sete camadas do modelo de referência OSI, revisadas nos itens e subitens da seção 2.2, o desenvolvimento e gerenciamento de uma rede do tipo *mesh* pode ser alocada e idealizada em diversos níveis do modelo.

A camada de enlace oferece o serviço de controle de acesso aos circuitos da rede propriamente ditos (MAC), em redes típicas esta etapa lida somente com conexões ponto a ponto, uma vez que o protocolo de roteamento é encarregado de redirecionamento de pacotes (dando origem a uma comunicação multiponto). A principal diferença é que numa rede *mesh* o MAC precisa lidar com comunicação para e de mais de um ponto, uma vez que uma mensagem pode ter origem de um nó que serviu apenas como ponte (AKYILDIZ; WANG; WANG, 2016). Os autores afirmam também que na rede *mesh* o MAC precisa ser cooperativo e distribuído, uma vez que não há um concentrador que gerencia a rede como na topologia estrela, então todos os pontos da rede devem ser utilizados com o intuito de garantir o desempenho da rede. Grossglauser e Tse, 2002, estabeleceram a relação entre a mobilidade e desempenho do protocolo MAC em redes *ad-hoc* e puderam concluir que a partir de mudanças de posições aleatórias dos nós da rede a taxa de transmissão mantém-se constante. Este estudo foi motivado perante a observação dos autores de que redes *ad-hoc* fixas perdiam muito desempenho com o aumento de nós em uma determinada área.

"*Wireless Mesh Networks* serão estreitamente integradas com a internet e o IP tem sido aceito como protocolo de rede para muitas redes sem fio, inclusive as redes *mesh*" (AKYILDIZ; WANG; WANG, 2016, tradução do autor). O protocolo IP em geral não é utilizado em redes de malha e existem algumas variações de técnicas para construção destes protocolos como o *Topology Broadcast Based on Reverse-path Forwarding* (TBRPF), no qual roteadores da Firetide Networks baseiam-se ou ainda a técnica *Dinamic Source Routing* (DSR) a qual foi utilizada pela Microsoft em seu protocolo de rede *mesh* (AKYILDIZ; WANG; WANG, 2016; MICROSOFT, 2007). O principal fator que diferencia as redes *mesh* das redes *ad-hoc* ou estrela, avaliam Akyildiz, Wang e Wang (2016), são requisitos de eficiência energética e mobilidade, uma vez que muitos nós da rede provavelmente não terão fonte de alimentação disponível por serem móveis e esses fatores remetem a protocolos de rede eficientes

e econômicos na medida do possível. Ainda que estudos anteriores tenham sido feitos, os especialistas reiteram que ainda há muito espaço para novos estudos de técnicas de roteamento para redes do tipo malha em consenso de que nem todas as variáveis de desempenho foram avaliadas até então. No entendimento dos autores, os protocolos de roteamento precisam ter como característica escalabilidade, diferentes formas de avaliar o desempenho (métricas de performance), tolerância a falhas de rotas e balanceamento de tráfego. Dito isso, entende-se que integrar diferentes medidas para obtenção do melhor desempenho de um roteamento ainda é um desafio.

Os autores salientam que um desenvolvimento entre camadas tende a ser a melhor alternativa para redes em malha e destaca-se duas maneiras plausíveis para tal. Uma das alternativas é melhorar o desempenho de uma das camadas com base em parâmetros obtidos através de rotinas em outras camadas inferiores, um exemplo simples é a camada física informar a taxa do sinal entre dois pontos para a camada de roteamento. Por outro lado, também é possível mesclar características de diversas camadas em apenas um elemento completo e com isso o desempenho geral deve ser mais elevado se comparado com um *design* de camadas distintas. No caso do desenvolvimento *cross-layer* (camadas cruzadas) deve-se ter em mente que a arquitetura geral do protocolo pode tornar-se confusa de modo que não haja abstração de componentes devido ao grande número de interações entre tarefas de camadas. Por isso, apesar do melhor desempenho após o processo estar concluído, o caminho de desenvolvimento de comunicação *cross-layer* é custoso e requer tempo, além do mais nem sempre poderá ser atualizado sem que inúmeras mudanças sejam necessárias devido a correlação dos componentes do projeto (AKYILDIZ; WANG; WANG, 2016; ESIGN; AWADIA; ECHNOLOGIES, 2005).

2.6 LoRa

Os capítulos 2.2 à 2.5 trataram de uma revisão a cerca de conceitos importantes tal como redes em geral, com ênfase em *mesh*, LPWANs e alguns conceitos de IoT. É de entendimento comum que aplicações IoT tendem a ter limitação energética e este é, em resumo, o principal motivo do surgimento da especificação

das LPWANs: “oferecer cobertura de rádio em uma grande área através de estações que adaptam a taxa, potência e modulação de transmissão[...] de forma que os dispositivos finais tenham um baixo consumo energético enquanto conectados” (AUGUSTIN et al., 2016).

Long Range (LoRa) é um protocolo LPWAN exclusivamente de camada física, concedido pela Cycleo na França e adquirido pela americana Semtech no ano de 2012 que mantém os direitos do protocolo até hoje. A Semtech (2015) define o protocolo como um esquema de modulação de espectro difuso, derivado do existente *Chirp Spread Spectrum* (CSS), que visa o balanço entre taxa de transmissão e alcance em um canal de largura de banda fixo. O LoRa:

> [...] implementa uma taxa de transmissão variável, usando fator de espalhamento ortogonal, que possibilita o desenvolvedor do sistema optar por taxa de transmissão ou potência, para otimizar o desempenho da rede numa banda constante” (SEMTECH, 2015, p. 4) .

A modulação CSS foi inicialmente concebida para utilização militar e serviços de troca de mensagens seguras na década de 1940, porém a baixa potência de transmissão e robustez contra interferências de outros canais, bem como atenuação e bom comportamento do sinal a fenômenos físicos como o efeito Doppler (SEMTECH, 2015). Atualmente o CSS faz parte do padrão IEEE como uma técnica de modulação e consta na ramificação 802.15.4 da norma.

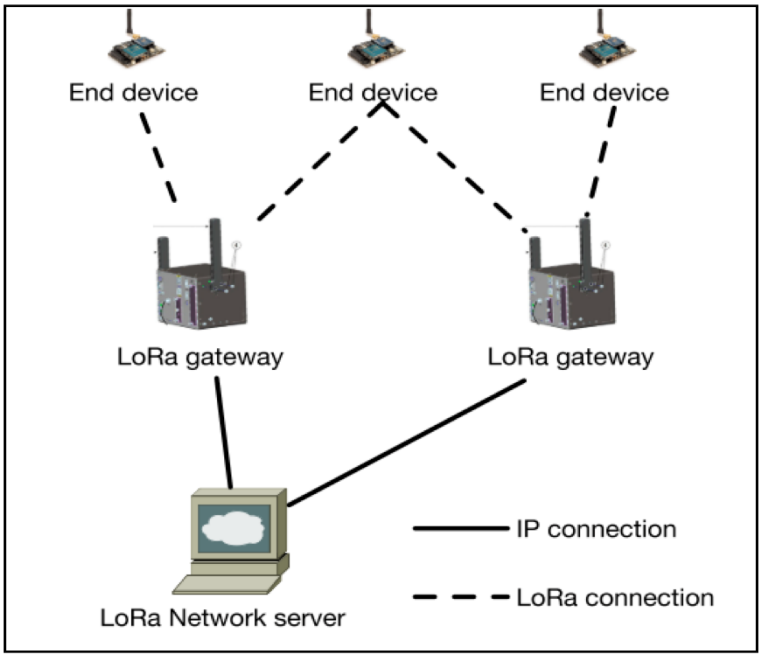
LoRa pode ser definida, hoje, em duas camadas diferentes do modelo OSI. Uma delas é uma camada física que utiliza técnica de modulação CSS, a outra define-se como um protocolo da camada MAC (Enlace) através do *Long Range Wide Area Network* (LoRaWAN).

A camada física, desenvolvida pela Semtech, opera nas frequências 433, 868 e 915 MHz dependendo da região, estas frequências pertencem a faixa *Industrial, Scientific and Medical* (ISM) destinada a outras aplicações que não telecomunicações e portanto são livres de regulações. Cada pacote LoRa contém entre 2 e 255 *bytes* e a taxa de transmissão atinge cerca de 50 kbps (AUGUSTIN et al., 2016).

Em contrapartida a modulação, a camada de controle de acesso ao meio (MAC) é mantida por uma organização denominada LoRa Alliance que mantém o acesso aos

documentos do protocolo chamado de LoraWAN completamente livre (AUGUSTIN et al., 2016).

Figura 8 – Exemplo de rede LoRaWAN



Fonte: AUGUSTIN et al. (2016, p. 4).

2.6.1 Camada Física

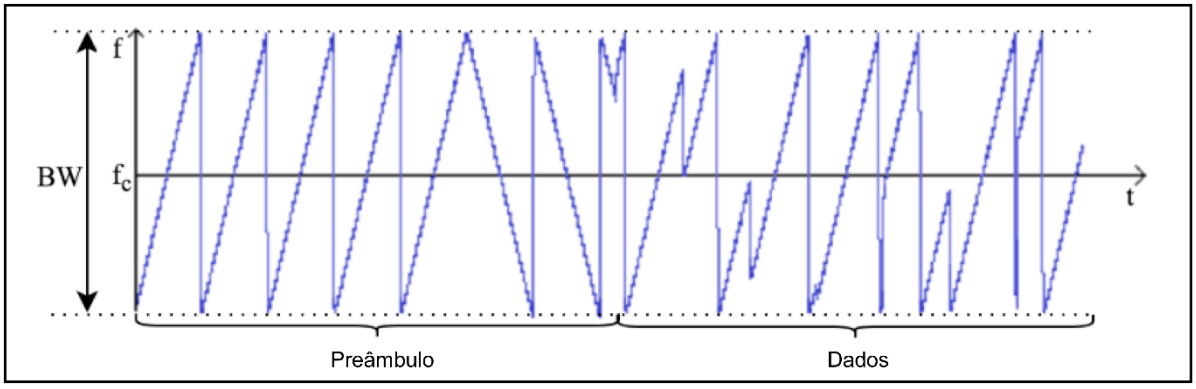
Conforme já mencionado, LoRa é baseado na modulação CSS que pode ser explicada como um aumento ou diminuição da frequência do sinal ao longo do tempo para decodificar os *bits*. Augustin et. al. (2016), complementa que a linearidade desta variação faz com que os *offsets* (diferença) entre transmissor e receptor sejam interpretados como *offsets* de tempo que podem ser facilmente eliminados, isso faz com que a modulação LoRa seja imune ao efeito Doppler (gera *offset* de frequência entre dois pontos) e, portanto, pode ser empregada a dispositivos em movimento.

A diferença entre as frequências do sinal entre transmissor e receptor “podem atingir até 20% da largura da banda sem que seja impactada a decodificação”, por isso os transmissores LoRa tornam-se mais baratos uma vez que seus cristais osciladores não necessitam extrema precisão para que a transmissão ocorra (AUGUSTIN et al., 2016).

Muito embora boa parte do protocolo da camada física esteja indisponível, existem alguns estudos a respeito dos três parâmetros configuráveis na transmissão LoRa e seus efeitos para com os dados. Pode-se parametrizar, no sinal LoRa, a largura da banda (BW), o *Spreading Factor* (SF), em português fator de espalhamento, e o *Code Rate* (CR), em português taxa de código, é uma verificação de redundância utilizada para mitigar ruídos e interferências externas). Tais variáveis influenciam a taxa de transmissão, a resistência a interferência e a dificuldade na decodificação (AUGUSTIN et al., 2016).

Para Augustin et. al. (2016) a largura da banda é o parâmetro mais significativo da modulação LoRa, um símbolo[2] é composto por 2^{SF} mudanças de frequência (*chirps*) que cobrem toda a largura da banda. A Figura 9 representa a variação da frequência em uma transmissão ao longo do tempo. A posição de cada descontinuidade observada é o que codifica o dado, como ocorrem 2^{SF} variações, um símbolo é capaz de codificar SF *bits* de informação.

Figura 9 – Variação da frequência do sinal em uma transmissão LoRa



Fonte: AUGUSTIN et al. (2016, p. 5).

Em resumo, “a taxa de variação de frequência é igual a largura de banda (uma variação de frequência por segundo por Hertz da largura de banda)” (AUGUSTIN et al., 2016). Com isso, pode-se admitir que um aumento da largura da banda implica na diminuição da sensibilidade do receptor, à medida que o aumento do fator de

[2] Na modulação LoRa, um símbolo corresponde a um *chirp*. Um *chirp* deve entendido como o aumento ou diminuição da frequência de um sinal ondulatório. Um símbolo pode ser entendido, então, como a representação de um ou mais *bits* no meio físico (SEMTECH, 2015).

espalhamento aumenta esta sensibilidade. O autor ainda define a equação 1 que representa a taxa de transmissão útil em *bits* por segundo baseada nos parâmetros mencionados.

$$R_b = SF * \frac{BW}{2^{SF}} * CR \quad (1)$$

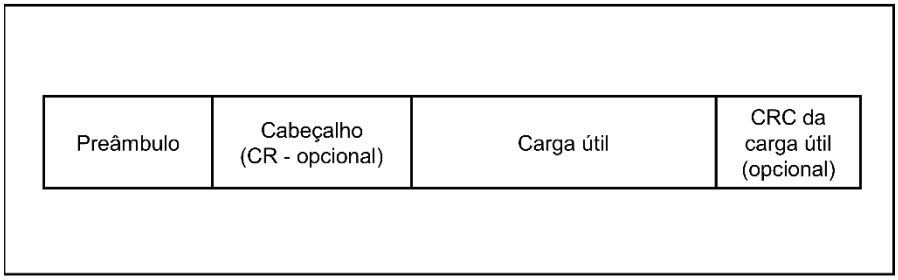
2.6.2 O frame LoRa

É importante conhecer o formato de um *frame* LoRa, ou seja, o que o sinal codificado LoRa carrega. Sabe-se que o frame implementado nos transmissores e receptores do protocolo tem a largura de banda e o fator de espalhamento constantes (AUGUSTIN et al., 2016).

