UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FLOATINGBLUE: UMA ARQUITETURA DE RSSF DE BAIXO CONSUMO NÃO INFRAESTRUTURADA BASEADA EM REDES TOLERANTES A ATRASOS E DESCONEXÕES E *BLUETOOTH LOW ENERGY*

Ruan Carlos Mota Teixeira

Manaus – Amazonas

Janeiro de 2024

Ruan Carlos Mota Teixeira

FLOATINGBLUE: UMA ARQUITETURA DE RSSF DE BAIXO CONSUMO NÃO INFRAESTRUTURADA BASEADA EM REDES TOLERANTES A ATRASOS E DESCONEXÕES E *BLUETOOTH LOW ENERGY*

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas de Controle e de Automação Modernos.

Prof. D.Sc. Celso Barbosa Carvalhos (orientador)

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Teixeira, Ruan Carlos Mota
T266f FloatingBlue: uma arquitetura de RSSF de baixo consumo não infraestruturada baseada em redes tolerantes a atrasos e desconexões e Bluetooth Low Energy / Ruan Carlos Mota Teixeira . 2024
79 f.: il. color; 31 cm.

Orientador: Celso Barbosa Carvalho
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Redes tolerantes a atrasos e desconexões. 2. Bluetooth Low Energy. 3. Redes de Sensores Sem Fio. 4. Monitoramento ambiental. 5. Redes não infraestruturadas. I. Carvalho, Celso Barbosa. II. Universidade Federal do Amazonas III. Título

Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

RUAN CARLOS MOTA TEIXEIRA

FLOATINGBLUE: UMA ARQUITETURA DE RSSF DE BAIXO CONSUMO NÃO INFRAESTRUTURADA BASEADA EM REDES TOLERANTES A ATRASOS E DESCONEXÕES E BLUETOOTH LOW ENERGY

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 26 de janeiro de 2024.

BANCA EXAMINADORA

Prof. Dr. Celso Barbosa Carvalho
Presidente
Universidade Federal do Amazonas

Assinado ICP Brasil D4Sign

Prof. Dr. Angilberto Muniz Ferreira Sobrinho, Membro
Universidade Estadual do Amazonas

Assinado ICP Brasil D4Sign

Prof. Dr. Fabio de Sousa Cardoso, Membro
Universidade Estadual do Amazonas

PPGEE Programa de Pós-Graduação em Engenharia Elétrica - UFAM
Pós-Graduação em Engenharia Elétrica.
Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Campus Universitário, Setor Norte - Coroado, Pavilhão do CETELI.
Fone/Fax (92) 99271-8954 Ramal:2607. E-mail: ppgee@ufam.edu.br

Resumo da Dissertação apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

FLOATINGBLUE: UMA ARQUITETURA DE RSSF DE BAIXO CONSUMO NÃO INFRAESTRUTURADA BASEADA EM REDES TOLERANTES A ATRASOS E DESCONEXÕES E *BLUETOOTH LOW ENERGY*

Ruan Carlos Mota Teixeira

Orientador: Prof. D.Sc. Celso Barbosa Carvalhos

Programa: Pós-Graduação em Engenharia Elétrica

A implementação de RSSF em áreas amplas e remotas apresenta desafios, como a necessidade de baixo consumo de energia e alcance de comunicação de longo alcance. Para superar esses desafios, neste trabalho é proposta uma nova arquitetura de rede não infraestruturada chamada *FloatingBlue*, que combina nós concentradores móveis (ex: *drones*) equipados com comunicação baseada em tecnologias tolerante a atraso e desconexões, utilizando o *Bundle Protocol 7* e nós de terminação de baixo consumo de energia usando o protocolo Bluetooth Low Energy (BLE). A arquitetura proposta visa aumentar a cobertura da rede e reduzir o consumo energético. O trabalho apresentado descreve a integração desses protocolos em um módulo computacional, o desenvolvimento de *firmware* para os nós de terminação e sua aplicação no monitoramento de uma área agrícola. Os resultados obtidos com a utilização da *FloatingBlue* demonstram êxito na coleta e transmissão de parâmetros ambientais em um ambiente de grande extensão e cenário de longa desconexão. Além disso, foram identificados resultados satisfatórios de consumo energético dos nós de terminação: 42 μJ em modo de transmissão de mensagens e 13 μJ em modo ocioso. Palavras-chave:. Redes tolerantes a atrasos e desconexões, Bluetooth Low Energy, Redes de Sensores Sem Fio, Monitoramento ambiental, Redes não infraestruturadas.

Abstract of Dissertation presented to UFAM as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

FLOATINGBLUE: AN ARCHITECTURE OF LOW-POWER INFRASTRUCTURELESS WSN BASED ON DELAY AND DISRUPTION-TOLERANT NETWORKS AND BLUETOOTH LOW ENERGY

Ruan Carlos Mota Teixeira

Advisor: Prof. D.Sc. Celso Barbosa Carvalhos

Department: Postgraduate in Electrical Engineering

The implementation of WSN (Wireless Sensor Network) in large and remote areas poses challenges, such as the need for low power consumption and long-range communication reach. To overcome these challenges, this work proposes a new infrastructure-less network architecture called FloatingBlue, which combines mobile concentrator nodes (e.g., drones) equipped with delay-tolerant and disruption-tolerant communication technologies using the Bundle Protocol 7, and low-power termination nodes using Bluetooth Low Energy (BLE) protocol. The proposed architecture aims to increase network coverage and reduce energy consumption. The presented work describes the integration of these protocols into a computational module, the development of firmware for termination nodes, and their application in monitoring an agricultural area. The results obtained using FloatingBlue demonstrate success in collecting and transmitting environmental parameters in a vast area with a prolonged disconnection scenario. Additionally, satisfactory energy consumption results were identified for termination nodes: 42 μJ in transmission mode of bundles and 13 μJ in idle mode.

Keywords: Delay Tolerant Network, Bluetooth Low Energy, Wireless Sensor Networks, Environmental monitoring, Infrastructureless Network

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AA Agentes de Aplicação

ADU *Application Data Units*

AES *Advanced Encryption Standard*

API *Application Pro- gram Interfac*

ATT Protocolo de Atributo

BP *Bundle Protocol*

BPA *Bundle Protocol Agent*

CL Convergence Layer

CLA Convergence Layer Adapter

CM Concentradores Móveis

CoAP *Constrained Application Protocol*

CRC *Cyclic Redundancy Check*

DFU *Device Firmware Update*

DSP *Digital Signal Processor*

DTN *Delay Tolerant Network*

EID *Endpoint Identifiers*

EN *End Node*

FHA *Frequency Hopping Adaptative*

FSK *Frequency Shift Keying*

GAP *Generic Access Profile*

GATT *Generic Attribute Profile*

GFSK *Gaussian Frequency Shift Keying*

HCI *Host Controller Interface*

I2C *Inter-Integrated Circuit*

IEEE *Institute of Electrical and Electronics Engineers*

IoT *Internet of Things*

ISM *Industrial, Scientific, and Medical*

JSON *JavaScript Object Notation*

L2CAP *Logical Link Control and Adaptation Protocol*

LPWAN *Low Power Wide Area Network*

MQTT *Message Queuing Telemetry Transport*

OS *Operating System*

PDA *Public device address*

PDU *Protocol Data Unit*

RCF *Request for Comments*

RDA *Random Device Address*

RSSF Redes de Sensores Sem Fio

RTOS *Real-Time Operating Systems*

SDIO *Secure Digital Input Output*

SDK *Software Development Kit*

SIG *Special Interest Group*

SM *Security Manager*

SoC *System-on-a-chip*

SPI *Serial Peripheral Interface*

TCP-IP *Transmission Control Proto- col/Internet Protocol*

TOML *Tom's Obvious Minimal Language*

UART *Universal Asynchronous Receiver Transmitter*

USB *Universal Serial Bus*

UWB *Ultra Wideband*

VANT Veículo Aéreo Não Tripulado

WuR *Wake-up Radio*

Sumário

Capítulo 1

Introdução

A fim de realizar operações de forma mais rápida e assertiva, nota-se que a sociedade busca realizar monitoramento e controle autônomo de parâmetros físicos percebidos pelos diversos objetos do cotidiano, com aplicações em diversos contextos como indústria, agricultura, saúde e bem-estar. Para que o cenário de monitoramento e controle autônomo ocorra, é necessário que objetos e ambientes do cotidiano sejam equipados com comunicação em rede que possibilitem que informações de sensoriamento e controle cheguem respectivamente ao usuário e objeto controlado. Nesse contexto de transformação digital, as tecnologias de comunicação sem fio impulsionam e, também, são impulsionadas devido a praticidade e a mobilidade da operação e o baixo custo operacional da instalação e manutenção [1].

Entretanto, a utilização de Redes de Sensores Sem Fio (RSSF) para monitoramento e controle em áreas amplas e remotas traz alguns desafios. As principais problemáticas incluem a necessidade de: i) nós de terminação (sensores ou atuadores) com baixo consumo de energia, visto que geralmente são alimentados por bateria; ii) longo alcance da comunicação, para cobrir áreas amplas utilizando uma baixa densidade de nós. Nessa perspectiva, existem as tecnologias LPWAN (*Low-Power Wide-Area Network*) que possibilitam alcance de comunicação de dezenas de quilômetros e baixo consumo dos transceptores de rádio [2].

Utilizando a arquitetura LPWAN e, a fim de reduzir o consumo energético, através da diminuição da vazão de dados no canal de comunicação entre sensores e *gateway*, existem trabalhos que sugerem a utilização de nós especiais chamados de concentradores, que coletam a informação dos nós de terminação e encaminham

para o *gateway* [3, 4]. Outros trabalhos [5, 6] utilizam a possibilidade de diminuir, ou anular, a utilização de *gateways*, por meio do emprego de nós concentradores que coletam e encaminham, em múltiplos saltos, os dados agregados para outros nós concentradores até a entrega dos dados à estação central de processamento. Entretanto, comenta-se que as LPWANs são tecnologias que geralmente necessitam de rede infraestruturada com utilização de *gateways* para concentrar os dados coletados pelos sensores e, posteriormente encaminhar tais dados para uma estação remota de monitoramento e controle [7].

Outras pesquisas [8–10] utilizam como estratégia a utilização de nós concentradores móveis que se comunicam em múltiplos saltos e se deslocam até os nós de terminação, possibilitando aumentar a área de cobertura da rede de comunicação. Contudo, com este método, a comunicação entre os nós concentradores fica suscetível a desconexões [8]. Uma estratégia para estes casos é a utilização de DTN (*Delay Tolerant Network*), aptas a operar em situações de comunicação com grandes latências e desconexões entre os nós sem fio. Alguns trabalhos utilizam essa tecnologia, por meio de nós concentradores acoplados a veículos aéreos não tripuláveis (VANT), para comunicação em cenários de catástrofe, onde a infraestrutura de rede foi danificada [9, 10]. Contudo, não é proveitosa a utilização das tecnologias DTN em dispositivos de terminação uma RSSF, uma vez que os nós sensores necessitariam de hardware com grande poder de processamento e elevado consumo para operarem como nós DTN [11–13].

Neste trabalho, foi notada a existência de oportunidades para o emprego de nós de terminação equipados com tecnologias de comunicação de baixo consumo energético e curta distância, como o *Bluetooth Low Energy* (BLE). Por possuir consumo de energia reduzido, o BLE é uma das tecnologias pilares da RSSF, apesar de ser uma tecnologia de curto alcance, ela oferece uma comunicação com maior taxa de transmissão de dados, quando comparado com as LPWANs [14]. Além disso, um nó BLE pode fornecer uma alternativa de menor consumo energético, por meio do seu modo de operação *broadcast*, situação na qual o protocolo permite a comunicação sem a necessidade de conexão para transmitir pacotes e, consequentemente gastar energia adicional para estabelecer o pareamento com outros dispositivos BLE da rede [15].

Dessa forma, é proposto o desenvolvimento de uma nova arquitetura de RSSF não infraestruturada nomeada de *FloatingBlue* que realiza simultaneamente os objetivos de: i) aumentar a cobertura da rede através da utilização de nós concentradores móveis que se comunicam por meio de protocolos DTN; ii) reduzir o consumo energético dos nós de terminação por meio da utilização do protocolo BLE para comunicação entre eles e nós concentradores móveis. O objetivo do presente trabalho é contribuir com o estado da arte em oferecer alternativas de arquitetura de rede para RSSF não infraestruturadas e de cobertura ampla que combine comunicação confiável e resiliente, com nós de terminação de baixo consumo. O trabalho também possui como motivação o elevado custo operacional e a manutenibilidade no monitoramento de áreas amplas e remotas, como florestas, grandes latifúndios e o setor hídrico.

Para desenvolvimento da solução, foi proposta e implementada a integração entre os protocolos BLE e DTN, em um módulo computacional. Além, disso, foi realizada a implementação de software do tipo *firmware* para o nó de terminação empregado em cenário de monitoramento ambiental. Como prova de conceito, a solução foi aplicada no monitoramento de uma área agrícola. Ao longo do trabalho, serão descritos os detalhes da solução proposta e das aplicações de software embarcado desenvolvidas.

1.1 Objetivo Geral

Implementação de uma arquitetura de RSSF não infraestruturada, de baixo consumo e cobertura ampla, aplicável ao monitoramento de áreas amplas e remotas, baseada em nós de terminação *Bluetooth Low Energy* e nós concentradores móveis sujeitos a atrasos e desconexões.

1.2 Objetivo Especifico

1. Elaborar documento que contenha levantamento de trabalhos do estado da arte sobre alternativas de RSSF para área ampla e baixo consumo dos nós de terminação;

2. Implementar, utilizando kits de desenvolvimento BLE, *firmware* para nós de terminação/sensores aplicáveis a cenários de monitoramento ambiental e com rotinas de software que otimizem o consumo do transceptor BLE;

3. Desenvolver módulos de software que executem serviços de gerenciamento e integração dos protocolos BLE e DTN em módulo computacional dos dispositivos concentradores sem fio;

4. Implementar uma RSSF, utilizando a arquitetura proposta, em um estudo de caso para monitoramento de áreas agrícolas;

5. Realizar a análise do consumo energético dos nós de terminação no contexto da proposta *FloatingBlue*.

1.3 Organização do trabalho

Este documento está organizado em 5 capítulos, incluindo o presente capítulo de introdução, conforme a descrição a seguir:

1. O Capítulo 2 apresenta o levantamento do estado da arte relacionado a temática do trabalho. Primeiramente, serão descritos os principais trabalhos que aplicam RSSF baseadas em BLE e posteriormente o estado da arte dos trabalhos que implementam DTN. Além disso será comentado o panorama de estudos de RSSF que utilizam nós concentradores móveis. Por fim, serão discutidas as principais oportunidades que surgem quando analisados as três abordagem (BLE, DTN e nós concentradores) em uma arquitetura de RSSF, visando, sobretudo, a otimização do consumo energético;

2. O Capítulo 3 apresenta os fundamentos que cercam o desenvolvimento do trabalho. Os fundamentos teóricos de RSSF, BLE e DTN serão comentados, com enfoque nas principais tipologias e recursos que foram utilizados no trabalho. No capítulo também serão abordados os materiais que foram utilizados para implementação da solução, como bibliotecas e API (*Application Program Interface*) de software, plataformas de *hardware* e sensores;

3. O capítulo 4 comenta a solução e os métodos utilizados. Será apresentado a proposta de arquitetura de RSSF e a pilha de protocolos para comunicação. Além do detalhamento das rotinas de software desenvolvidas para integração dos elementos da arquitetura *FloatingBlue*.

4. O Capítulo 5 apresenta os métodos e resultados que validam a proposta deste trabalho. Inicia-se com um estudo de caso da arquitetura *FloatingBlue* implementada em um ambiente amplo e remoto. Em seguida, apresenta-se testes realizados no nó de terminação para avaliar seu consumo energético e eficácia na comunicação à distância. O capítulo conclui com uma análise comparativa desses resultados em relação a estudos similares na literatura científica.

5. O Capítulo 6 apresenta as conclusões deste trabalho e, com base nos resultados obtidos, propõe direções para pesquisas futuras.

Capítulo 2

Estado da Arte

2.1 Trabalhos relacionados

2.1.1 Aplicações de RSSF baseadas em Bluetooth Low Energy

As Redes de Sensores Sem Fio (RSSF) proporcionam o monitoramento de fenômenos a partir da distribuição espacial de sensores autônomos que se comunicam entre si. Há diversas tecnologias que podem ser utilizadas no elemento transceptor do sensor para estabelecer a comunicação em uma RSSF, como exemplo *Zigbee*, *Bluetooth*, *Wibree* (*Baby Bluetooth*), WiFi, *Wavenis*, *Ultra Wideband* (UWB) IEEE e *Z-Wave Dash7* [16]. Uma das tecnologias que vem ganhando notoriedade é o BLE devido a elevada taxa de dados e o baixo consumo, sendo também uma das principais tecnologias para a evolução da chamada Internet das Coisas (*Internet of Things* - IoT).

Diversas aplicações de RSSF são propostas na literatura científica baseadas em BLE. Uma das implementações mais comum é a utilização de dispositivos BLE para realização de localização *indoor* [17, 18]. Outros trabalhos visam analisar o desempenho e a funcionalidade de uma RSSF baseada em BLE em diversos estudos de casos. Os autores em [19] objetivam o monitoramento de segurança patrimonial por meio de sensores de presença e sensores acústicos. Já o autor em [20] aplica o BLE para monitoramento de patrimonio histórico, por meio de sensores de temperatura, umidade, intensidade e irradiância da luz. No trabalho [21] o autor sugere a utilização de sensores de comunicando BLE para monitorar vagões de trem.

Alguns trabalhos analisam e otimizam o tempo de sincronização entre os nós [22, 23]. Já outros visam analisar a vazão e taxa de dados de uma RSSF baseada em BLE. O trabalho [24] analisa esse aspecto na perspectivas das gerações 4.0, 4.2, 4.3 e 5.0 do *Bluetooth Low Energy*.

O consumo energético das redes é outro tema de estudo que tem grande notoriedade em aplicações com BLE aplicada a monitoramento a partir de RSSF. Isso pode ser observado no trabalho de [25] que visa otimizar o consumo de uma rede BLE através da metodologia de aquisição de dados. Outra solução que pode ser adotada para otimizar o consumo dos nós é por meio dos *Wake-up Radios (WuR)*. Essa tecnologia consiste em um sistema de rádio de baixa potência e secundário ao transceptor principal BLE. O WuR possui a função de gerar uma interrupção para transicionar o sistema do modo de baixo consumo para o modo de operação.

Alguns trabalhos abordam arquitetura de comunicação RSSF baseada em BLE em conjunto com o WuR [26–28]. O autor em [29] além de abordar WuR como elemento da solução de RSSF também sugere a utilização de sistema de colheita de energia, ou seja, quando o sensor, de forma autônoma, é capaz de capturar e armazenar energia de fontes externa (solar, térmica, cinética, entre outras). Nesse sentido, o trabalho [30] sugere utilizar a própria modulação *Gaussian Frequency Shift Keying* (GFSK) de uma rede BLE como fonte de energia para o sistema, a partir de um retificador de rádio frequência para corrente contínua.

Uma das estratégias também utilizadas para melhorar o desempenho energético da rede é utilização do modo *broadcasting*. Normalmente essa tipologia é utilizada como recurso de descoberta do nó vizinho, quando o dispositivo não está conectado e apenas enviando pacotes de anúncio (*advertising*) para ser descoberto. Nesse modo o sensor possui o consumo de energia menor e, assim, pode ser extraído melhores resultados de economia energética. Com isso, os autores em [31] visam encontrar configurações ótimas do modo *broadcasting* BLE para alcançar resultados satisfatórios em termos de consumo energético e latência. Semelhante forma, para atingir esses resultados, os autores em [32] visam criar um algoritmo para otimizar a janela de varredura de nós vizinhos *advertising* a partir dos resultados de varredura anteriores eliminando nós vizinhos redundantes.

O modo *broadcasting* BLE, além de permitir a descoberta de nós, possibilita

também utilizar esse recurso para implementar *beacons* BLE, uma alternativa de dispositivo BLE que permite a transferência de dados sem a necessidade de estabelecer conexão, porém possuindo restrições de segurança. Uma das principais aplicações de *beacons* consiste em sistemas de localização *indoor* [33]. Entretanto, também, pode ser utilizada para outros estudos de casos, como monitoramento da utilização de Equipamento de Proteção Individual (EPI) em cenários de construção civil [15] ou em *weareble* inteligentes que possibilitam o monitoramento dos cuidados da saúde e prevenção de doenças [34, 35]. A Tabela 1 sumariza os trabalhos relacionados de RSSF que utilizam BLE em diversos contextos.