O *frame* inicia com uma sequência fixa de variação de frequência cobrindo toda a banda de frequência, esta etapa é denominada de preâmbulo. As duas últimas variações são correspondentes a um valor contido em um *byte* que é utilizado para diferenciar redes LoRa que utilizam a mesma frequência, chamado de *sync word*, desta forma um receptor configurado com uma *sync word* diferente irá ignorar qualquer sinal captado que contenha um preâmbulo diferente (AUGUSTIN et al., 2016).

Após a introdução, o frame pode contar, ou não, com um cabeçalho (*header*). Esta fatia indica o tamanho da mensagem em bytes (*payload*), a taxa de codificação utilizada e pode conter um código de detecção de erros caso o tamanho da mensagem seja variável na rede implementada. Por fim a mensagem em si é adicionada ao frame como mostra a Figura 10 (AUGUSTIN et al., 2016).

Figura 10 – Estrutura de um frame LoRa



Fonte: Adaptado pelo autor de Augustin et al. (2016, p. 7).

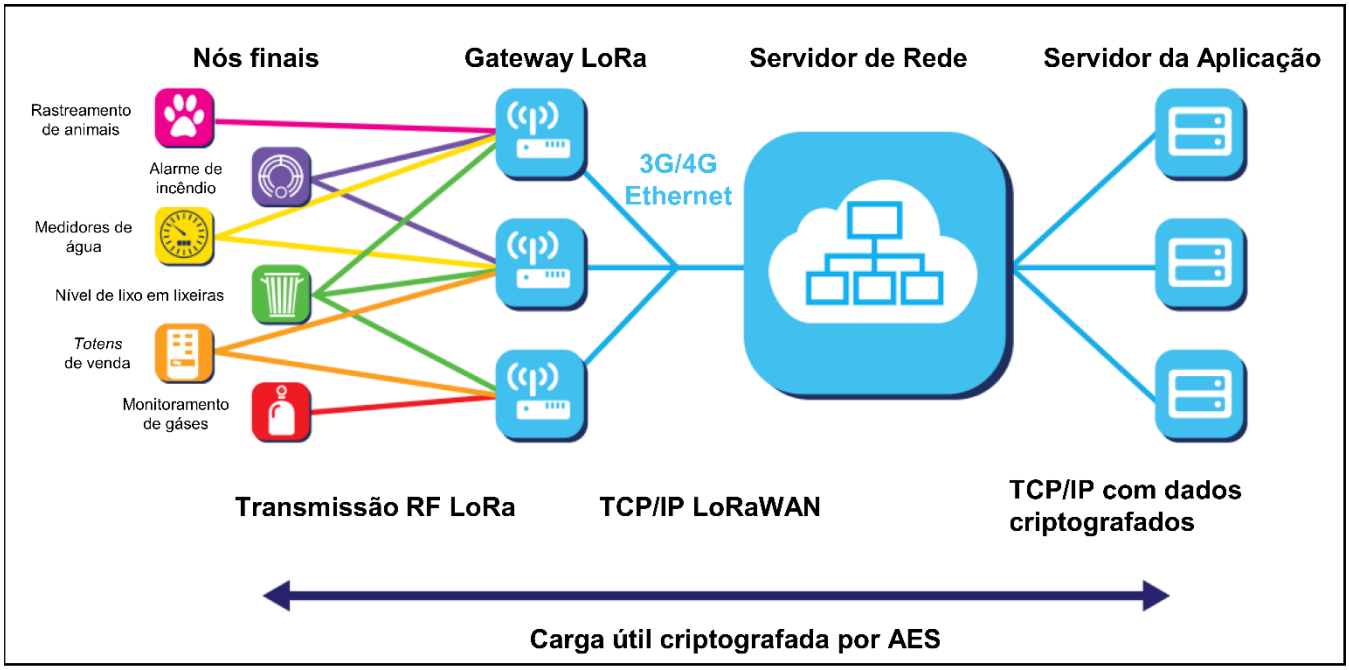
2.7 LoRaWAN

LoRaWAN trata-se de um protocolo de rede construído em cima da camada física e de enlace LoRa da Semtech. O protocolo foi concebido em 2015 pela organização LoRa Alliance que o mantém até hoje. Esta organização tem cerca de 500 membros (entre eles empresas e fabricantes) que “estreitamente colaboram e trocam experiências para promover e impulsionar o sucesso do protocolo LoRaWAN como padrão global aberto líder para conectividade IoT LPWAN segura” (LORA ALLIANCE, 2015).

Uma rede LoRaWAN é considerada uma topologia do tipo estrela de estrelas, na qual diversos *gateways* encaminham mensagens dos dispositivos finais para um servidor central da rede (WATTEYNE et al., 2017). O princípio básico desta rede é de que os nós finais enviam os dados, através de uma conexão direta, aos *gateways*, os quais têm algum tipo de conexão com a Internet, seja ela por tecnologia celular, *Wi-Fi* ou *Ethernet*, com o objetivo de consolidar os dados a um servidor acessível (LORA ALLIANCE, 2015).

Nesta rede, os nós não estão diretamente associados com um *gateway*, desta forma, os dados são transmitidos livremente e podem ser recebidos por mais de um *gateway* de rede. Cada pacote recebido será encaminhado para o servidor da rede, que então lidará com possíveis duplicações, autenticação e reconhecimento de entrega. Esta característica é importante do ponto de vista de aplicações IoT pois é natural que haja mobilidade de nós e este tratamento não individual de cada nó exclui a necessidade de reconfiguração quando ocorrem deslocamentos (LORA ALLIANCE, 2015). A Figura 11 define claramente a estrutura de uma aplicação LoRaWAN.

Figura 11 – Estrutura de uma rede de aplicação com LoRaWAN



Fonte: Adaptado pelo autor de LORA ALLIANCE (2015, p. 8)

LoRaWAN define três tipos diferentes de dispositivos em sua rede, separados em classes (A, B e C). O dispositivo classe A tem como característica o aguardo de uma resposta por um período (configurável, geralmente 2 segundos) logo após o envio de algum dado. Esta classe tem o menor consumo de energia entre as classes pois só recebe informações da rede (portanto consome energia 'ouvindo') quando o uma mensagem chegou com sucesso a um *gateway* (WATTEYNE et al., 2017). O dispositivo Classe B é semelhante, porém a janela de recebimento de dados é configurada e portanto não tem característica aleatória como no classe A. Na classe B o dispositivo recebe um despertar do *gateway* em períodos pré-determinados, desta forma há garantia de que o dispositivo está 'ouvindo' (LORA ALLIANCE, 2015). Já a classe C de dispositivos tem o maior período de recepção dentre as classes de dispositivos, sua janela é contínua e apenas é fechada em caso de transmissão. A classe A é recomendada para sensores a bateria, atuadores a bateria são recomendados que utilizem classe B enquanto que atuadores com fonte de energia constante são geralmente utilizados na classe C (LORA ALLIANCE, 2015; LORA ALLIANCE TECHNICAL COMMITEE, 2017; WATTEYNE et al., 2017).

É importante mencionar que o protocolo LoRaWAN utiliza duas camadas de segurança, uma de rede e uma de aplicação. A primeira visa garantir genuinidade de

um nó da rede, a segunda garante que o operador da rede não tenha acesso a dados da aplicação. Para isso utiliza-se encriptação do tipo AES[3] em trocas de chave com identificador IEEE EUI64.

[3] Padrão de Criptografia Avançada é uma técnica de criptografia para dados eletrônicos baseada em uma chave simétrica, isso significa que ela é usada tanto para criptografar como para descriptografar os dados.

3 PROJETO DO SISTEMA

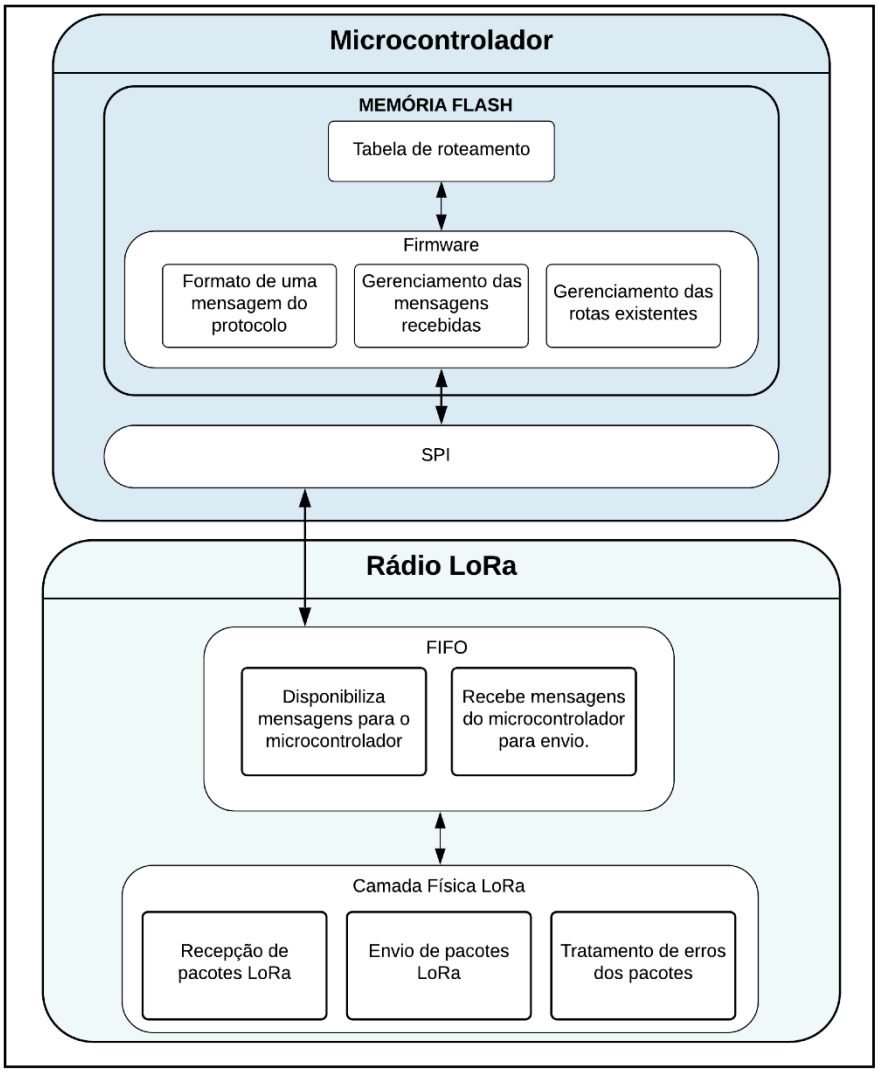
O *software* embarcado a ser desenvolvido capacitará a inicialização uma rede de dispositivos semelhantes (ou juntar-se a uma existente) para desempenhar a tarefa de encaminhamento de mensagens de uma forma eficiente. Lê-se “de uma forma eficiente” como: comunicar dois dispositivos da rede que não estejam disponíveis para comunicação direta. Com isso objetiva-se atingir maiores distâncias de tráfego de informações em comparação com a comunicação ponto a ponto LoRa, ou da rede estrela LoRaWAN.

O projeto do protocolo de comunicação *mesh* através de LoRa engloba desenvolvimento de *hardware* e *software*, uma vez que as placas de desenvolvimento existentes para LoRa não oferecem boa construção e acabam por ser ineficientes nas suas transmissões.

O projeto conta então, com o desenvolvimento de uma placa de circuito impresso previamente, que engloba o microcontrolador, o rádio LoRa e alimentação para ambos os módulos. Para os testes preliminares é provável que esta etapa de desenvolvimento da placa ainda esteja em andamento, então os ensaios deverão ser desenvolvidos em bancada de testes com *protoboard*.

A Figura 12 representa um panorama geral do projeto, os itens serão discutidos nas próximas sessões.

Figura 12 – Estrutura lógica dos componentes de projeto



Fonte: Autor.

3.1 *Hardware*

As sessões a seguir descrevem o *hardware* escolhido para o projeto.

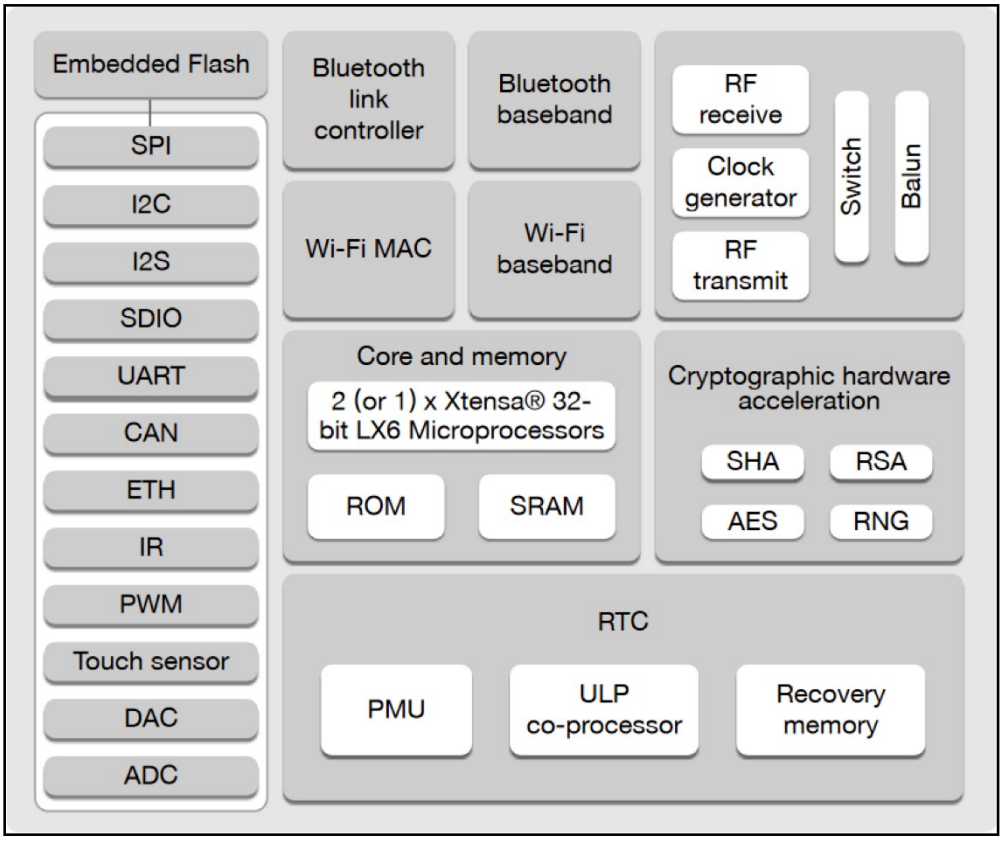
3.1.1 ESP32

A unidade de processamento escolhida para o desenvolvimento é o Espressif ESP32, além do chip, o modelo escolhido conta com um *display* OLED de 0.96” que facilita a visualização de elementos importantes durante o desenvolvimento. O ESP32 opera seus pinos a 3.3 V e conta com comunicação *Wi-Fi* e *Bluetooth*, foi desenvolvido para aplicações de IoT, conta com circuitos auxiliares para gerenciamento de energia e consequentemente diversos modos de operação.