Tabela 1: Aplicações de RSSF baseadas em BLE

Trabalho	**Ano**	**Aplicação**
[17, 18, 33]	2022, 2020, 2020	Geocalização indoor
[19, 24]	2022, 2018	Monitoramento e segurança patrimonial
[21]	2022	Monitoramento do transporte rodoviário
[22, 23]	2018, 2017	Otimização da sincronização dos nós
[20]	2019	Otimização de *throughput*
[25]	2019	Melhorar o consumo energético através do protocolo de aquisição de dados.
[26–28]	2021, 2017, 2020	Melhorar o consumo energético através de WuR
[30]	2021	Utilização de colheita de energia como forma de alimentação dos nós
[31, 32]	2019	Melhorar o consumo energético dos nós a partir da otimização da configuração dos parâmetros de *broadcasting* do BLE
[15]	2019	Monitoramento de segurança do trabalho a partir de EPI com *beacons*
[34, 35]	2020, 2018	*Beacons* para moniramento da saúde e prevenção de doenças

2.1.2 Aplicações de redes tolerantes a atrasos e desconexões

As redes DTN foram concebidas para aceitar condições desafiadoras como desconexões, elevadas latências, limitações de largura de banda e altas probabilidade de erro nas comunicações. A principal aplicação das redes DTN se dá em cenários quando não há uma infraestrutura de comunicação. Na arquitetura DTN, estabelecida em [36] é definido o protocolo de agregação (*Bundle Protocol*), responsável por implementar o mecanismo de encaminhamento de mensagens, conhecido

como *store-carry-forward*. Muitos trabalhos sugerem a utilização de DTN para aplicações de monitoramento e comunicação em cenários de catástrofes [11] [37]. Ainda no contexto de cenários remotos e inóspitos, no trabalho [38] o autor implementa uma RSSF baseada em DTN para monitoramento do continente antártico, enquanto os autores de [39] propuseram a utilização de DTNs para sensores subaquáticos que operam em condições de alta latência devido o cenário de comunicação aquática.

Outra aplicação de destaque é no monitoramento ambiental e na agricultura. O trabalho [13] relata uma solução de arquitetura de RSSF para monitoramento de parâmetros ambientais e de geolocalização sem a necessidade de uma infraestrutura de rede com *gateways*. Na área da agricultura também há aplicações de RSSF tolerantes a atrasos e desconexões. Os autores em [40] sugerem a utilização de nós DTN em máquinas agrícolas que coletam os dados dos sensores de monitoramento estáticos, espalhados no latifúndio, e posteriormente, quando os tratores retornam para a base, os dados coletados são descarregados em um servidor computacional.

Os primeiros modelos de DTN utilizavam o *Transmission Control Protocol/Internet Protocol* (TCP-IP) e Wi-Fi nas camadas inferiores [41] [36]. Entretanto os avanços das tecnologias de DTN também estão possibilitando a implementação de tecnologias de LPWAN nas camadas inferiores à camada de agregação. Os trabalhos [11, 12] utilizam rádio LoRa para possibilitarem a comunicação entre os nós DTN em longo alcance em cenário de catástrofe. De forma semelhante, os autores em [42] também utilizaram tecnologia LPWAN na implementação da DTN. Foi empregado o *Zigbee* como tecnologia LPWAN, onde o trabalho propõe que os *gateways Zigbee*, responsáveis por uma sub-rede, possibilitem as comunicações em situações de atrasos e desconexões. Outros trabalhos utilizam *hardware* com dois tipos de tecnologias de comunicação, curto e longo alcance, para otimizar o consumo energético da rede [43].

Outra possibilidade é a utilização de tecnologias de comunicação de curto alcance integradas ao protocolo de agregação do DTN, como o BLE e Wi-Fi (IEE 802.11). O trabalho [44] propõe utilizar pilha de protocolos de RSSF que agrega as camadas inferiores do BLE e as camadas superiores do *Bundle Protocol*. Já no trabalho [37] os autores propõe a utilização de um *gateway* multi-protocolo que assumem o papel de nós DTN e, são equipados com comunicação BLE e Wi-Fi. Há

outros trabalhos que investigam cenários simulados com BLE em redes DTN com enfoque em otimização de algoritmos de roteamento [45]. A Tabela 2 sintetiza os trabalhos relacionados à aplicações de DTN.

Tabela 2: Aplicações de DTN

Trabalhos	**Ano**	**Aplicação**
[11, 12, 37, 42]	2020,2020, 2020, 2011	Comunicação em cenário de catástrofe
[38, 39]	2022, 2022	Monitoramento de ambientes inóspitos
[44]	2022	Proposta de pilha de protocolos contendo protocolo de agregação e BLE nas camadas inferiores
[13]	2018	Moninoramento de condições ambientais
[40]	2011	Moninoramento agrícola
[43]	2007	Otimização dos recursos energéticos da rede
[45]	2013	Otimização do algoritmo de roteamento de uma DTN a partir de um rádio secundário BLE

2.1.3 Arquiteturas de RSSF com nós concentradores móveis

Os nós concentradores em algumas arquiteturas de RSSF também arroga-se da função de *gateway*, eles são os elementos que vão coletar a informação de uma sub-rede (ou *cluster*) com diversos nós de terminação. Os nós concentradores normalmente são fixos, porém há autores sugerem RSSF na qual esse elemento é móvel, trazendo algumas oportunidades e desafios.

Alguns trabalhos utilizam nós concentradores móveis por meio de Veículo Aéreo Não Tripulado (VANT) para coletar dados dos sensores. O trabalho [46] utiliza um *gateway Zigbee* acoplado a um VANT para coletar dados de sensores de monitoramento do ambiente: temperatura, umidade e pressão atmosférica. Ainda no sentido de utilizar LPWAN em conjunto com VANT, o trabalho [8] sugere a utilização de *gateway* da tecnologia LoRa acoplados em VANT para coletar dados de sensores em um ambiente florestal. Outros trabalhos sugerem a utilização de nós concentradores móveis que implementam tecnologias de comunicação sem-fio de menor alcance, como o padrão IEEE 802.11, atrelado a um sistema de WuR para despertar os sensores e otimizar o consumo energético dos mesmos, como pode ser observado nos trabalhos [47, 48].

Outra oportunidade é a implementação de DTN na comunicação dos nós concentradores. O trabalho [9] sugere a utilização de uma rede de comunicação

para cenário de desastre em que os nós concentradores integrados a VANTs capturam mensagens de smartphones, com auxilio de um *tranceiver* LoRa acoplado ao dispositivo celular. Os nós concentradores se comunicam por meio de uma implementação de DTN, até um dos nós encontrarem uma estação para descarregar as mensagens. De modo semelhante, ao utilizar nós concentradores móveis DTN, o trabalho [10] propõe um protocolo de roteamento DTN para uma rede baseada em VANTs, como nó concentrador, aplicada a cenários de busca e resgate. O protocolo utiliza as informações geográficas dos VANTs para tomar decisões apropriadas de encaminhamento de mensagens.

O BLE é outra tecnologia que pode ser empregada em redes que possuem na arquitetura nós concentradores móveis. Os autores do trabalho [9] propõe o emprego de VANT para coletar informações de *Wireless Ground Sensors (WGS)* no cenário de agricultura 3.0. No trabalho citado foram feitos simulações da arquitetura da rede proposta utilizando algumas tecnologias de comunicação sem fio, dentre delas o BLE. Na proposta de utilizar ambiente simulado para analisar a performance de uma rede BLE com nós concentradores móveis, o trabalho [49] propôs uma otimização dos indicadores (densidade da rede, roteamento, cobertura e vida útil da bateria) da RSSF, levando em consideração diversos fatores, inclusive a velocidade e atitude do nó concentrador móvel. Em uma abordagem mais prática o trabalho [50] implementa a proposta de uma RSSF baseada em VANT assumindo a função de nó concentrador coletando informações de sensores por meio do BLE.

Tabela 3: RSSF com nós concentradores móveis

Trabalho	Ano	Comunicação	DTN	Aplicação	Consumo
[51]	2011	DIVERSAS	SIM	Cidades inteligentes e agricutura	633 nJ
[12]	2020	LORA	SIM	Comunicação em cenário de desastre	72 uW
[11]	2020	LORA	SIM	Comunicação em situação de emergência	231 mW
[9]	2019	LORA	SIM	Comunicação em cenário de desastre	NA
[46]	2019	ZIGBEE	NÃO	Monitoramento ambiental	50uW
[43]	2007	ZIGBEE	SIM	Monitoramento de frota de ônibus	80 mW
[52]	2018	ZIGBEE	SIM	NA	181 mW
[8]	2018	LORA	NÃO	Comunicação em cenário de desastre	NA
[53]	2019	LORA	NÃO	Monitoramento ambiental	NA
[47]	2022	IEEE 802.11	NÃO	Monitoramento ambiental	9 mJ
[48]	2020	IEEE 802.11	NÃO	Agricultura	12 uW
[49]	2016	BLE	NÃO	Monitoramento ambiental	57 mW
[50]	2019	BLE	NÃO	Arquitetura de comunicação de escopo aberto	NA

2.2 Discussão dos trabalhos

Os trabalhos da literatura citados evidenciam a importância e abrangência da pesquisa da área de RSSF. Primeiramente, nota-se que o BLE é uma tecnologia de comunicação fundamental para as RSSFs, sobretudo, devido ao consumo de energia reduzido dos nós [20,26–28,32] . Diversas aplicações podem ser utilizadas para RSSF baseada em BLE, tais como localização *indoor*, agricultura, monitoramento ambiental e segurança patrimonial. Ainda para minimizar o consumo dos nós é possível utilizar o modo *broadcasting* do BLE [15, 34, 35]. Entretanto, devido ao alcance da comunicação reduzido, uma RSSF baseada em BLE é considerada não recomendável para monitoramento de ambientes amplos, sendo preferível as tecnologias LPWAN.

As tecnologias LPWAN normalmente são infraestruturadas com *gateways* no centro de redes com tipologia estrela [2, 7]. Porém, alguns trabalhos sugerem a utilização de nós concentradores móveis, visando aumentar a área de cobertura da rede, sem a utilização de infraestrutura de rede adicional com nós concentradores móveis [46–51, 53]. Isso implica em desafios na sincronização do *duty-cycle* dos nós sensores e nó concentrador. Alguns métodos podem ser utilizados para resolver o problema, como nós com *duty-cycle* longos, que consomem mais energia ou utilização de tecnologias de interrupção assíncrona por meio de WuR [26–28]. A desvantagem

do WuR é o fato da tecnologia adicionar mais complexidade ao protocolo de comunicação, por adicionar um transceptor ao *hardware* do nó de terminação.

No contexto de nós concentradores móveis, o BLE retorna como alternativa em relação as tecnologias de LPWAN, visto que o nó concentrador móvel pode realizar trajetórias que minimizem a distância de comunicação dos nós. Além disso, é possível aos nós sem fio transmitir dados de leitura por meio dos pacotes de *advertising* BLE, com um consumo menor ou similar ao modo *sleep*, quando comparado um nó LPWAN [31, 32].

Pensando em uma abordagem de arquitetura de comunicação RSSF com vários nós concentradores móveis, é importante que esses possuam mecanismo de *store-forward*, visto que em aplicações para lugares amplos e restritos, a comunicação é suscetível a desconexões devido a falta de conectividade fim-a-fim entre nós sem fio da rede e a infraestrutura de comunicação, não sendo possível descarregar os dados em tempo real no servidor de aplicação. Um dos protocolos que podem ser utilizados nesse contexto, é a implementação do *bundle protocol* para DTNs, normalmente utilizado em cenários sem infraestrutura de rede, zonas remotas e inóspitas [11, 12, 37, 42]. A tecnologia permite que os nós sem fio transfiram dados, de forma confiável, sob condições de desconexões da rede e elevadas latências.

Nesse sentido, alguns trabalhos da literatura propõem a combinação de nós DTN como nó concentrador móvel [9, 11, 12, 43, 51, 52]. Normalmente o objetivo da arquitetura de rede DTN é a comunicação fim-a-fim em ambientes sem infraestrutura (ex. cenários de desastres) não sendo o consumo de energia otimizado na maior parte dos trabalho [11]. Contudo, não considerar o consumo energético dos nós sem fio em redes DTN, pode levar ao desenvolvimento de nós DTN que utilizam dispositivos não otimizados. Isso pode implicar em *hardware* com maior poder de processamento e com maior consumo de energia.

No presente trabalho utilizou-se *beacons* BLE como nós de terminação juntamente com nós concentradores móveis que implementam DTN, sendo que isto mostrou-se como uma alternativa com foco na redução do consumo energético dos nós. Alguns trabalhos da literatura apresentam arquiteturas que utilizam RSSF baseada em *beacons* BLE para comunicação com nós concentradores móveis DTN. Contudo, há poucos trabalhos que empregam uma abordagem que une todos esses

elementos. O trabalho [51] apresenta modelos simulados que empregam a arquitetura BLE/DTN/*gateway* móvel, porém, conformem comentado o trabalho de literatura não realiza experimentações práticas que contempla a proposta realizada no presente trabalho.

Capítulo 3

Fundamentos e Materiais

3.1 Arquitetura e modelo de referência de protocolo de uma RSSF

Uma Rede de Sensores Sem Fio (RSSF) é definida como um conjunto de sensores repassadores de dados que operam de forma colaborativa, visando o monitoramento/controle de um ou mais parâmetros físicos dentro de uma área limitada e sem a utilização de cabeamento na camada física. Os dados obtidos pela RSSF, alimentam um sistema supervisório de monitoramento e controle. A principal vantagem de uma RSSF é que ela estende a área de cobertura de um sensor individual para uma área maior. Normalmente os sensores são alimentados por pequenas baterias, por serem dispositivo de baixa potência [54]. A Figura 1 apresenta arquitetura de referência de uma RSSF.

Uma arquitetura típica de uma RSSF possui os seguintes elementos:

1. campo de atuação dos sensores;
2. nós sensores sem fio de aquisição e retransmissão de dados;
3. nós de controle ou concentrador de dados e *gateway*;
4. enlace de Rádio Frequência (RF) entre nós;
5. gerenciamento de aplicação para monitoramento e controle dos dados.

Os nós de sensoriamento normalmente são compostos por um microcontrolador, o elemento sensor, um tranceptor e bateria. Na rede normalmente está presente

um nó concentrador que recebe os dados dos sensores e conecta ao gerenciador de aplicação para armazenamento, visualização e manipulação dos dados a nível de usuário.

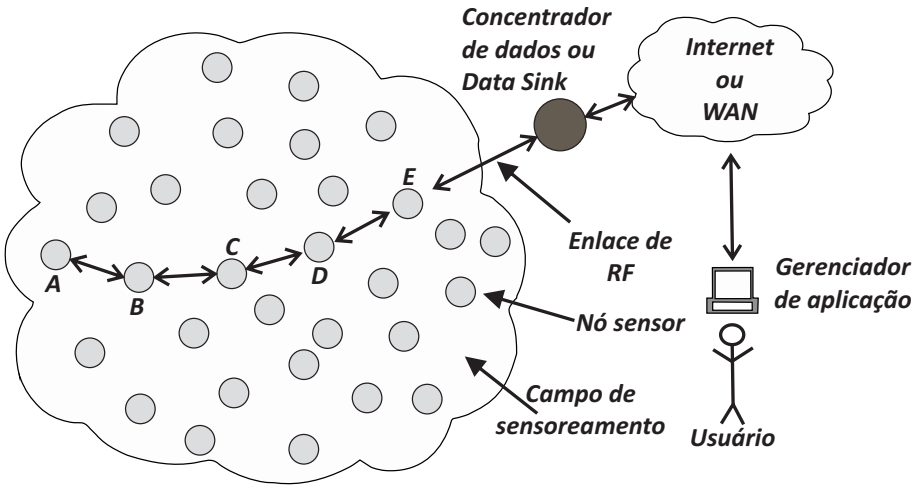

Figura 1: Elementos referência de uma arquitetura de RSSF [54].

3.1.1 Critérios de comunicação

Antes de adotar uma tecnologia de comunicação RSSF é importante avaliar quais critérios serão prioridades na arquitetura e tecnologia de rede, como: alcance de comunicação, consumo de potência, topologia, largura de banda, vazão e latência [55].

1. **Alcance de comunicação:** distância que o sinal pode propagar, ou seja, a área para operação da tecnologia sem fio. O alcance é classificado em curto, médio e longo. As tecnologias de curto alcance são de dezenas de metros no máximo, como o Bluetooth. O médio alcance suporta geralmente até 1 milha entre dois dispositivo é o caso do Wi-Fi. Longo alcance é considerado a comunicação entre dois dispositivos a uma distância superior a 1 milha, como as tecnologias celulares (2G, 3G, 4G e 5G) e LPWAN.

2. **Consumo de potência:** os nós de uma RSSF podem ser alimentados por uma fonte de potência conectada a rede elétrica de distribuição de energia, nessa modalidade a comunicação normalmente não é limitada por esse critério, entretanto possui menos mobilidade. Enquanto o nó pode ser alimentado por

uma bateria que traz mais flexibilidade para a RSSF, porém isso restringe a conectividade e a capacidade de processamento do *hardware*.

3. **Topologia de rede:** As principais topologia de rede são estrela, malha e ponto-a-ponto. Destaca-se a rede malha por proporcionar um alcance maior, através de nós intermediários que retransmitem o tráfego para outros nós. Nas redes em malha utilizam-se algoritmos de roteamento para minimizar o consumo dos nós e otimizar o tráfego dos dados.

4. **Largura de banda e vazão:** são relacionados com a capacidade de transferência de dados em um intervalo de tempo, sendo a largura de banda relacionada com a máxima capacidade de transmissão do canal, enquanto vazão é o que foi de fato transferido com sucesso, visto que as mensagens estão sujeitas a perdas, retransmissões, colisões e ruídos. Em uma RSSF esses critérios auxiliam a dimensionar os elementos da rede, conforme a aplicação desejada.

5. **Latência:** é o tempo médio gasto entre o envio do pacote e o recebimento bem sucedido no destino. Esse critério impacta diretamente na escolha do protocolo de rede adotado, sobretudo em aplicações de tempo real.

3.2 Redes Tolerantes a Atrasos e Desconexões

As Redes Tolerantes a Atrasos e Desconexões (DTN) é a área que aborda os desafios de uma rede sujeita a desconexões e interrupções, sem conexão ponta a ponta entre origem e destino final dos dados. A DTN é designada para operar eficientemente em distâncias extremas e latências longas, podendo ser mensuradas em dias. O protocolo de internet não é eficaz em tolerar longos atrasos, eventuais interrupções de comunicação e longas distâncias, devido as características de sua arquitetura, como: definição de uma rota ponta a ponta entre a origem e o destinatário; retransmissão baseada em um *feedback* estável do destinatário; todas estações e rotas suportam protocolo TCP/IP.

Enquanto a arquitetura DTN pode ser resumida nas seguintes características: abstração da mensagem de tamanho variável para facilitar a capacidade da rede para agendamento ou decisões da seleção da rota; armazenamento de dados dos nós sem

fio para suportar operações de *store-and-forward* em várias rotas e mecanismo de segurança para proteger a infraestrutura de usuários não autorizados [56].

3.2.1 Arquitetura da DTN

A arquitetura de uma rede DTN é definida na *Request for comments* 4838 (*RCF 4838*) [57]. O documento comenta a organização dos nós para realizar os procedimentos de *store-and-forward* em um cenário de altas latências e desconexões.

Quanto as pilhas de protocolo DTN, há uma camada particular, entre a camada de transporte e aplicação, nomeada de camada de agregação (*bundle layer*) que possui a função de garantir a interoperabilidade entre diferentes redes. Além disso, a camada de agregação emprega armazenamento persistente para combater interrupção da rede, transferência segura e confiável de pacotes por salto a salto e diagnóstico e gerenciamento de funcionalidades. As camadas abaixo da camada de agregação são definidas conforme o cenário da aplicação. Ou seja, conforme a Figura 2, cada nó DTN pode ser equipados com diferentes pilhas de protocolos, conforme sua região (ex. Região A, região B).