O processamento do ESP32 é realizado por um Xtensa *dual-core* de 32 bits operando a 240 MHz e há uma infinidade de variações de memórias SRAM e ROM para o dispositivo, o escolhido para este projeto conta com 520 KB de memória SRAM e 4 MB de memória *flash* para uso geral (suporta até 16 MB). Também estão inclusos uma unidade RTC com 16 KB SRAM e 2 *timers* de 64 bits de resolução cada com *watchdog* dedicado para cada.

São englobados ao todo 34 GPIOs (todos com suporte para acoplamento de interrupções) programáveis que envolvem 18 canais de conversores analógico-digitais de 12 bits cada, dois conversores digital-analógico de 8 bits, 10 pinos de sensores de toque, 4 interfaces SPI, 2 interfaces I^2C, 3 interfaces UART, até 16 canais de PWM. A Figura 13 ilustra em blocos todas as funcionalidades disponíveis para o microcontrolador.

Figura 13 – Diagrama de blocos do ESP32



Fonte: Espressif (2019, p. 5).

Todas as informações foram retiradas da folha de dados do fabricante Espressif.

3.1.2 RFM95

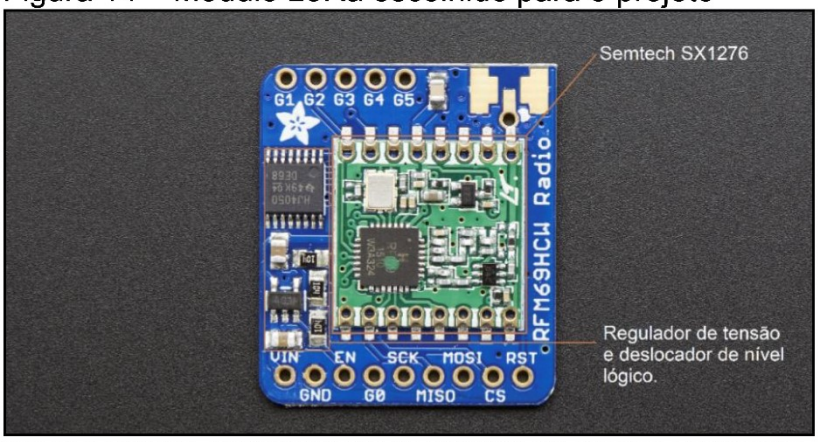
A placa RFM95 é um rádio modulador e demodulador de pacotes LoRa de +20 dBm desenvolvida pela empresa americana Adafruit. Esta placa está baseada no chip da Semtech SX1276, que contém a implementação da camada física do LoRa. A comunicação com a unidade de processamento é feita através de *Serial Peripheral Interface* (SPI), em português interface serial de periféricos.

Segundo a Adafruit, o pico de consumo quando transmitindo em sua potência máxima (+20 dBm) é de aproximadamente 100 mA, enquanto no período de recebimento gira em torno de 30 mA.

Existem variantes desta placa conforme a frequência de operação dos rádios LoRa (433, 868 ou 915 MHz), a banda ISM no Brasil é 915 MHz, portanto foi a escolhida para o projeto. A placa já prevê a adição de um conector coaxial do tipo *SubMiniature version A* (SMA) para antena, o que pesou na escolha deste módulo, pois o desenvolvimento de *hardware* não é foco deste projeto.

Por ser uma placa de desenvolvimento, ela conta com um regulador de tensão e deslocador de nível lógico para 3.3 V (operação do chip SX1276), desta forma permite ligação de até 5 V ampliando a gama de unidades de processamento compatíveis.

Figura 14 – Módulo LoRa escolhido para o projeto



Fonte: Adaptado em 22/05/2019 de https://learn.adafruit.com/adafruit-rfm69hcw-and-rfm96-rfm95rfm98-lorapacketpadiobreakouts/pinouts.

A placa de circuito da Figura 14 tem a dimensão de uma moeda de R$ 1.

Em conjunto com o módulo RFM95 será utilizada uma antena linear com carcaça de borracha, de 915 MHz com potencial de amplificação de aproximadamente 2.15 dBi.

3.1.3 Projeto do circuito de validação

Como parte do projeto, um dos objetivos é desenvolver um circuito que facilite o manejo durante a realização dos testes do projeto como um todo. Visando estas duas características, propõe-se uma placa que engloba a unidade de processamento ESP32 e o *chip* RFM95, além da alimentação necessária para ambos os módulos. A intenção é que seja facilitada a manutenção em caso de falha de algum dos componentes do projeto, por isso, o circuito contará com barras de pinos para conexão do ESP32 e do RFM95 o que torna possível a remoção dos mesmos.

De forma resumida, o objetivo do circuito é conectar fisicamente os pinos necessários para realização da comunicação entre a unidade de processamento e o rádio e fornecer alimentação adequada para ambos os circuitos.

A alimentação da placa será regulada através do CI AMS1117 que suporta até 15 V de entrada e regulará a saída em 3.3 V, sua escolha ocorre, pois ele é empregado na maioria das placas de desenvolvimento de ESP32 e sua aplicação é bastante simples. Apesar da placa do ESP32 e RFM95 já possuir um regulador próprio, optou-se por adicionar mais um regulador para proteger os reguladores dos dispositivos e prolongar a vida útil dos mesmos.

3.2 Protocolo

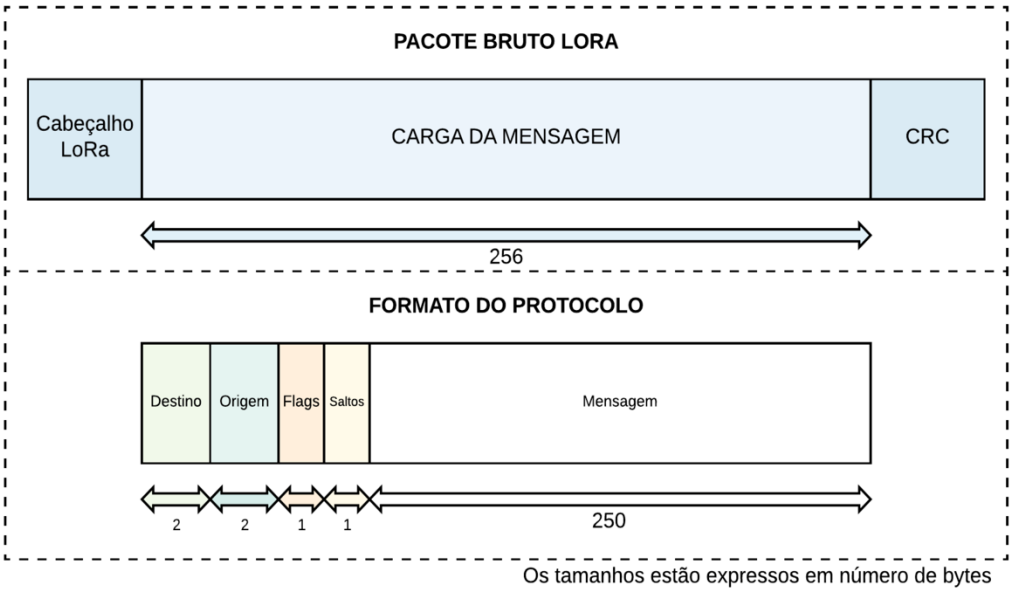
Para melhor esclarecimento do principal objetivo do trabalho, serão abordados três pontos definidos como principais para que os objetivos sejam atingidos. As próximas sessões tratam do formato das mensagens, das regras de encaminhamento de mensagens (roteamento) e da criação e manutenção da rede entre os nós.

3.2.1 Formato de uma mensagem

Em harmonia com o que foi revisado nos capítulos anteriores, é necessário definir um formato para a mensagem que será enviada através dos pontos da rede. A mensagem bem definida é o primeiro passo para que as regras de encaminhamento sejam interpretáveis pelos dispositivos.

Define-se que uma mensagem deve conter todas as informações necessárias para que um dispositivo qualquer que contenha a implementação do protocolo saiba como agir quando a recebe. O *payload* LoRa não deve ultrapassar a quantia de 255 *bytes*, portanto este é o tamanho base de uma mensagem.

Figura 15 – Estrutura definida de uma mensagem



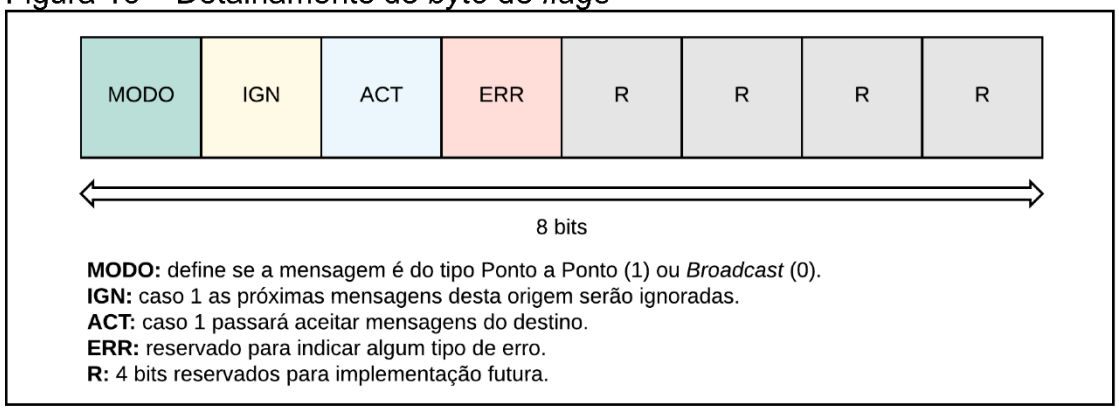
Fonte: Autor.

A Figura 15 apresenta os componentes de uma mensagem do protocolo. São reservados 2 *bytes* para o endereço de destino, da mesma forma 2 *bytes* para o endereço de origem. Adiciona-se 1 *byte* chamado de *flags* que são de uso geral, para apontamento de diretivas referentes ao protocolo, além de um *byte* para sinalização de número de saltos (fica definido que o número máximo de saltos é 255). Com isso 250 *bytes* estão disponíveis para envio de mensagens no protocolo, possibilitando o envio de 250 caracteres da tabela ASCII simples, ou 12 valores inteiros, ou ainda 62 valores do tipo *float*.

Em termos de eficiência do protocolo, parâmetro que relaciona o *overhead* (*bytes* enviados como controle) com o *payload* (carga útil, informação enviada) de uma mensagem está em 97,65%. Se comparado com o RS-232, que para enviar 8 *bits* de informação necessita um total de 11 *bits* (início, fim e paridade) tendo uma eficiência de 72%, o número atingido do projeto mostra-se promissor.

Em laranja, na Figura 16, está representado o *byte* reservado para *flags* do protocolo assim como a disposição de cada *bit*.

Figura 16 – Detalhamento do *byte* de *flags*



Fonte: Autor.

O modo de uma mensagem indica ao dispositivo receptor qual o seu tipo, sendo ponto a ponto uma mensagem endereçável e seu conteúdo de interesse único ao destino, no caso do *broadcast* a mensagem deve ser interpretada por todos os dispositivos da rede. Os bits IGN e ACR são utilizados para controle de fluxo de mensagens, possibilitando que um nó da rede passe a ignorar mensagens de uma origem ou então passe a aceitar mensagens do destino a qual a mensagem pertence, esta é uma característica útil tanto para o gerenciamento da rede manualmente tanto quanto para os testes em bancada do protocolo, uma vez que todos os dispositivos estarão no alcance. O *bit* ERR foi reservado para indicar algum tipo de erro no protocolo. O *nibble*[4] restante está livre para implementações futuras.

[4] *Nibble* é utilizado para representar meio *byte*, ou seja, 4 *bits*.

3.2.2 Comportamento dos dispositivos

A partir do corpo de uma mensagem definido, pode-se estabelecer o funcionamento de cada nó da rede. Serão estabelecidos dois tipos possíveis de dispositivos, um deles funcionando como *gateway* de dados e outro como nó gerador de dados.

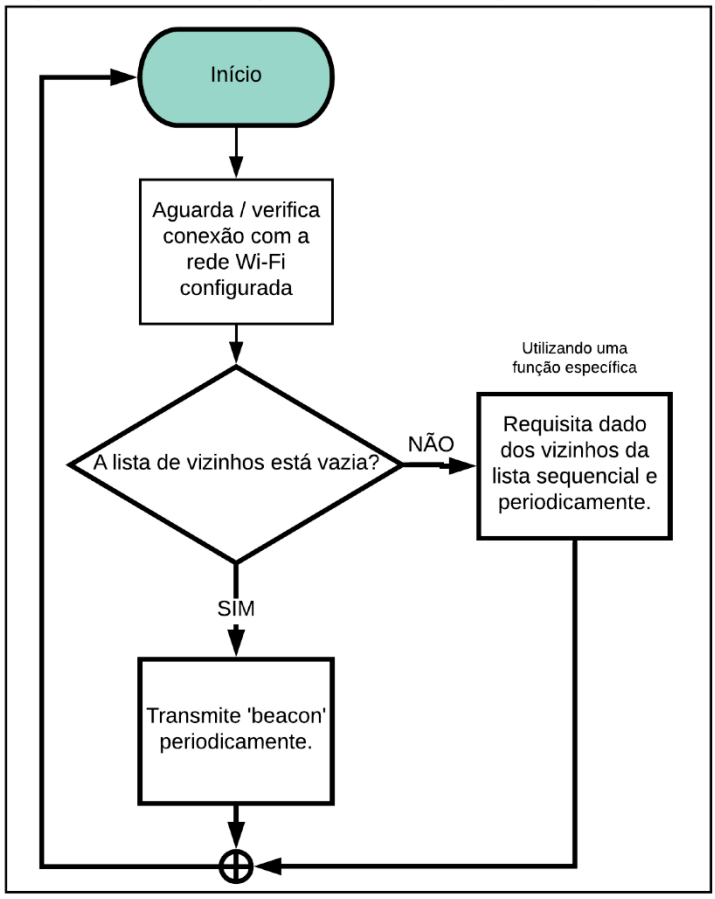
Por tratar-se de uma implementação de baixo nível, o *gateway* é o responsável por requisitar informações dos demais nós existentes, desta forma pode-se evitar colisões entre nós de rede torna-se dispensável a elaboração de uma pilha de mensagens. Apesar de ser um padrão simples se comparado com as redes atuais de comunicação, para o ramo IoT geralmente as informações fluem de forma menos frequente (em comparação com TCP/IP, ou até um protocolo de comunicação industrial) e deve-se priorizar o recebimento da informação.

3.2.2.1 Dispositivo *Gateway*

O *gateway* tem este nome, por ser o caminho pelo qual os dados dos nós da rede poderão fluir para outra rede, como a internet, deste modo o *gateway* pode-se conectar a uma rede *Wi-Fi* previamente cofigurada via *firmware*.

Em primeira instância, ao ligar, este dispositivo irá conectar-se a rede configurada e então iniciar suas atividades de gerenciamento do protocolo. Deve-se atentar que não é objetivo deste trabalho realizar qualquer comunicação com algum servidor externo, garante-se apenas o acesso a Internet. Para melhor entendimento foi desenvolvido um fluxograma que consta abaixo. O dispositivo *gateway* pode ser qualquer nó do conjunto, adota-se que o primeiro dispositivo a ligar iniciará uma rede caso não detecte uma mensagem de alguma rede existente. Com isso, atinge-se um estado de projeto em que todos os microcontroladores poderão contar com o mesmo *firmware*, dando robustez e agilidade no desenvolvimento.

Figura 17 – Fluxograma de um nó *gateway*



Fonte: Autor.

Beacon é um termo utilizado na área de redes para identificar uma mensagem que pode ser comparada com um convite para juntar-se a rede.

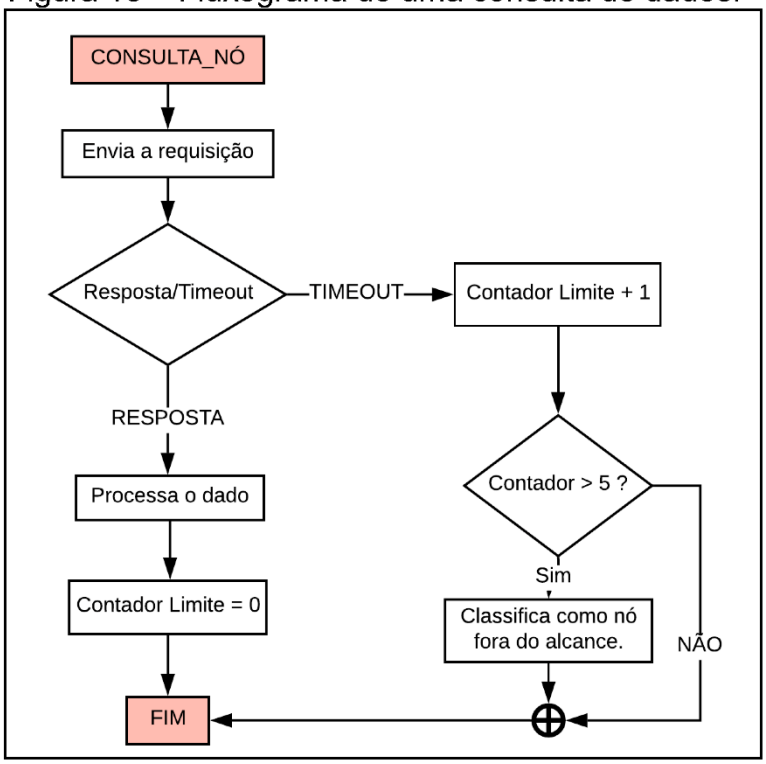
A lista de vizinhos deve ser entendida como a tabela de roteamento. Por tratar-se de uma tabela, os dados são armazenados em um arquivo formato *Comma-separeted Values* (CSV) na memória *flash* do ESP32. Este arquivo conterá todos os nós conhecidos pelo *gateway* além do número de saltos da última vez em que uma mensagem retornou do mesmo.

Uma função específica do *gateway* será responsável por requisitar dados periodicamente aos nós da rede, através de uma marca temporal, por isso, pode-se dizer que o acesso aos dados dos nós se dará de modo determinístico.

O próximo fluxograma é um resumo da função de requisição de dados aos nós, observa-se que há um limite de requisições para um mesmo endereço, caso este

endereço não responda ele é sinalizado como nó fora do alcance e uma mensagem do tipo *broadcast* será enviada para toda rede a fim de notificar a ausência deste destino.

Figura 18 – Fluxograma de uma consulta de dados.



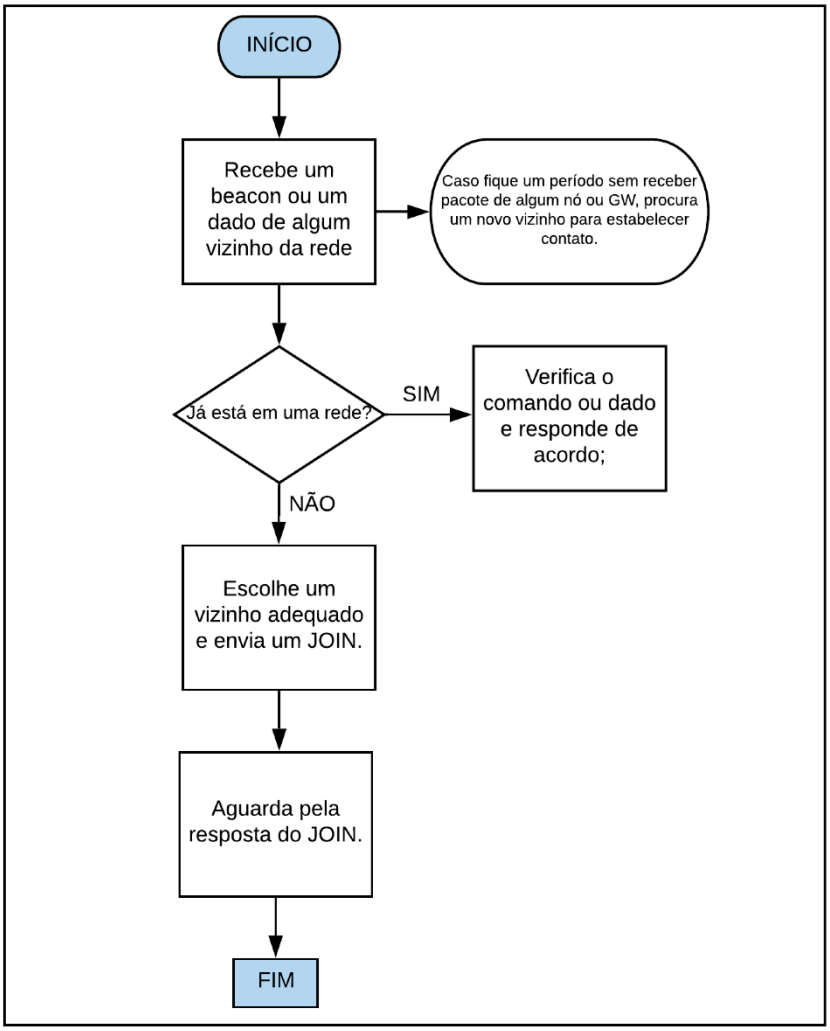
Fonte: Autor.

3.2.2.2 Nó da rede

Um nó de rede é o dispositivo responsável por gerar dados ou informações para o nível de aplicação. Além de responder as requisições de dados do *gateway*, deve servir como ponte para pacotes os quais o destino não é ele mesmo. Da mesma forma fluxogramas são apresentados para melhor visualização da sequência de operação.

O fluxograma abaixo é executado pelo microcontrolador em *loop*, contando o tempo em que o nó está ocioso. Caso atinja um tempo de ociosidade considerado alto (configurável), o nó volta a buscar novos vizinhos na rede, este tipo de verificação é importante na rede *mesh* pois garante a cura da rede.

Figura 19 – Rotina do *loop* de um nó da rede.

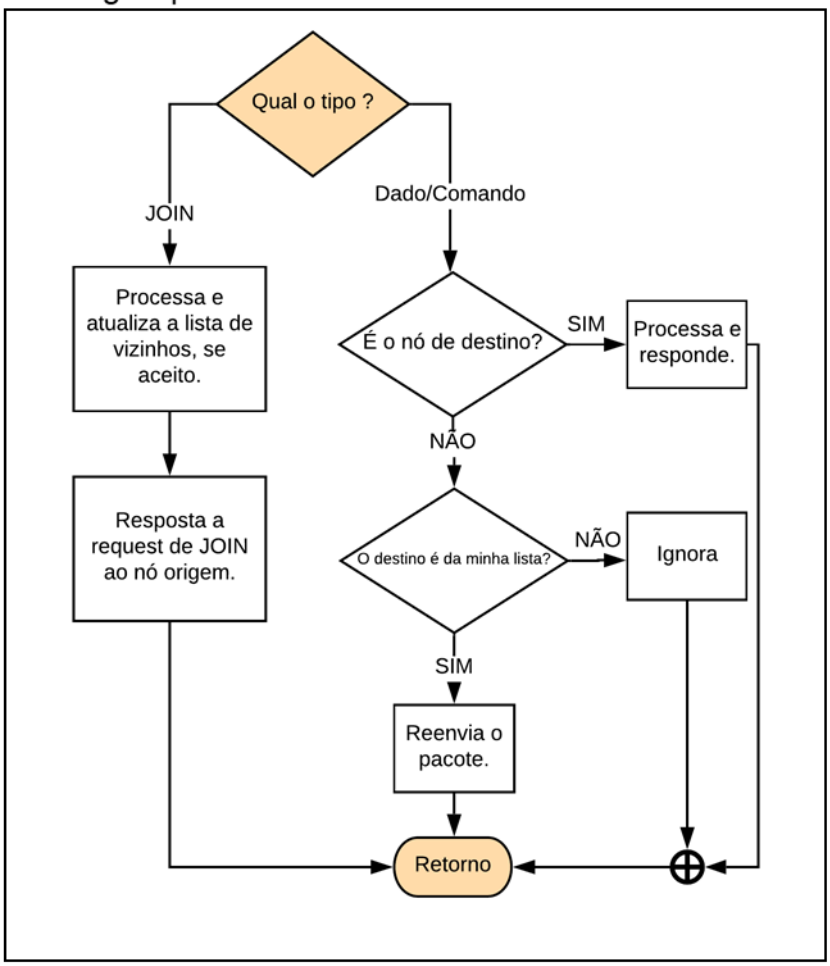


Fonte: Autor.

Quando um nó ficar muito tempo sem receber um pacote para encaminhar ou alguma requisição, entende-se que o vizinho pelo qual ingressou na rede não está mais disponível e a esta altura é provável que uma mensagem de *broadcast* já tenha sido enviada a toda rede pelo dispotivo *gateway* informando a ausência deste nó, então sua rotina passa a ser aguardar o recebimento de alguma mensagem para que possa voltar a rede.

A Figura 20 apresenta o tratamento por um nó de um pacote de dados recebido do rádio LoRa.

Figura 20 – Fluxograma do recebimento de uma mensagem pelo nó.



Fonte: Autor.

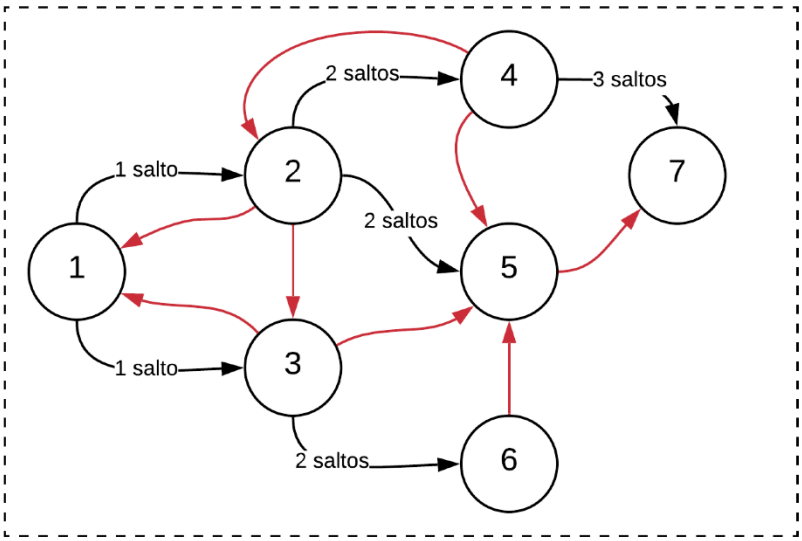
O tipo de pacote *join* é definido quando no *byte* de *flags* da mensagem ocorrer dos bits IGN e ACT estarem em nível alto. Quando identificado esta condição, o nó receptor da mensagem irá tratar de enviar um endereço válido para o novo nó e transmitir uma mensagem do tipo *broadcast* notificando os outros pares da rede do novo destino. A troca de mensagens para realizar estes passos será definida por uma rotina especial durante o desenvolvimento.

3.2.2.3 Fluxo de mensagens na rede

Quando um nó enviar uma mensagem com destino a outro nó, é necessário que sejam estabelecidos critérios de falha para que se evite congestionamento na rede ou perpetuação de uma mensagem com falha.

Definiu-se que cada mensagem carregará a informação de quantos saltos já ocorreram, o incremento de um salto ocorre toda vez que uma mensagem chega a um nó que não é o destino final. O número de saltos máximo será estabelecido na implementação, no caso deste valor ser atingido a mensagem automaticamente será descartada pelos nós.

Figura 21 – Exemplo do fluxo de mensagem.



Linhas em vermelho indicam que a mensagem será ignorada pelo nó destino.