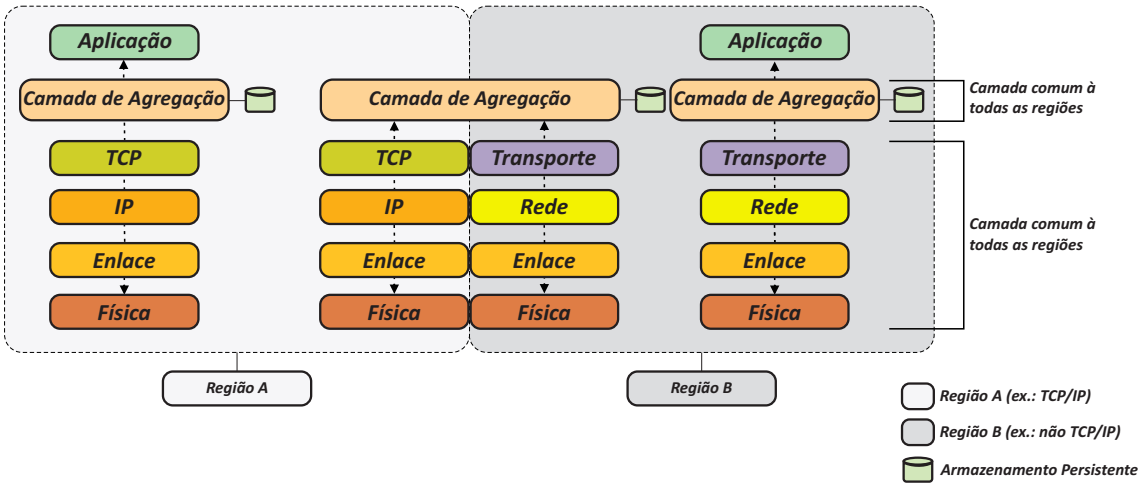

Figura 2: Camadas de uma rede tolerante a atrasos e desconexões.

As mensagens enviadas em uma DTN possuem tamanho arbitrário denominadas (*Application Data Units, ADUs*). Os ADU são fragmentados pela camada de aplicação em um ou mais unidades, chamadas *bundles* que são encaminhados pelos nós DTN. Os *bundles* tem um formato contendo dois ou mais blocos, cada bloco contem dados da aplicação ou informação de controle para encaminhar a mensagem

ao destinatário. A vantagem de cada *bundle* conter informações como *timestamp* da origem da mensagem, classe de prioridade e o tamanho do pacote é que auxilia na decisão do algoritmo de roteamento.

O nó origem e o destinatário são identificados com o *Endpoint Identifiers (EIDs)*. Um EID pode referenciar um ou mais nós DTN. Uma rede DTN não espera que a conexão de rede esteja sempre disponível ou confiável. Dessa forma, os nós podem armazenar os *bundles* enfileirados, por um determinado tempo, até retomar a disponibilidade da rede, esse processo se chama de armazenamento persistente. Diferente de uma rede internet convencional, em que nem todos os nós são alcançáveis. Dessa forma, na arquitetura DTN há o conceito de contato, correspondente a oportunidade favorável na qual ocorre o encaminhamento dos *bundles*. A otimização de um contato requer da camada de adaptação, a fragmentação dos *bundles*, conforme a limitação da conexão, e posteriormente roteamento dos fragmentos.

Outro fator das DTNs é que comunicação é suscetível às perdas de pacotes, devido a breve janela de comunicação dos contatos. Para garantir a entrega confiável de uma mensagem em uma rede DTN há duas possibilidades, o reconhecimento por (*acknowledgment*) fim-a-fim da comunicação por meio da camada de aplicação ou transferência de custódia. Na transferência de custódia o protocolo de agregação (BP, do inglês *Bundle Protocol*) e o protocolo de transporte se encarregam de realizar a persistência da retransmissão dos pacotes perdidos ou corrompidos, por meio de uma delegação da responsabilidade de entrega de um nó para outro. O nó custodiante deve armazenar o *bundle* até outro nó da rota, não necessariamente precisa ser o próximo o nó, aceite a sua custódia ou até atingir o tempo de vida do pacote. Um nó DTN não é obrigado a receber a transferência de custódia, por diversos limitações, como: bateria do nó, espaço de armazenamento do *buffer*, políticas de segurança e tempo de vida da mensagem [58]. A Figura 3 ilustra a transferência de um *bundles* entre os nós DTN, até a entrega aos nós destino. Como pode ser observado, nem todos os nós são responsável pela custódia do *bundle*. Na Figura 3, o nó 2 não é um nó custodiante, porém realiza o armazenamento persistente que salva os *bundles* em fila até o retorno da disponibilidade da conexão da rede (procedimento de *store-and-forward*).

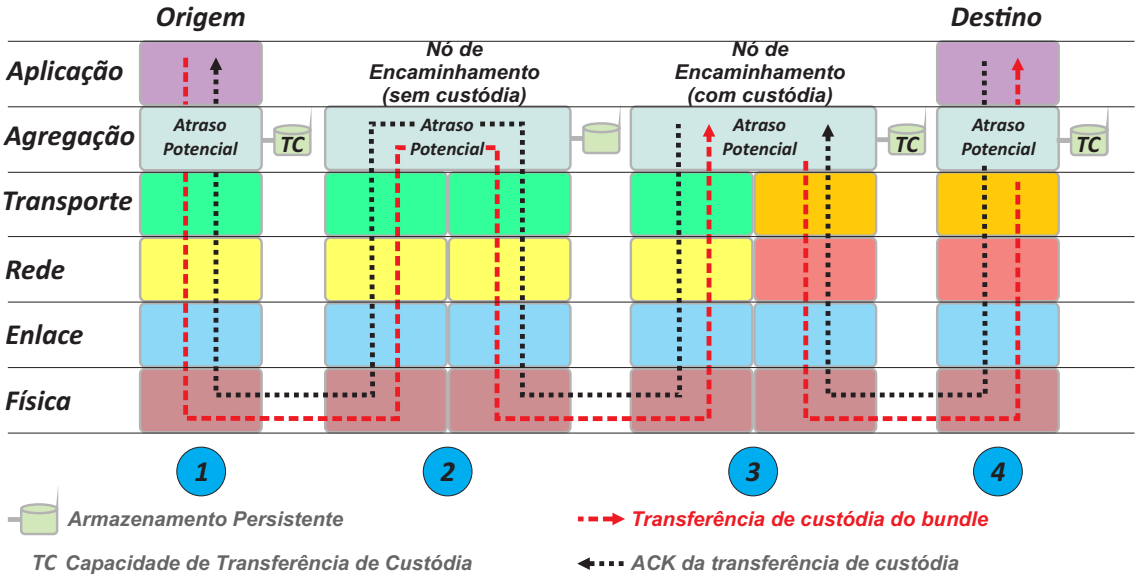

Figura 3: Transferência de *bundles* em uma DTN [58]

3.2.2 *Bundle Protocol 7*

Em 2007 foi definido pela RCF 5050 [59] a especificação do protocolo de agregação de uma DTN. No documento é descrito o formato dos *bundles* passados entre os elementos participantes do protocolo BP. A partir do documento o protocolo foi implementado em diversas linguagens de programação e plataformas de computação. A especificação mais atual do protocolo é a versão 7 (RCF 9171). Para compreensão do protocolo é necessário compreender o formato dos blocos que compõe o *bundle* e os componentes do nó DTN [60].

3.2.2.1 Blocos

Um *bundle* contem um bloco primário e blocos canônicos. O bloco primário contem informações do *bundle* com os campos: versão, controle de *flag* para prover informação do pacote como fragmentação e relatórios; *Cyclic Redundancy Check* (CRC) opcional; EID do nó destino; EID do nó origem; EID do nó "relatório para"(*Report-to*), esse nó tem função de salvar os registros de um *bundle*; *timestamp* de criação; tempo de vida do pacote e, tamanho total do pacote e do fragmento.

Os blocos canônicos são opcionais, e podem conter informações da idade do *bundle*, contador de saltos e nó anterior. Nos blocos canônicos há também os blocos de e o bloco de *payload*. Os blocos canônicos são identificados por um código (*Type Code*), possuem *flags* de controle, tem campos de dados e um campo número

identificador do bloco no contexto de um *bundle* [60]. A Figura 4 ilustra os blocos de um *bundle* segundo *Bundle Protocol 7*.

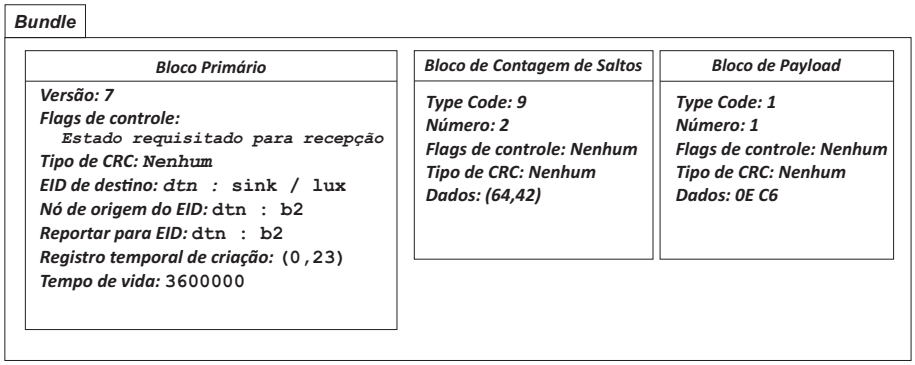

Figura 4: Bloco de um *bundle* [61].

3.2.2.2 Componentes do nó DTN

1. **Agente do protocolo *bundle*:** O Agente do protocolo *bundle* (BPA, do inglês *Bundle Protocol Agent*) executa serviços de gerenciamento da comunicação entre os Agentes de Aplicação (AA) e o Adaptador da Camada de Convergência (CLA, do inglês *Convergence Layer Adapter*). O BPA também estrutura o *bundle* para o AA.

2. **Agente de aplicação:** O componente AA define a interface entre o BPA e uma aplicação. Um genérico AA deve receber *bundles* de entrada e compor pacotes de saída (e vice-versa).