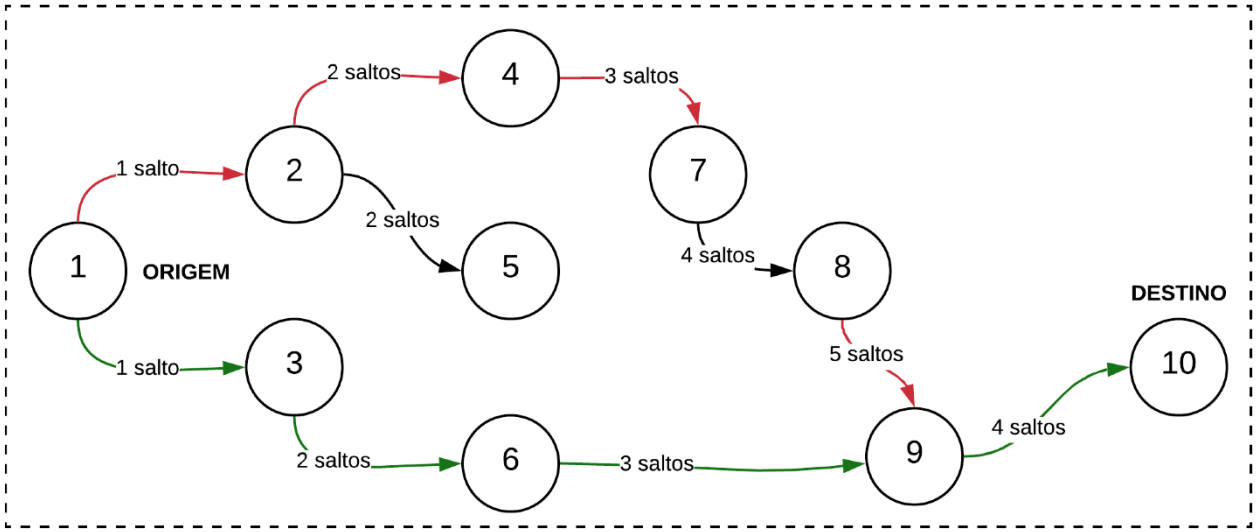
Fonte: Autor.

Todo nó da rede armazena a última mensagem que recebeu para encaminhar, quando uma mensagem recebida for igual a anterior ela é descartada. Este caso deve ocorrer com frequência, pois os dispositivos próximos irão enviar a mensagem a todos os nós vizinhos, fazendo com que o nó de origem receba novamente a mensagem.

A Figura 21 apresenta um exemplo de encaminhamento de uma mensagem com origem no nó 1 com destino ao nó 7. O caminho tomado de 1, 2, 4 e 7 foi estabelecido de forma automática com base nos vizinhos conhecidos de cada nó. O nó 1 enviou a mensagem para os destinos vizinhos em sua tabela de roteamento (2 e 3), já o nó 2 enviou para os vizinhos 1, 3, 4 e 5 e por fim o nó 4 encaminhou ao destino final. É preciso observar que as mensagens do nó 2 para o vizinho 1 e 3 devem ser ignoradas pois a mensagem já havia passado pelos nós.

Já na Figura 22 demonstra-se o caso em que o número de saltos de uma mensagem é ultrapassado.

Figura 22 – Modelo visual do comportamento do sistema para estouro de saltos



Fonte: Autor.

Definido 5 como número máximo de saltos, o comportamento da rede é como o observado acima. O nó de destino (10) não está ao alcance do nó 8, sendo o nó 9 seu único vizinho próximo. A rota destacada em vermelho seria descartada, porém, uma rota possível para a seguinte mensagem seria a destacada em verde atingindo o destino final abaixo do limite de saltos possível.

A limitação do número de saltos implementa uma característica importante para o funcionamento do sistema, uma mensagem não se espalhará pela rede por um caminho que não seja viável até o seu destino. Em redes com número pequeno de nós, é provável que uma mensagem se espalhe por toda a rede, porém o impacto não é grande justamente pelo tamanho da rede. Já em uma situação em que há grande número de dispositivos espalhados é interessante que não ocorra a dissipação da mensagem para uma direção que não é seu destino.

É necessário mencionar que o número de saltos máximo configurado inicialmente deve variar ao passo que a rede cresce, uma vez que novos pontos da rede se encontrem a distâncias maiores do que o número de saltos preestabelecido.

3.2.3 Tabela de repasse e roteamento

Definiu-se que a tabela de repasse de cada dispositivo será montada conforme a rede vai sendo construída em conjunto com a adição de mais nós ao longo do tempo. Cada dispositivo terá em sua tabela de repasse os endereços dos nós os quais estão ao alcance, desta forma a tabela de repasse conterá informações de pontos de comunicação direta. Associado a cada endereço da tabela de repasse, há ainda a intensidade do sinal deste ponto, como exemplifica a Figura 23.

Figura 23 – Exemplo de tabela de repasse

Repasse nó 1	
Endereço	Intensidade do Sinal (dB)
2	-90
3	-97
4	-104
5	-125

Fonte: Autor.

A tabela de repasse de um dispositivo tende a permanecer em constante atualização conforme mensagens cujo arranjo corresponda ao acordado anteriormente cheguem ao nó, por isso uma rotina será responsável pela manutenção e atualização deste registro.

Quando uma mensagem chegar a um nó da rede, após a comparação do destino a qual pertence e o endereço do nó, ela será redirecionada a toda a tabela de roteamento daquele nó, conforme descrito no item anterior. Pode-se dizer que o repasse de uma mensagem ao longo da rede será semelhante a topologia física de barramento, na qual todos os pontos receberão a mesma mensagem.

A tabela de roteamento de cada dispositivo conterá uma linha correspondente a cada nó existente da rede. Quando um novo dispositivo se junta a rede, uma mensagem do tipo *broadcast* inunda a rede e informa a todos os outros do novo ponto de conexão. Ao passo que um dispositivo não responda a uma solicitação do *gateway* por período determinado, por exemplo 3 solicitações ou 2 minutos, o *gateway* também inunda a rede a fim de informar os dipositivos que àquele nó já não está mais

disponível. Estas duas situações são tratadas através de uma rotina responsável pelo gerenciamento da tabela de roteamento.

Conforme a Figura 24, uma linha da tabela de roteamento conterá basicamente os endereços numéricos de cada nó disponível bem como o número de saltos utilizados na última comunicação com o mesmo.

Figura 24 – Exemplo de tabela de roteamento

Roteamento nó 3	
Endereço	Número de saltos
1	1
2	1
4	2
5	2
6	3
7	4

Fonte: Autor.

Em conjunto, as duas tabelas devem ser suficientes para estabelecer conexão entre quaisquer dois pontos da rede. As informações de intensidade do sinal e número de saltos da última interação podem ser utilizadas para o desenvolvimento de algum algoritmo de roteamento mais sofisticado ao longo do trabalho caso seja verificada a necessidade de aprimoramento no método de “barramento virtual”.

3.3 Validação do sistema

Para ser possível mensurar o sucesso das regras e rotinas a serem implementadas, deverão ser realizados alguns testes para validar o desempenho geral do protocolo desenvolvido. Experimentos como a tomada de decisão do dispositivo de qual papel assumir, a entrada e saída de dispositivos da rede e o sucesso no repasse de pacotes para outros endereços estão entre os programados.

Para aferir a performance o principal indicador que deve ser analisado é a quantidade de pacotes de requisição enviados pelo *gateway* com resposta válida. É importante mencionar que, por tratar-se de rádios potentes com a capacidade de

propagação de dezenas de quilômetros, um ambiente controlado será utilizado para realizar todos os experimentos supracitados. Neste ambiente, a fim de reduzir fortemente a distância de propagação, configura-se o dispositivo com os parâmetros indicados pelo fabricante com a menor potência de dissipação possível e abdica-se do uso de antena. Estas duas medidas fazem com que a propagação do sinal LoRa pelo meio seja reduzido para ordem de centímetros tornando os testes em laboratório possíveis.

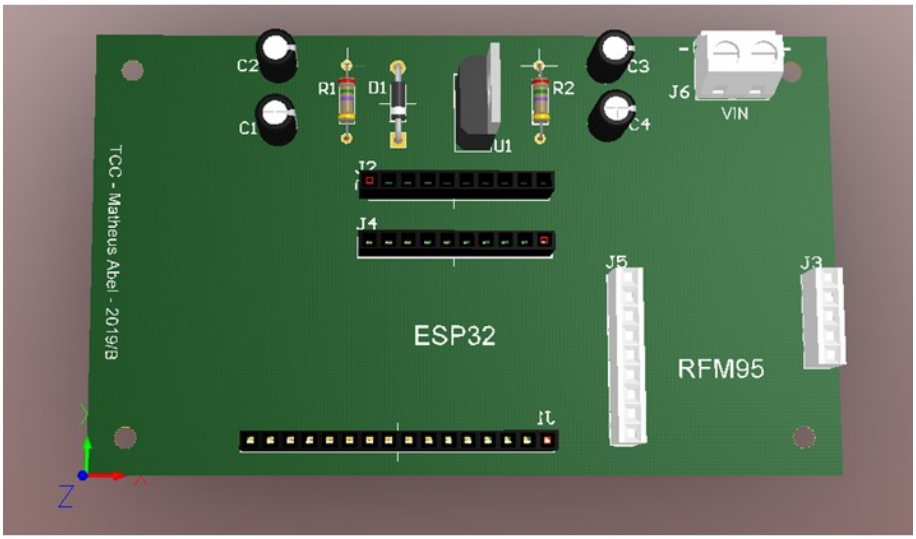
4 RESULTADOS E DISCUSSÃO

Ao longo do próximo capítulo serão apresentadas as etapas de desenvolvimento e os resultados obtidos para a proposta do trabalho.

4.1 *Hardware*

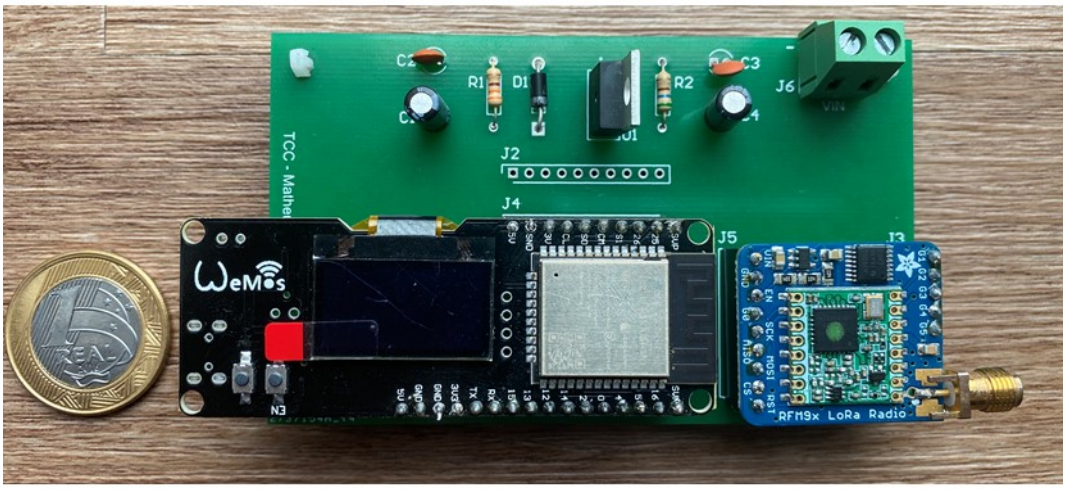
Como parte do trabalho, foi pretendida a criação de uma placa de validação própria, principalmente porque as disponíveis e acessíveis no mercado não ofereciam boa qualidade no que diz respeito ao rádio de modulação LoRa. A realização desta etapa pode ser vista de forma mais clara nas Figuras 25 e 26.

Figura 25 – Renderização do projeto da placa desenvolvida.



Fonte: Autor.

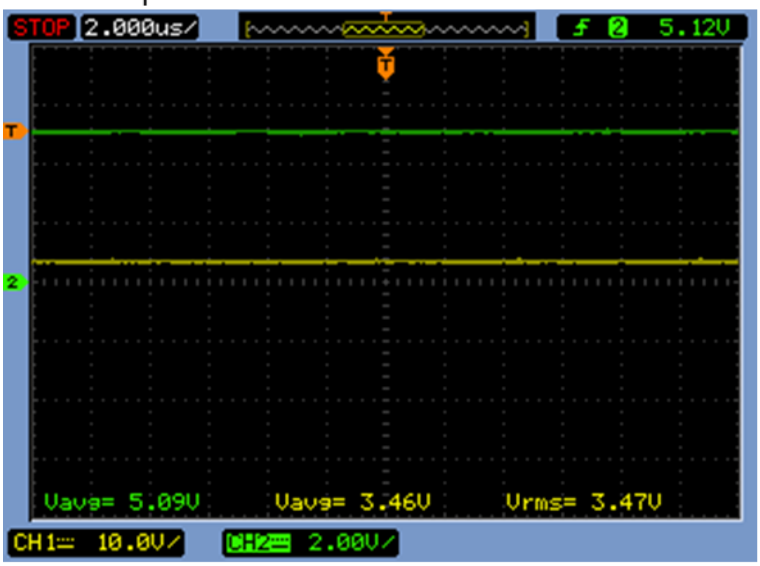
Figura 26 – Placa desenvolvida utilizada nos testes.



Fonte: Autor.

O determinante da qualidade do *hardware* desenvolvido é a alimentação tanto do módulo ESP32 quanto do RFM95, com um sinal tão constante quanto possível. Diferentemente dos módulos comerciais, que possuem somente um filtro de entrada, este esquemático conta com um filtro tanto na entrada quanto na saída, tornando-o ideal para seu propósito. A Figura 27 é uma captura da tela do osciloscópio, comprovando a estabilidade e a regulagem da tensão de saída (linha amarela) em relação a de entrada (linha verde).

Figura 27 – Captura das ondas obtidas no osciloscópio.



Fonte: Autor.

4.2 Rotinas para implementação das regras definidas

Esta seção apresenta os resultados obtidos na implementação das regras definidas no escopo do projeto e será apresentada em duas etapas, tratando o nó definido como *gateway* separadamente do nó regular.

Destaca-se que todos os dispositivos executam exatamente o mesmo programa, com isso, qualquer um é capaz de tornar-se um concentrador (*gateway*) desde que esteja ao alcance da rede sem fio configurada. É possível implementar esta condição visto que a tendência de aplicação é os nós estarem a distâncias longas uns dos outros, fazendo com que somente um conecte-se a rede sem fio e assuma o papel de concentrador.

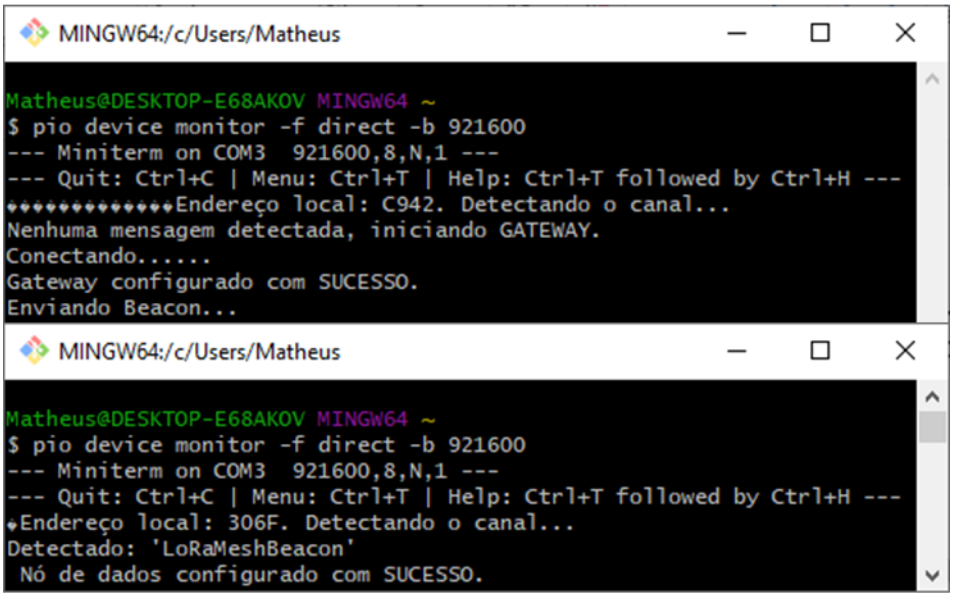
Ao longo das próximas figuras, para melhor compreensão, adotou-se um padrão de 5 cores de texto para identificação de cada ação tomada pelo dispositivo: Textos em (1) branco são registros genéricos ou mensagens que chegam ao nó; em (2) azul estarão representadas as mensagens enviadas pelo respectivo dispositivo; em (3) amarelo estão representados os alertas, na cor (4) vermelha estão expressos os erros e a cor de texto (5) verde representa sucesso na resposta da requisição.

4.2.1 Nó tipo *gateway*

De acordo com a Figura 17, do item 3.2.2.1, o nó tipo *gateway*, ao ser iniciado, deveria verificar a conexão com a *Wi-Fi* antes de iniciar suas tarefas, porém, devido a possibilidade de unificação do código descrita no projeto e discutida no último parágrafo do item 4.2 adicionou-se uma etapa que antecede este procedimento.

A nova etapa adicionada define que todo nó, ao ser iniciado, permanece por um período de tempo configurável varrendo o meio físico LoRa e, caso não seja detectado alguma mensagem do protocolo, assume-se que não existe uma rede em andamento e este nó passa a comportar-se como *gateway,* executando o fluxograma da Figura 17. Por outro lado, se uma mensagem for detectada, o dispositivo imediatamente assume papel de um nó regular da rede. A Figura 28 demonstra ambos os comportamentos supracitados.

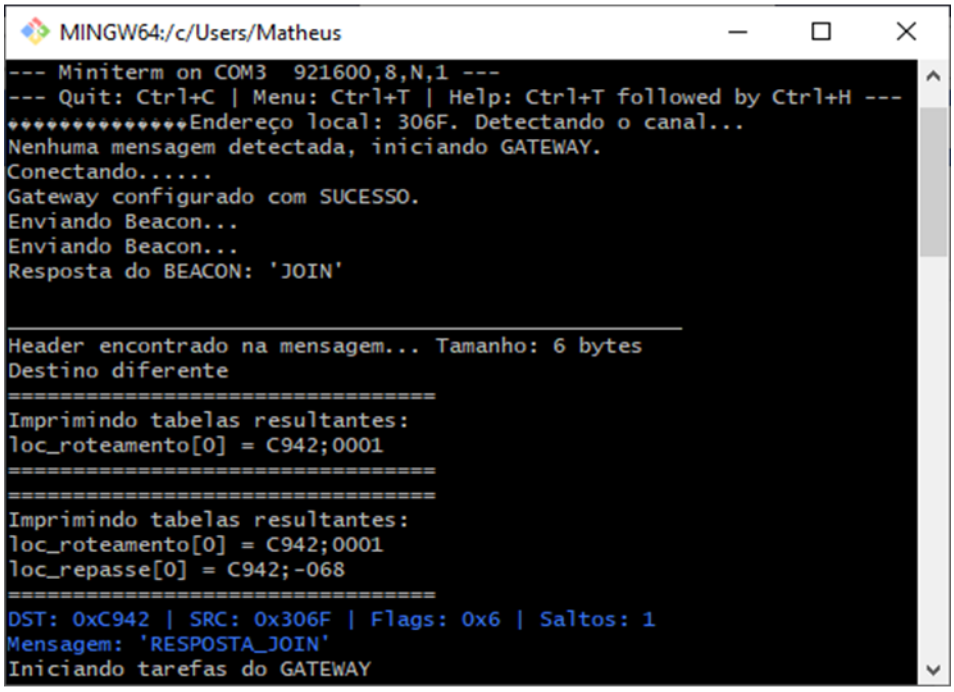
Figura 28 – Saída serial da etapa de configuração do nó para *gateway* ou operação regular.



Fonte: Autor.

Quando o *gateway* finaliza sua autoconfiguração, ele inicia o envio de *beacons* pelo rádio a fim de iniciar a rede até que um nó vizinho detecte a mensagem e dê início ao procedimento de associação. A Figura 28 mostra os registros por parte do *gateway* desta etapa da rede; Os procedimentos são o recebimento de um pacote do tipo *join* e, sendo um pacote válido, o endereço requerente é adicionado às tabelas de roteamento e repasse e, por fim, uma resposta endereçada ao requerente é enviada para que este inicie a operação como nó.

Figura 29 – Início da criação da rede, saída serial do *gateway*.



Fonte: Autor.

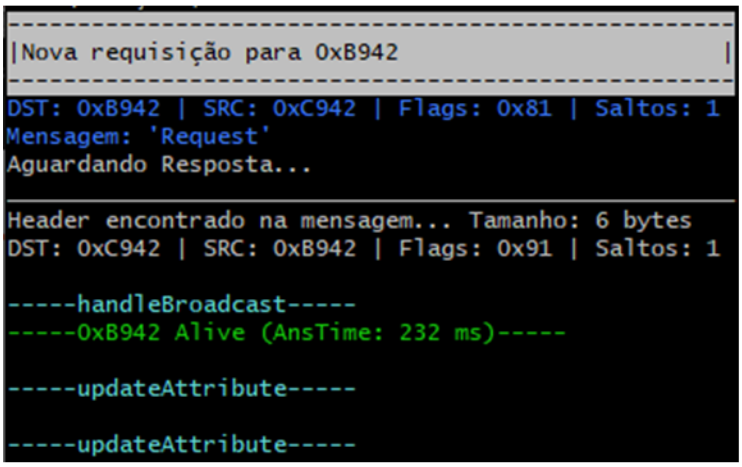
A partir do momento que este dispositivo possui um endereço em suas tabelas, inicia-se o envio das requisições para manutenção da rede. Em uma seção especial dos resultados serão expostos e discutidos resultados acerca do desempenho, como o tempo de resposta e a perda de pacotes em relação à quantidade de requisições por tempo.

4.2.1.1 Envio de uma requisição

A requisição para um nó associado ao *gateway* ocorre ao enviar um pacote endereçado e, para que ocorra a resposta, criou-se uma janela de espera de até 4 segundos (configurável via *software*). É importante mencionar que a janela de tempo não trava o dispositivo para o recebimento de outras mensagens pertinentes do protocolo como, por exemplo, um *broadcast* informando que há um novo endereço disponível. As Figura 30 e 31 mostram os dois desfechos para este procedimento, tanto de resposta positiva quanto de negativa por parte do nó requisitado, respectivamente. Na Figura 30 os textos destacados em azul ciano são as rotinas executadas pelo dispositivo ao detectar um pacote no canal; Primeiramente, verifica-

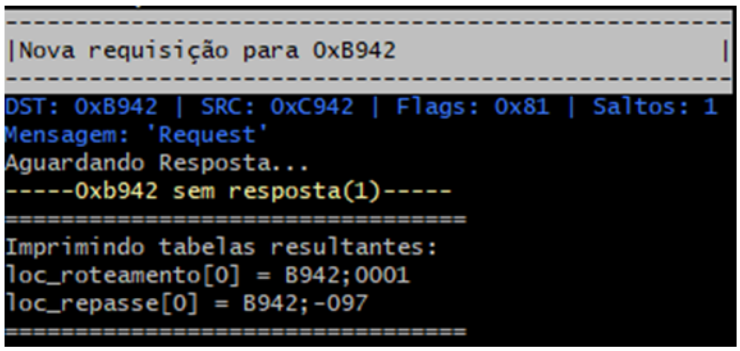
se a possibilidade de ser um *broadcast* e, após, se a mensagem é a resposta de requisição. Caso a mensagem seja a resposta, executa-se as rotinas para atualizar os parâmetros do respectivo endereço.

Figura 30 – Registro de uma requisição respondida.



Fonte: Autor.

Figura 31 – Registro de uma requisição com falha.



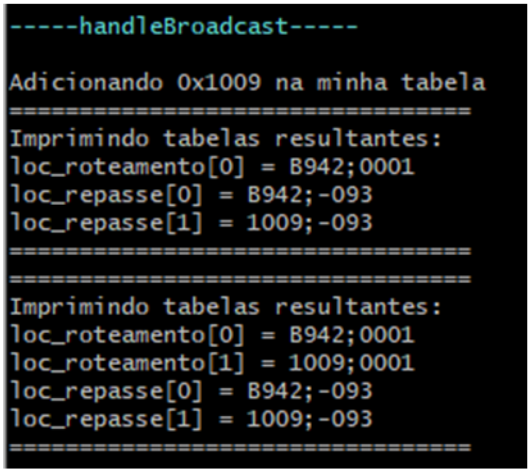
Fonte: Autor.

4.2.1.2 Cadastro de um novo nó

Os nós adjacentes ao *gateway* são capazes de resolver a entrada de novos nós na rede. O *gateway* irá adicionar novos endereços às suas tabelas na medida em que uma nova mensagem com características de novo nó seja recebida, ou seja, que contenha as *flags* IGN e ACT definidas como verdadeiras e sejam do tipo *broadcast*. Na Figura 32 pode-se verificar as tabelas resultantes na montagem de uma rede quando uma mensagem do tipo *broadcast* é detectada; Em concordância com o item

3.2.3 deste documento, a tabela de roteamento contém o número de saltos utilizados para realizar a comunicação com o respectivo endereço enquanto que a tabela de repasse guarda a intensidade do sinal capturado pelo rádio.

Figura 32 – *Gateway* recebendo e tratando mensagem *broadcast* que adiciona novo endereço a suas tabelas.

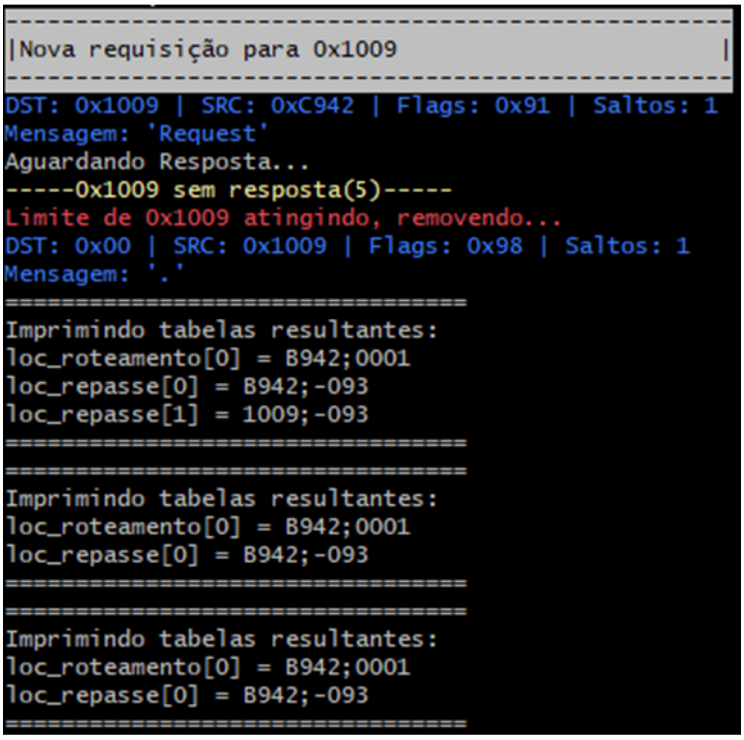


Fonte: Autor.

4.2.1.3 Estouro do limite de requisições com falhas

Foi definido que quando um endereço associado ao *gateway* atinge 5 requisições não respondidas ele deve ser considerado fora de alcance; Diante disso, ele é removido das tabelas de roteamento e repasse de todos os nós através de uma mensagem do tipo *broadcast*. A Figura 33 mostra os registros que o *gateway* exibe quando ocorre esta situação, removendo o endereço de sua própria tabela e inundando a rede com esta solicitação.

Figura 33 – Atingido o limite de requisições em falha o *gateway* inunda a rede com uma mensagem *broadcast* passando-se pelo endereço a ser apagado e acionando o bit “err” das *flags*.



Fonte: Autor.

4.2.2 Nó da rede

O item 4.2 descreveu uma nova característica adotada no projeto para todo o tipo de nó, trata-se do período ocioso que está a espera de uma mensagem LoRa para definição do tipo do nó. Para um nó tornar-se uma fonte de dados, deve-se ouvir um *beacon* enviado pelo *gateway*, no caso de ser o segundo nó ligado, ou, então, ouve-se uma mensagem com características válidas de rede (endereços de destino e de origem válidos, além de verificar se é uma mensagem do tipo ponto a ponto). No caso de um nó não detectar alguma mensagem, ele tenta conectar-se por 10s à rede *Wi-Fi* configurada e, por tratar-se de um nó que, provavelmente, estará fora do alcance do ponto de acesso a Internet, ele retorna ao modo de detecção do canal LoRa até que uma mensagem válida seja ouvida, como pode ser visto na Figura 34.