3. **Camada de convergência:** A camada de convergência (CL, do inglês *Convergence Layer*) estabelece o contato com o nó DTN e gerencia as característica da comunicação para que ocorra a entrega do *bundle*. O CLA é uma implementação do protocolo da camada de convergência. Há dois protocolos definidos na CL, um bidirecional TCP (TCPCL) e o mínimo unidirecional TCP (MTCP). Na camada de transporte atrelada à CL, há abordagens baseadas em tecnologias como Bluetooth e comunicação serial.

Os elementos podem ser visualizados na Figura 5.

3.2.3 DTN 7

DTN 7 é uma implementação *open source* de uma DTN baseada no *Bundle Protocol 7*, disponível na linguagem de programação GoLand, Rust e Kotlin. O DTN7 pode ser executado em diversas arquiteturas de *hardware* (x86, ARM e MIPS), baseado nos sistemas operacionais populares: Windows, MacOS e Linux. A arquitetura da DTN7 é dividida nos componentes de Armazenamento (*Store*), Descoberta de pares (*Discovery*), Roteamento (*Routing*), CLA, BPA e AA. No DTN 7 é utilizado o protocolo MTCP e TCPCL na camada de transporte para troca de mensagens entre nós DTN [61]. A Figura 5 demonstra a interação entre os componentes.

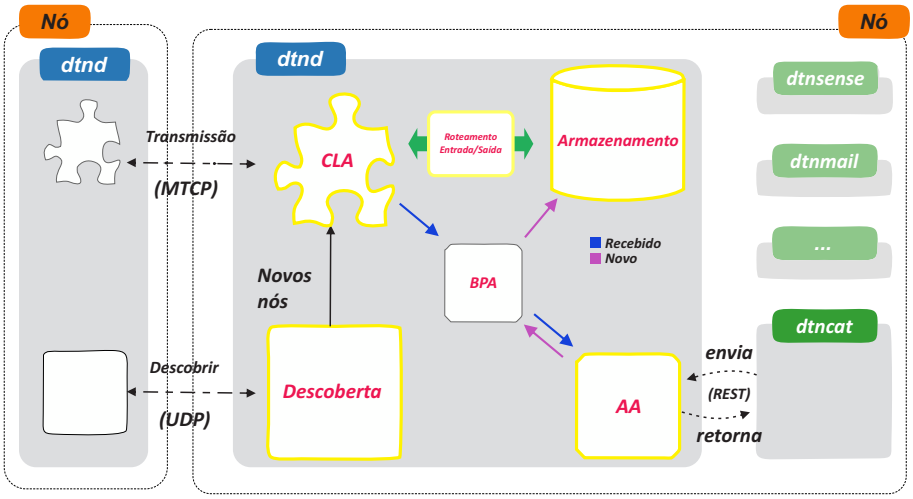

Figura 5: Fluxo de dados em DTN 7.

A comunicação entre nós DTN é feita por meio do componente de descoberta e o CLA. Múltiplos clientes podem se conectar ao Agente de Aplicação (AA). Para armazenar os *bundles* localmente, os dados são serializados e salvos no sistema de arquivo. Na DTN 7 é utilizado o módulo armazenamento para gerenciar um sistema de índices dos *bundle* com um determinado arquivo de sistema, com objetivo de realizar uma busca rápida por arquivo.

No DTN 7 o AA se comunica com as aplicações utilizando *Application Program Interface* (API) RESTful Web ou Websocket que não interage em formato de *bundles* apenas por meio do EID destino e o *payload*. Quanto a Camada de Convergência o CLA é o módulo que deve suportar os outros módulos para processamento e direcionamento do *bundle* de entrada e saída. O componente de Descoberta é

utilizado para anunciar a existência do nó e ouvir os vizinhos. Esse mecanismo de descoberta é baseado no envio contínuo do CLA para os nós vizinhos e notificação dos CLA recebidos. Os serviço que gerenciam as conexão entre os módulos da arquitetura é feita pelo BPA.

Cada *bundle* que não está endereçado para um nó particular é enviado para um ou mais CLA de nós vizinhos. A decisão da seleção do CLA é feita com base no algoritmo de roteamento. O DTN 7 suporta diversos algoritmos de roteamento: *epidemic*, *prophnet*, *spray and wait*, *binary spary*, entre outros. Por padrão é utilizado o algorítimo *epidemic*. O módulo CLA é informado, na recepção de um *bundle*, carrega informações sobre o remetente e o destinatário, além o algoritmo compilar um subconjunto de conexões conhecidas que ainda não receberam este pacote.

3.2.4 Algoritmo de roteamento epidêmico

O roteamento epidêmico é baseado em inundação. Os nós continuamente replicam e transmitem mensagens para contatos recém-descobertos que ainda não possuem cópia das mensagens. Além da inundação, podem ser utilizadas outras técnicas para limitar o número de transferência das mensagens. O roteamento epidêmico visa que os bancos de dados dos nós DTN estejam sempre sincronizados. As mensagens são descartadas somente quando o tempo de vida do *bundle* expira. Não é recomendado a utilização do roteamento epidêmico quando existe limitação dos recursos de *hardware* (armazenamento, energia e largura de banda). Normalmente utilizado em ambientes remotos com poucos nós [45, 62].

3.2.5 Biblioteca de software dtn7-go

O dtn7-go é a implementação do protocolo DTN 7 na linguagem GoLand [63]. A biblioteca possui dois elementos principais, o primeiro é o *dtnd*, um daemon de rede tolerante a atraso. Daemons são serviços que rodam em *background* em sistemas operacionais e são executados em conjunto com uma aplicação principal de forma paralela. O *dtnd* implementa os serviços de um nó dtn na rede que pode transmitir, receber e encaminhar *bundles*. A configuração do daemon é feita a partir de um arquivo de configuração no formato *Tom's Obvious Minimal Language* (TOML). Há duas formas para permitir a comunicação de uma aplicação externa com o dtnd:

API REST e uma API WebSocket. A API REST permite que um cliente registre um endereço, receba pacotes, crie e envie *bundles* por meio de POST de objeto JSON para o servidor RESTful HTTP do dtnd. Caso necessário uma comunicação bidirecional utilizando a API Websocket pode ser utilizada.

O segundo elemento principal da biblioteca dtn7-go é o *dtn-tool*, uma aplicação que utiliza WebSocket API para se comunicar com o *dtnd*. Como o nome sugere é uma aplicação de alto nível que implementa ferramentas para manipular os *bundles*, como criar novos *bundles* (gravados em um arquivo ou na saída padrão *stdout*), utilizando a função *create*. Além disso, permite imprimir um *bundle* em um JSON legível para o usuário, por meio da função *show*. Já para a troca de *bundles* entre os nós DTN, utiliza-se a função *exchange*, que permite monitorar um diretório, utilizando um módulo *watcher* da linguagem Go e, enviar todo novo *bundle* a uma instância dtnd, além de estabelecer um agente *websocket* do nó destino. Da mesma forma, os *bundles* recebidos são armazenados nesse diretório.

3.3 Módulo Python *subprocess*

O *subprocess* é um módulo Python incorporado que pode ser utilizado para criar novos processos e interagir com os fluxos de dados de entrada e saída. Dessa forma é possível executar comandos *shell* e executar binários atráves do *File System* do Linux. Diferentemente do módulo Python *Operating System (OS)*, a biblioteca *subprocess* permite interagir com as saídas dos comandos *shell*. A biblioteca *subprocess* pode ser utilizado em conjunto com o DTN 7 para executar os comandos *shell* do *dtn-tool* [64].

3.4 Bluetooth Low Energy

O *Bluetooth Low Energy* (BLE) iniciou a partir da tecnologia Wibree que tinha como objetivo ser a sucessora do Bluetooth. O diferencial da tecnologia Wibree está relacionado ao consumo menor que a tecnologia Bluetooth clássica. O BLE foi integrado na *Bluetooth Core Specification V4.0* por meio de uma nova pilha de protocolo, correspondente a tecnologia Wibree.

Pelo fato de possuir baixo consumo, praticidade e flexibilidade no desenvolvimento de aplicativos móveis, a tecnologia possibilitou ser utilizada em diversos dispositivos, sobretudo portáteis, como smartphone, relógios e fones de ouvido. O BLE é otimizado para transmitir blocos de 37 octetos no máximo, transmitidos a 1 Mbit/s em ciclos de transmissão. Na média a transmissão do BLE é de algumas centenas de bits por segundo e tempo de transmissão menor que 6 ms. Já o alcance de operação depende de vários fatores (ambiente de operação, antena, invólucro e orientação do dispositivo, entre outros) porém é focado em aplicações de curto alcance, com operação típica em um raio de até 5 metros [54].

3.4.1 Topologia de rede: *Broadcasting* e os *Beacons*

O BLE pode se comunicar com demais dispositivos de duas maneiras: *broadcasting* ou conexão. No tipo conexão os dados podem ser transmitidos em ambas direções, com um *payload* maior em relação ao tipo *broadcasting*. A conexão possibilita uma troca de dados permanente e periódica entre dois dispositivos. Já o modo sem conexão (*broadcasting*) é possível enviar os dados para qualquer dispositivo em estado *scanning* ou receptor. Entretanto, nesse mecanismo é permitido apenas transmissão de dados unidirecionais. O modo *broadcasting* possui duas funções definidas, *broadcaster* e observador. Na função *broadcaster* o dispositivo envia pacotes de anúncio (*advertising*) periodicamente para qualquer receptor bluetootth, enquanto a função observador escaneia repetidamente com uma frequência predefinida para receber quaisquer pacotes de *advertising*.

O modo *broadcasting* é a forma fácil e rápida de implementação BLE, normalmente adotado quando se deseja transmitir dados pequenos em um intervalo de tempo fixo ou quando deseja comunicar-se com múltiplos dispositivos simultaneamente. A principal aplicação dessa topologia consiste em envio de uma mensagem *broadcast* e descoberta de nós escravos para estabelecer conexão. A desvantagem desse modo consiste, sobretudo, na segurança e privacidade de dados [54].

Os *beacons* são implementações BLE na topologia *broadcasting*, possuem a característica de serem pequenos e com baixo consumo energético. A principal aplicação é a localização *indoor* onde o GPS não performa bem comparado a ambientes externos. A partir da intensidade do sinal bluetooth é possível identificar a locali-

zação dos dispositivos auxiliado pela interação com outros equipamentos equipados com a tecnologia bluetooth, tais como smartphones. Além disso há os *Nearables* que são *beacons* que interagem com outros dispositivos criando um ambiente inteligente. Eles permitem que uma série de regras sejam executadas com base em vários cenários para efetuar o comportamento desejado em dispositivos locais equipados com o BLE [65].

3.4.2 Introdução ao Protocolo BLE

A arquitetura de protocolos do BLE conforme a Figura 6. É formada por 4 blocos funcionais: o controlador hospedeiro, que está relacionado a camada física e de enlace; a interface *Host Controller Interface (HCI)* entre o controlador e hospedeiro; os perfis de adaptação do hospedeiro e a aplicação. A separação dos blocos se deve pelo fato que o módulo controlador possui pilhas com processamento com requisitos de tempo real. Assim, é importante que esse bloco seja separado do hospedeiro que possui menos restrições de tempo e implementa pilhas de protocolos mais complexas. As camadas do bloco controlador serão explicadas com maiores detalhes e as camadas do hospedeiro serão comentadas de forma conceitual nos tópicos abaixo.

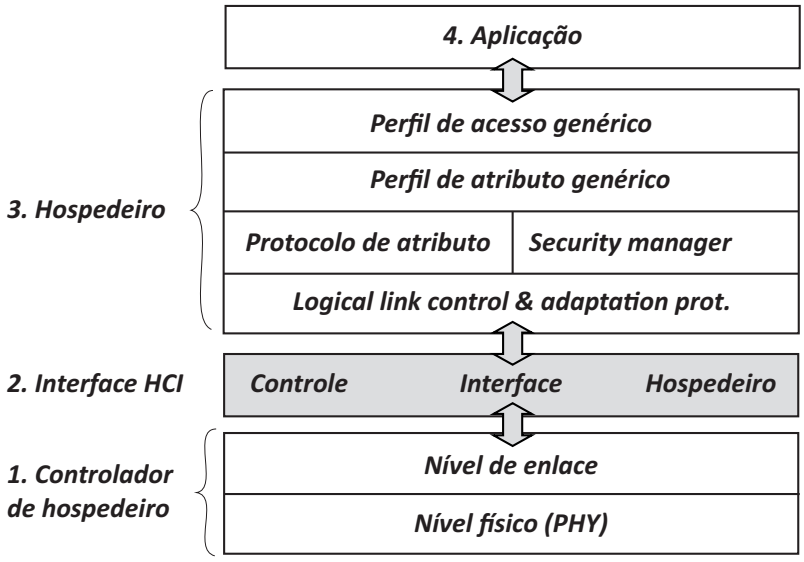

Figura 6: Arquitetura de protocolos BLE [54].

3.4.2.1 Camada física

O rádio utiliza a banda de 2.4 GHz ISM(do inglês, *Industrial, Scientific, and Medical*) para se comunicar, divididos em 40 canais com largura de banda de 2 MHz. Desses 40 canais, 3 são destinados a canais de *advertising* e 9 são destinados para comunicação de dados, os demais canais são compartilhados e estão sujeitos a interferência na presença de uma rede Wi-Fi 2.4 GHz. Devido o compartilhamento do BLE com Wi-Fi no mesmo espectro, adota-se o algoritmo *Frequency Hopping Adaptative (FHA)*, que define um padrão de saltos aleatórios de forma que não interfira em canais que possuem uma porção ocupada pela rede Wi-Fi, utilizando a seguinte fórmula [54].

$$canal = (atual + salto) mod 37 \tag{1}$$

A modulação adotada para codificar do sinal transmitido é a modulação em frequência *Gaussian Frequency Shift Keying (GFSK)*, utilizada no Bluetooth clássico. Essa modulação é uma variação da modulação *Frequency Shift Keying* (FSK) que utiliza filtro gaussiano para formar os pulsos antes deles serem modulados, visando reduzir a largura espectral e o espectro fora de banda [66]. Os canais de frequência para o BLE podem ser vistos com detalhes na Figura 7.

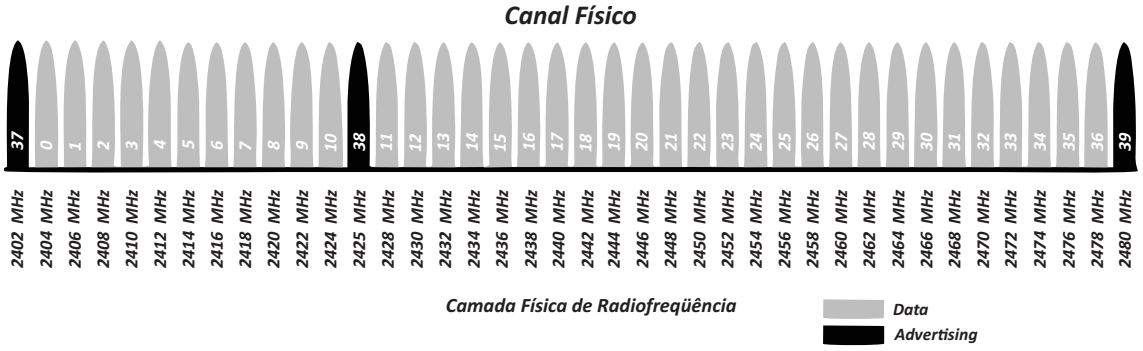

Figura 7: Canais na banda 2,4 GHz para o BLE [66].

3.4.2.2 Camada de enlace lógico

O nível de enlace lógico está diretamente ligado com a camada física e é uma combinação customizada de *hardware* e *software*. Além disso, é a única camada especifica de tempo real de toda pilha de protocolo BLE, já que ela é responsável

pelos requisitos de tempo definido pela especificação do Bluetooth 4.1. Entre as funcionalidades da camada de enlace relacionada a camada física estão a determinação do preâmbulo, endereço de acesso e enquadramento do protocolo da camada física; geração e verificação do CRC; geração de números randômicos e criptográfica com padrão avançado (*Advanced Encryption Standard - AES*) [66].

Os elementos de software implementados na camada de enlace são responsáveis por gerenciar as conexões de estado do rádio. Um dispositivo BLE pode ser um mestre, um escravo ou ambos, o dispositivo que inicia uma conexão é o mestre e os dispositivos que anunciam disponibilidade e aceitam a conexão serão escravos. Essa relação pode ser visualizada na Figura 8.

A camada de enlace define regras de nós em estado de *advertiser* e *scanners*. Além de definir os endereço dos dispositivos. O *Public device address* (PDA) é o endereço programado de fábrica, deve ser registrado com *IEEE Registration Authority* e não deve ser mudado durante o tempo de vida do dispositivo. Já o *Random Device Address* (RDA) pode ser pré-programado no dispositivo ou gerado dinamicamente em tempo de execução e possui aplicações na privacidade e segurança do protocolo [66].

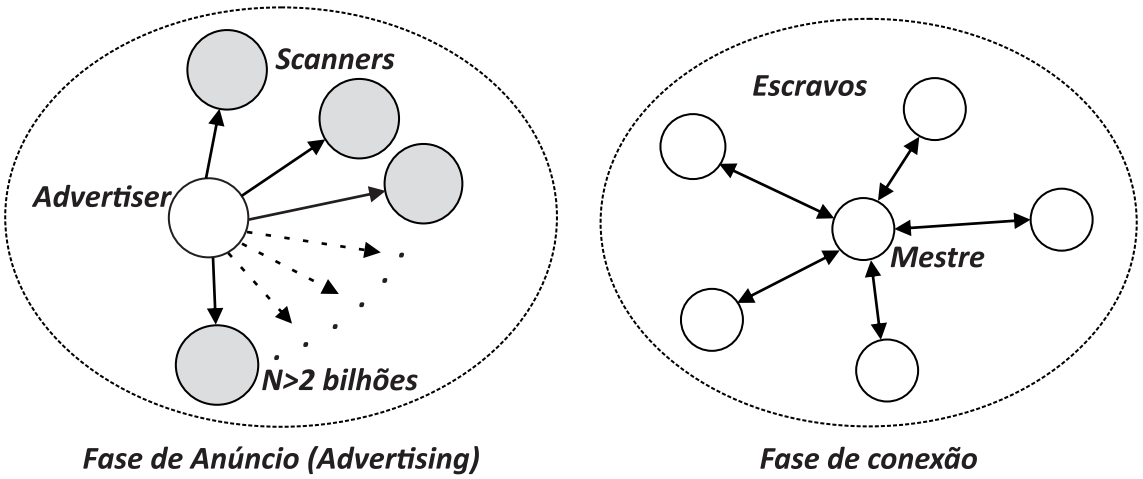

Figura 8: Topologia da rede nas fases de advertising e conexão [54]

O intervalo de envio do *advertising* pode variar entre 20 ms e 10,24 s. Para um pacote de *advertising* ser recebido pelo *scanner* é necessário uma sobreposisção da janela de *scan* com as mensagens periódicas de *advertising*. A Figura 9 ilustra um cenário onde pode ser visualizado a relação entre o janelamento do *scanner* e o intervalo de *advertising*. Contudo, o intervalo de *advertising* e a frequência e

intervalo do *scanner* estão diretamente relacionados com o consumo de energia dos dispositivo BLE.

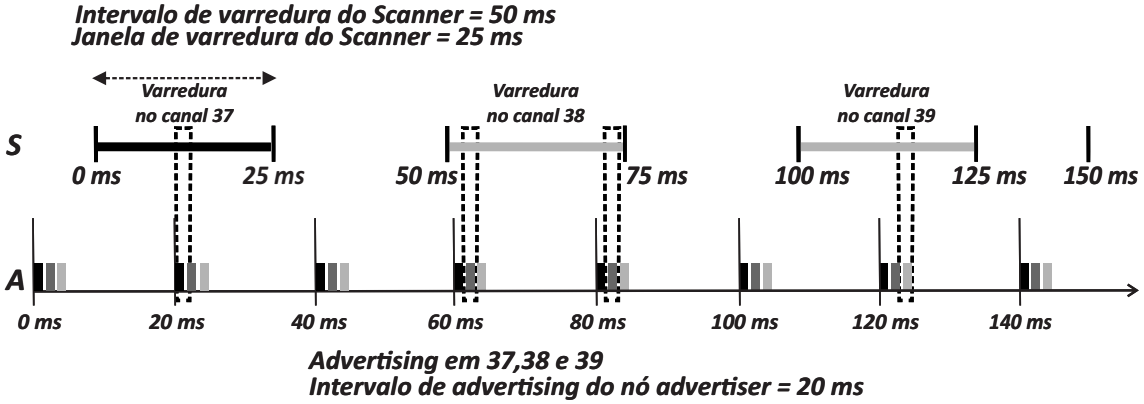

Figura 9: Intervalo de *Advertising* e *scanner* [66].