Figura 34 – Nó não detecta nenhuma mensagem no canal, tenta conectar-se a WiFi e falha, retornando a condição de nó comum para aguardar até uma mensagem válida ser detectada.

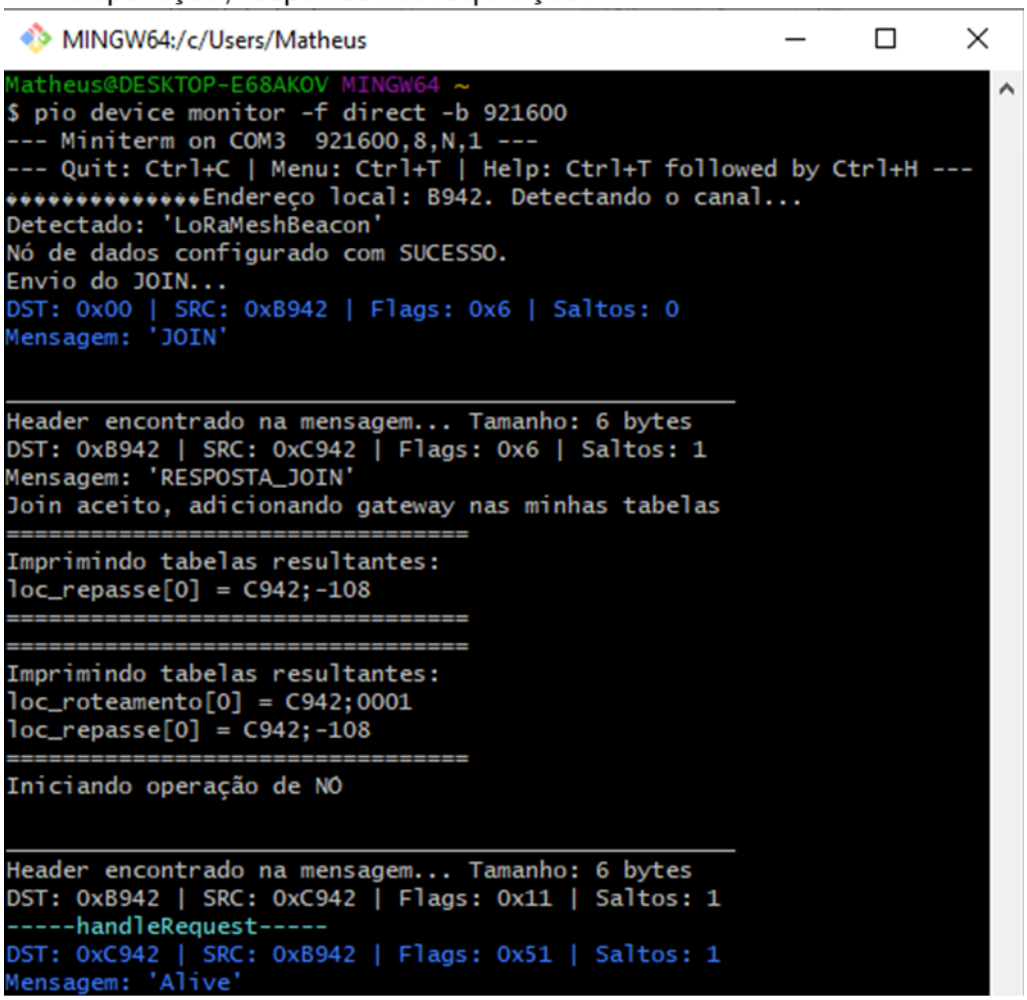
Fonte: Autor.

4.2.2.1 *Join* em uma rede existente

Esta seção prova, através da Figura 35, o procedimento que um nó adota ao se juntar a uma rede existente. A partir da detecção de uma mensagem válida do protocolo, com as mesmas características descritas na seção 4.2.2, um novo pacote é enviado com destino ao endereço detectado e contendo as caraterísticas previamente definidas para um *join* (*flags* IGN e ACT verdadeiras). A partir do envio do pacote, o equipamento passa a aguardar o retorno do pedido para iniciar sua operação corriqueira.

Vale mencionar que o período de aguardo do retorno de um *join* é contabilizado através de 10 mensagens, ou seja, podem ser identificadas até 10 mensagens pertinentes do protocolo, mas que não sejam necessariamente a resposta pretendida. Se este limite é atingido, o *join* é considerado sem sucesso fazendo o dispositivo reiniciar.

Figura 35 – Nó detecta atividade no canal, envia pacote do tipo *join* e inicia operação, respondendo requisição.



Fonte: Autor.

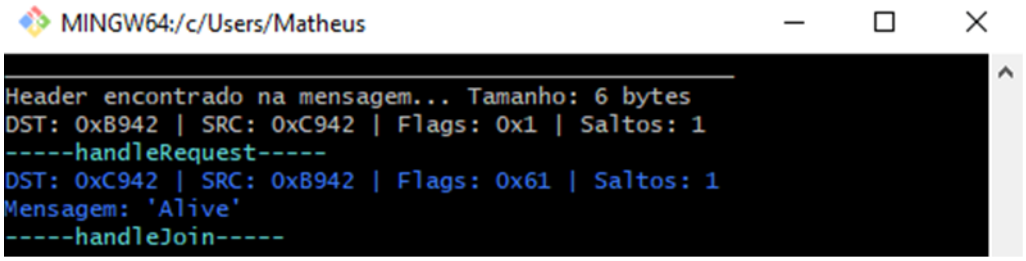
4.2.2.2 Tratamentos das mensagens em operação regular

Uma vez que o dispositivo se associou a uma nova rede, seu canal LoRa permanece aberto todo o tempo para novos pacotes do protocolo. São três os tipos de mensagens que podem ser recebidas, uma requisição direta, uma mensagem do tipo *broadcast* ou, então, uma mensagem que deve ser encaminhada. Foram desenvolvidas três rotinas diferentes para tratamento dos pacotes recebidos, sendo que elas são executadas baseado nas condições de identificação do tipo do pacote.

Um pacote de requisição é identificado através de duas condições, sendo o endereço de destino o endereço do nó que a recebeu e seu conteúdo sendo "*Request*". Este pacote é respondido pelo dispositivo, gerando um novo pacote

contendo o endereço de destino, o nó requerente (*gateway*), e o conteúdo “*Alive*”, como consta na Figura 36.

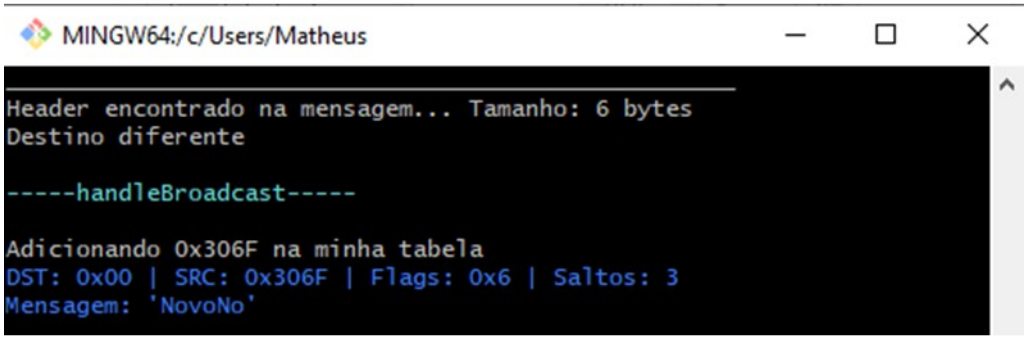
Figura 36 – Identificação de requisição (branco) e sua resposta (azul).



Fonte: Autor.

Quando um pacote recebido contém o endereço de destino igual a zero, este é considerado *broadcast* e invoca a rotina para tratamento do pacote. A rotina interpreta o pacote, utilizando tanto para adição quanto remoção de um endereço às tabelas locais. Na Figura 37 pode-se observar a adição de um endereço, no caso de remoção o processo é o mesmo e, portanto, foi suprimido. O texto em azul ciano indica a detecção da mensagem do tipo *broadcast*, depois, pode-se notar a interpretação da mensagem, com o endereço “0x306F” sendo adicionado e, por fim, a mensagem recebida é repassada adiante (azul).

Figura 37 – Demonstração de uma mensagem *broadcast* sendo tratada por um nó.

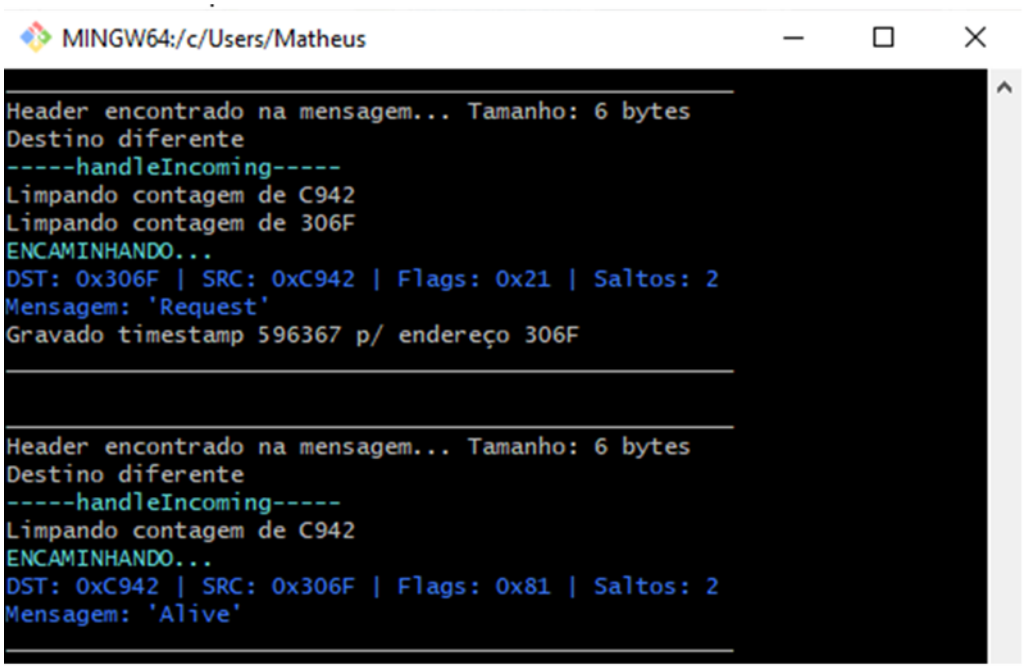


Fonte: Autor.

O último tipo de mensagem que pode chegar a um nó regular é um pacote no qual o destino é diferente do endereço local; Estes pacotes devem ser elegíveis para encaminhamento. Como consta na Figura 38, um pacote foi detectado e enviado

novamente pelo canal LoRa. Inicialmente projetou-se o número máximo de saltos configurados e se o conteúdo da mensagem em questão não era igual ao ultimo recebido como condição para envio.

Figura 38 – Demonstração de uma requisição com destino diferente sendo tratada por um nó.



Fonte: Autor.

Estes dois requisitos para encaminhamento fizeram com que, nos casos onde havia mais de um nó com conexão direta, a mensagem pudesse ficar transitando entre os nós, até que o estouro por saltos a anulasse.

Para reproduzir este comportamento, posicionou-se quatro dispositivos próximos ao *gateway*, desta forma, quando o concentrador de dados faz a requisição, seu destino é atingido com apenas um salto e, portanto, a resposta é recebida logo em seguida, sem a necessidade de nenhum encaminhamento. Os nós vizinhos encaminham o pacote *request* de qualquer forma, fazendo com que três novas mensagens aparecessem no canal; Ao mesmo tempo, a mensagem de resposta da requisição também foi recebida, sendo enviada novamente, pois ela é diferente da última encaminhada (*request*), e este ciclo permanece até que o limite de saltos seja atingido. Este tipo de reação fez com que, em alguns casos, os nós perdessem as mensagens pertinentes ao seu endereço em função do congestionamento do canal.

Tendo em vista este comportamento e com o objetivo de mitigar a perda de pacotes pertinentes, implementou-se uma nova funcionalidade para o encaminhamento que consiste em verificar a estampa temporal em que foi encaminhada uma mensagem do respectivo endereço. Para exemplificar este procedimento, toma-se como base a Figura 38, na qual cada endereço cadastrado nas tabelas do dispositivo contém uma estampa temporal e um contador de mensagens encaminhadas. Em um determinado nó, uma requisição essencialmente engloba duas mensagens constando aquele endereço, o envio e a resposta. Com base nisso, por exemplo, a primeira vez que uma mensagem contendo o endereço 0x306F é identificada, uma estampa temporal é associada a este endereço e encaminha-se somente mais uma mensagem (provável resposta) nos próximos segundos. A quantidade de segundos até que a estampa temporal seja renovada para que se possa encaminhar 2 novas mensagens correspondentes àquele nó é determinada pelo intervalo entre as requisições configurado previamente, tomando vantagem da unicidade do código em todos os dispositivos.

Então, tem-se três condições para encaminhamento da mensagem em ordem de relevância: número de saltos, última mensagem encaminhada e estampa temporal da última mensagem encaminhada, além de, obviamente, não encaminhar mensagens do seu próprio endereço.

4.3 Desempenho geral

As próximas seções expõem alguns parâmetros extraídos dos testes realizados, conforme programado no item 3.3.

4.3.1 Criação de rede com todos os nós ao alcance

Este teste consistiu na criação da rede com 5 dispositivos onde cada um era ligado após o sucesso da entrada do seu antecessor. Para fins de comparação, foram definidos dois intervalos diferentes entre as requisições, como mostra o Quadro 1.

Quadro 1 – Dados coletados no experimento.

		Requisições			
		25	**250**	**500**	**1000**
Intervalo de 1s	**Pacotes perdidos**	2	3	34	120
	Perdas (%)	8,0%	1,2%	6,8%	12,0%
Intervalo de 5 s	**Pacotes perdidos**	2	2	2	2
	Perdas (%)	8,0%	0,8%	0,4%	0,2%

Fonte: autor.

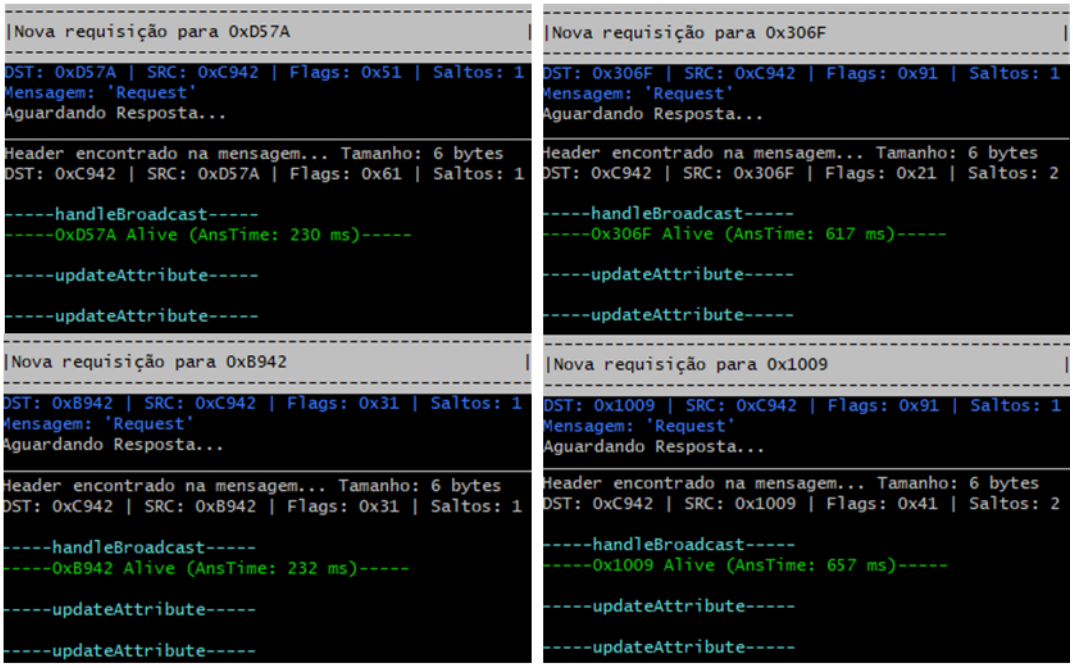
É possível observar que para esta quantidade de dispositivos o intervalo de 5 segundos tornou a perda praticamente nula a longo prazo, mostrando ser um bom parâmetro para rede com este tamanho. É interessante destacar que, a partir das informações compiladas, pôde-se prever para trabalhos futuros o desenvolvimento de um algoritmo para determinar o melhor tempo de requisição baseado na quantidade de nós associados, visando atingir a maior estabilidade ao longo do tempo de operação.

4.3.2 Criação de rede mista

Com este experimento buscou-se elucidar o comportamento do protocolo quando uma rede com cinco dispositivos distribuídos (de forma que dois estejam com alcance direto ao *gateway* e dois sejam dependentes de saltos para receber e responder suas requisições) formam uma topologia *mesh* mista. Os tempos de resposta de cada endereço e o número de saltos de distância constam na Figura 39.

Em geral o comportamento foi estável e foram realizadas coletas de informações a 150, 300 e 450 requisições no *gateway, o* que retornou um índice de falha médio de 7,06%. Ao comparar a porcentagem de perda com as tecnologias de transmissão de internet sem fio, por exemplo, o valor referente é considerado alto. Comparar, porém, os tratamentos oferecidos no *Wi-Fi*, que contemplam diversas regras de compensação de perda e evasão de colisão, não é ideal visto que apenas foi implementado para este trabalho a regra de "*listen-before-talk".* Desta forma é correto afirmar que a taxa de perda do projeto implantado é aceitável.

Figura 39 – Captura dos registros do *gateway* para rede mista.

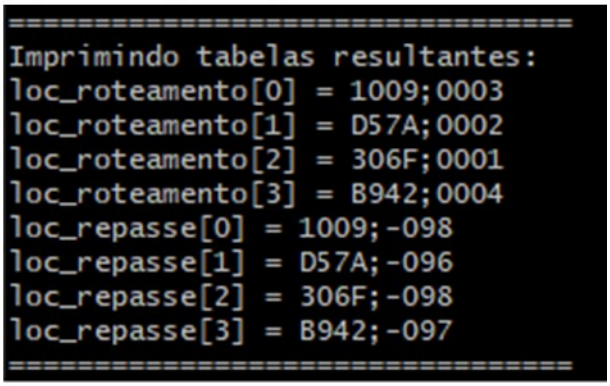


Fonte: Autor.

4.3.3 Rede sem caminhos redundantes

Por último, fez-se uma avaliação de uma rede montada de forma que houvesse apenas um caminho para os pacotes percorrerem. Com isso, buscou-se simular o pior cenário, uma vez que não há redundância de caminhos para as requisições realizadas pelo *gateway*.

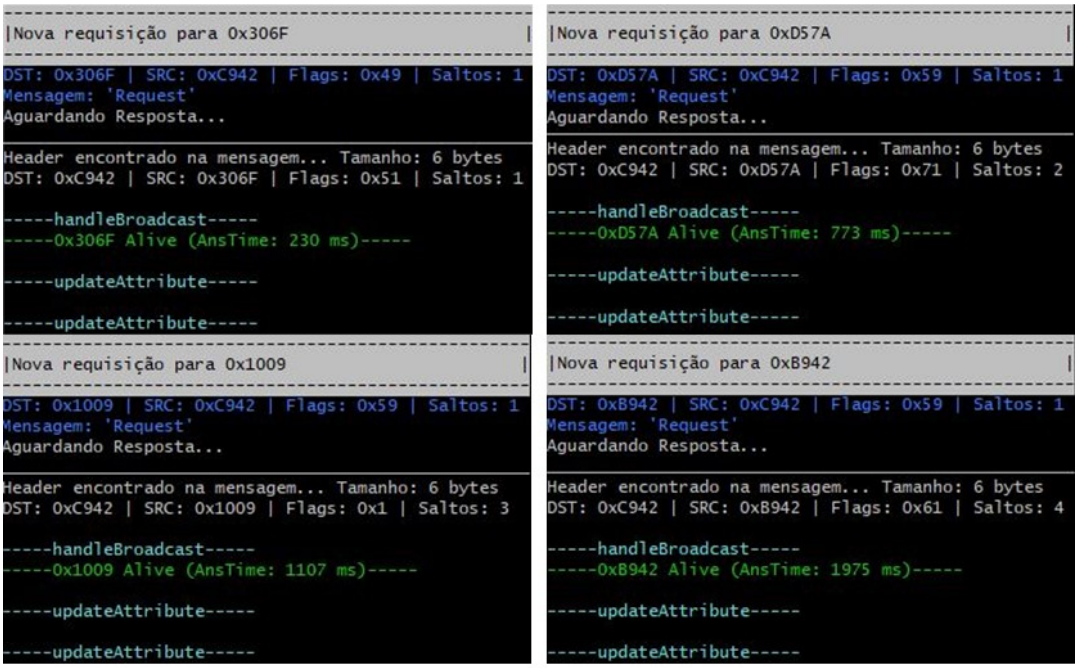
Figura 40 – Tabelas de roteamento e repasse do *gateway*.



Fonte: Autor.

A Figura 40 mostra as tabelas obtidas no final da montagem da rede. É válido mencionar que a intensidade do sinal registrada para cada endereço é referente ao nó que realizou o último salto, daí vem a semelhança entre todos os valores.

Figura 41 – Captura dos registros do *gateway* para endereços sem redundância.



Fonte: Autor.

A Figura 41 exprime, do ponto de vista do *gateway*, as respostas obtidas com as requisições para os nós a 1, 2, 3 e 4 saltos de distância. O objetivo principal deste experimento foi verificar a influência dos saltos no tempo de resposta. Pôde-se notar que no equipamento a dois saltos houve acréscimo de 500 milissegundos em relação ao nó com apenas um salto. Ao analisar o nó com resposta obtida em três saltos foram adicionados 430 milissegundos, aproximadamente; E a resposta com quatro saltos adicionou cerca de 800 milissegundos. Neste cenário, o índice de perda de pacotes registrado foi de cerca de 11,6% após 500 requisições espaçadas em 5 segundos.

5 CONCLUSÃO

A partir da apresentação dos resultados e das discussões no capítulo anterior pode-se afirmar que os objetivos do trabalho foram plenamente atingidos. O circuito desenvolvido foi capaz de entregar a estabilidade necessária para alimentação do microcontrolador e do rádio LoRa. Em adição, o protótipo mostrou-se um facilitador para realização dos testes, por oferecer, principalmente, flexibilidade no manuseio dos principais componentes do projeto e, também, por prover duas opções para fornecimento de energia aos mesmos.

Assim como a porção de *hardware* do escopo do projeto, a parcela de *software* também mostrou desempenho satisfatório. O cabeçalho desenvolvido agiu como facilitador na implementação das regras mais complexas, como por exemplo, a tomada de decisão para encaminhar as mensagens. Inicialmente as tabelas dos dispositivos, as quais guardam as informações dos dispositivos adjacentes em conjunto com os parâmetros como intensidade do sinal e quantidade de saltos, foram concebidas para serem armazenadas na memória *flash* do dispositivo, contudo, a característica do projeto de buscar o mínimo de perda possível de mensagens fez com que elas deixassem de ser gravadas, principalmente pelo atraso que a escrita na memória *flash* provê. Outro fator que influenciou na decisão desta alteração foi o bom desempenho verificado na rotina para entrada e saída da rede, por isso, verificou-se que não era mais relevante armazenar informações quando um dispositivo era desligado.

A rotina de encaminhamento das mensagens mostrou-se eficaz desde a sua concepção inicial, sendo capaz de fornecer ao protocolo uma topologia lógica do tipo *mesh*. A principal dificuldade relacionada a este tópico foi devido à tomada de decisão de quando não encaminhar mensagens. Para que se tornasse mais robusto o

encaminhamento, no sentido de realizar mais requisições com sucesso, a implementação da estampa temporal nos pontos ordinários da rede em conjunto com a inclusão de um *token* gerado na origem e na resposta das requisições fez com que menos mensagens circulassem no meio físico, acarretando em menos mensagens perdidas pelos dispositivos.

Pôde-se constatar a eficiência geral do conjunto desenvolvido através do índice de perda de pacotes médio que atingiu 11,6% no pior cenário testado, com índice de recepção próximo a -100 dB e realizando até 4 saltos para completar uma requisição. Para casos em que todos os nós estão ao alcance, o protocolo mostrou-se ainda mais eficiente, com perdas inferiores à 1%. Por último, através da medição de tempo para resposta de cada requisição, conclui-se que cada salto adiciona cerca de 550 milissegundos neste parâmetro, sendo plenamente satisfatório para aplicações de longa distância com fornecimento de dados não contínuos.

O projeto foi concebido para realizar comunicação de longa distância e, portanto, as regras definidas foram pautadas nesta premissa. Deve-se mencionar que apesar da concepção inicial para utilização com um módulo de rádio LoRa, nada impede que o mesmo protocolo seja utilizado por outro rádio disponível no mercado, bastando alterar o *driver* responsável pela comunicação com o microcontrolador. Do ponto de vista de engenharia esta característica se torna importante, pois garante portabilidade e escalabilidade para outros desafios que possam ser beneficiados por uma topologia *mesh*.

REFERÊNCIAS BIBLIOGRÁFICAS

ABINC. **Previsões para o Mercado de IOT – ABINC**. Disponível em: <https://abinc.org.br/previsoes-para-o-mercado-de-iot/>. Acesso em: 4 abr. 2019.

AKYILDIZ, I. F.; WANG, X.; WANG, W. Wireless mesh network: A survey. **Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016**, v. 47, p. 1966–1970, 2016.

AUGUSTIN, A. et al. A study of Lora: Long range & low power networks for the internet of things. **Sensors (Switzerland)**, v. 16, n. 9, p. 1–18, 2016.

ESIGN, C. R. A. D.; AWADIA, V. I. K.; ECHNOLOGIES, B. B. N. T. A Cautionary Perspective on Cross-Layer Design. **Ieee Wireless Communications**, n. February, p. 3–11, 2005.

FACCIN, S. M. et al. Wireless Mesh Networking Mesh Wlan Networks : Concept and System Design. **Ieee Wireless Communications**, April 2006, p. 10–17, 2008.

HUANG, L.; LAI, T.-H. **On the scalability of IEEE 802.11 ad hoc networks**. Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing - MobiHoc '02. **Anais**...New York, New York, USA: ACM Press, 2002. Disponível em: <http://portal.acm.org/citation.cfm?doid=513800.513822>

INTERSABERES (Org). **Redes**. Curitiba: Editora InterSaberes, 2014.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet:** uma abordagem top-down. 6. ed. São Paulo: Pearson Education do Brasil, 2013.

LIMA FILHO, E. C. **Fundamentos de Redes e Cabeamento Estruturado**. São Paulo: Pearson, 2015.

LINKLABS. **Low Power, Wide Area Networks networks (LPWANs).** Annapolis: Link Labs, 2017.

LORA ALLIANCE. **A technical overview of LoRa ® and LoRaWAN ™**

LoRaWAN ™ What is it? San Ramon: Lora Alliance, 2015.

LORA ALLIANCE TECHNICAL COMMITEE. LoRaWAN 1.1 Specification. **LoRaWAN 1.1 Specification**, n. 1.1, p. 101, 2017.

LUETH, K. L. **State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating**. Disponível em: <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>. Acesso em: 4 abr. 2019.

MEKKI, K. et al. A comparative study of LPWAN technologies for large-scale IoT deployment. **ICT Express**, v. 5, n. 1, p. 1–7, mar. 2019.

MICROSOFT. **Self Organizing Wireless Mesh Networks - Microsoft Research**. Disponível em: <https://www.microsoft.com/en-us/research/project/self-organizing-wireless-mesh-networks/?from=http%3A%2F%2Fresearch.microsoft.com%2Fmesh%2F>. Acesso em: 3 abr. 2019.

SEMTECH. LoRa™Modulation Basics. v. 2. Camarillo: Semtech, 2015.

WATTEYNE, T. et al. Understanding the Limits of LoRaWAN. **IEEE Communications Magazine**, v. 55, n. 9, p. 34–40, 2017.

WHITMORE, A.; AGARWAL, A.; DA XU, L. The Internet of Things—A survey of topics and trends. **Information Systems Frontiers**, v. 17, n. 2, p. 261–274, 12 abr. 2015.

XU, L. DA; HE, W.; LI, S. Internet of Things in Industries: A Survey. **IEEE Transactions on Industrial Informatics**, v. 10, n. 4, p. 2233–2243, nov. 2014.