Há duas formas de realizar os procedimentos de (*scanning*):

1. **Passivo:** O *Scanner* simplesmente esculta os pacotes de *advertising* e não há uma mensagem de retorno para confirmar a recepção do pacote;

2. **Ativo:** O *Scanner* envia um pacote de solicitação (*Scan Request*) após receber uma mensagem de *advertising*, então o *Advertiser* responde com o *payload*. Dessa forma, é necessário sempre dois pacotes de *advertising* para efetivar a comunicação.

O pacote de *advertising* pode ser classificados segundo 3 propriedades: conectividade, dirigibilidade e tipo de 'escaneabilidade'. Destacam-se os dois primeiros. Primeiramente em relação a conectividade o pacote pode ser conectável, quando possibilita estabelecer a operação de conexão e não conectável, quando o pacote possui apenas aplicação de *broadcast*. Quanto a dirigibilidade o pacote pode ser direcionado, quando possui apenas dados no *payload* para estabelecer uma conexão, direcionado a um determinado endereço de um dispositivo Bluetooth *Scanner*, já o pacote não direcionado possui nenhum dispositivo alvo *Scanner* em particular e possibilita um *payload* com dados do usuário a ser transmitido a múltiplos destinos sem ser direcionado.

O protocolo de mensagem *advertising* possui 4 campos obrigatórios: preâmbulo, endereço de acesso, *Protocol Data Unit (PDU)* e o CRC. O PDU é utilizado

para incluir dados que descrevem o *broadcaster* e suas capacidades, além de informações adicionais. O BLE permite, além do pacote advertising padrão, um segundo pacote de *advertising* opcional, conforme referenciado em [67].Para ilustrar como esses pacotes são estruturados, a Figura 10 exibe o formato das mensagens de *advertising* opcional.

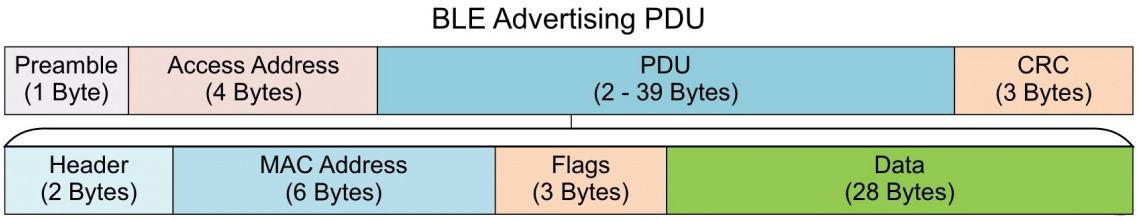

Figura 10: Estrutura das mensagens de *Advertising* [14].

3.4.2.3 Interface HCI

Host Controller Interface é um padrão de procolo que permite a comunicação entre o bloco hospedeiro e o controlador, baseados em um conjunto de comandos e eventos, juntamente com regras de controle de fluxo. Do ponto de vista da especificação física, o hospedeiro (e a camada de aplicação) é executado na CPU principal do hardware, enquanto o controlador é localizado em um *chip* transceptor BLE. A interface HCI conecta a CPU e o *chip* por por meio de um protocolo: *Universal asynchronous receiver/transmitter* (UART), *Universal Serial Bus* (USB), *Secure Digital Input Output* (SDIO), entre outros [66].

3.4.2.4 *Logical Link Control and Adaptation Protocol (L2CAP)*

O objetivo dessa pilha é dividida em duas funcionalidade. Primeiramente ela possui uma função de encapsular os diversos protocolos das camadas superiores em um formato de pacote BLE padrão e descompactar o pacote BLE oriundo das camadas inferiores para a camada de aplicação. Além disso, a camada também realiza combinação e fragmentação de pacote. Na transmissão ele fragmenta os pacotes das camadas superiores quando ultrapassa o *payload* de 27 bytes e também combina os pacotes das camadas inferiores para as superiores em caso de recepção de mensagens. A camada L2PAC também é responsável por suportar os protocolos de Gerenciamento de Segurança e Protocolo de Atributo [66].

3.4.2.5 Protocolo de Atributo (ATT)

O Protocolo de Atributo é um protocolo cliente/servidor baseado em atributos apresentados por um dispositivo. Cada dispositivo BLE é um cliente, um servidor ou ambos, independente de ser mestre ou escravo. Os dados são salvos em um servidor de atributo que o cliente de atributo pode ler e escrever [66]. Há 6 tipos de mensagens entre o cliente e o servidor de atributo:

1. requisição enviadas do cliente para o servidor;
2. respostas as solicitações do cliente;
3. comandos do cliente para o servidor sem respostas;
4. notificação enviadas do servidor para o cliente sem confirmação;
5. indicações/confirmações assíncrona enviadas do servidor para o cliente;
6. resposta do cliente ao servidor referente a uma indicação do servidor.

3.4.2.6 *Security Manager (SM)*

A camada de SM estabelece conexão utilizando um conjunto de algoritmos de segurança projetados para fornecer a pilha de protocolos BLE a capacidade de gerar e trocar chaves de segurança que permitem comunicação em pares de forma segura através de uma conexão criptografada. Em uma comunicação em *broadcast* essa camada é utilizada, sobretudo, para detecção do dispositivo BLE. Visto que as principais funcionalidades estão relacionadas a uma comunicação baseada em conexão [68].

3.4.2.7 *Generic Access Profile (GAP)*

O *Generic Access Profile (GAP)* é o pilar que permite a interoperabilidade de dispositivos BLE [68]. A camada apresenta uma estrutura que uma implementação deve seguir e assim permitir que os dispositivos realizem operações padrão. Os aspectos de interações dos dispositivos definidos pela GAP são:

1. Papel: cada papel impõe restrições e requisitos comportamentais. Um dispositivo pode possuir mais de um papel ao mesmo tempo;

2. Modos: um modo é um estado que dispositivo estar por um intervalo de tempo para permitir que um dispositivo realize um determinado procedimento;

3. Procedimento: Um procedimento é uma sequência de ações, normalmente relacionado a um controle da camada de enlace;

4. Segurança: GAP controla os modos e procedimentos de segurança que definem e aplicam o nível de segurança exigido para uma troca de dados especifica.

3.4.2.8 *Generic Attribute Profile (GATT)*

Acima do GAP o *Generic Attribute Profile (GATT)* define os tipos de atributos e como eles serão organizados e trocados entre as aplicações. Essa camada mantém a mesma arquitetura cliente/servidor do ATT, porém os dados são encapsulados em serviços, que consistem em uma ou mais características. Cada característica pode ser definida como uma união de pedaços de dados dos usuários, juntamente com o metadado que descreve informações dos valores, assim como propriedade, nome visível para o usuário, unidade e outras informações [68].

3.4.3 Biblioteca Python BLE: bluepy

A biblioteca *bluepy* é um projeto que fornece uma API para o acesso a dispositivo BLE partir da linguagem Python em dispositivos Linux, a biblioteca foi desenvolvida, sobretudo, para o módulo computacional Raspberry Pi [69]. Dentre as classes da bliblioteca, destaca-se a classe *scanner* que é baseada em um objeto para realizar a varredura dos dipositivos BLE que estão transmitindo dados de *advertising* e retornar uma lista de objetos com as informações dos *Advertiser* descobertos, conforme o tipo do dado do *advertising* [70], podendo ser utilizada para aplicações de *broadcasting* com *beacons*.

3.5 Zephyr OS

Zephyr é um sistema operacional de tempo real (RTOS, do inglês *Real Time Operating System*) baseado em *kernel* e sistemas embarcados, com suporte a variadas

arquiteturas e permissivamente licenciado sob a *Apache 2.0 license*. As arquiteturas suportas pelo *kernel* do Zephyr são diversas, incluso a arquitetura *Advanced RISC Machine* (ARM) (Cortex-M, Cortex-A e Cortex-R). Quanto as funcionalidades destacam-se: serviços de *kernel* (*Multi-threading*, interrupção, alocação de memória, gerenciamento de bateria e outros); arquitetura cruzada; proteção de memória; definição de recurso em tempo de compilação; BLE 5.0; desenvolvimento em Linux, macOS e Windows; poderosa estrutura de registro multi *back-end*; interface *shell* amigável ao usuário; configuração e armazenamento de memória não volátil [71].

3.6 nRF52840 *Dongle*

O nRF52840 *Dongle* é um dispositivo de baixo custo USB *Dongle* para aplicações de BLE, Bluetooth, *mesh*, Zigbee e IEEE 802.15.4 utilizando o *System-on-a-chip (SoC)* nRF52840 da empresa *Nordic Semiconductor*. O SoC possui um ARM Cortex M4, 15 GPIO's, *Digital Signal Processor (DSP)*, extensões para *advertising*, *throughput* de 2 Mbs, 1 MB de mémoria Flash e 256 kB de memória RAM [72].

Na família do nRF5x o *host* e controlador são implementados pelo *softDevice* que dá suporte aos modos (central, observador, periférico e *broadcaster*). O *softDevice* possui uma API para as camadas GATT, GAP e L2CAP [73], conforme a Figura 11.

O *softDevice* é uma imagem binária pré-compilada que implementa o Bluetooth 5.1 Low Energy na pilha de protocolo do nRF52. O *softDevice* consiste em três componentes principais: implementação de uma API para compartilhar os recursos de hardware com a aplicação coexistente, API para gerenciamento do *softDevice* (habilitar e desabilitar) e implementação das camadas do protocolo BLE.

O *Software Development Kit* (SDK) do nRF5 complementa o *softDevice* com as implementações das aplicações, interagindo por meio de API. O SDK se integra com o *Zephyr RTOS*, também inclui *middware* como os protocolos *Constrained Application Protocol* (CoAP) e *Message Queuing Telemetry Transport* (MQTT). Além de diversas bibliotecas, *drivers* de hardware e *bootloader* seguro. O Zephyr, encapsulado pelo SDK, além de interagir com *softDevice*, sob a própria pilha de

protocolo BLE, realiza também a interação e configuração dos recursos de *hardware* do SoC [74].

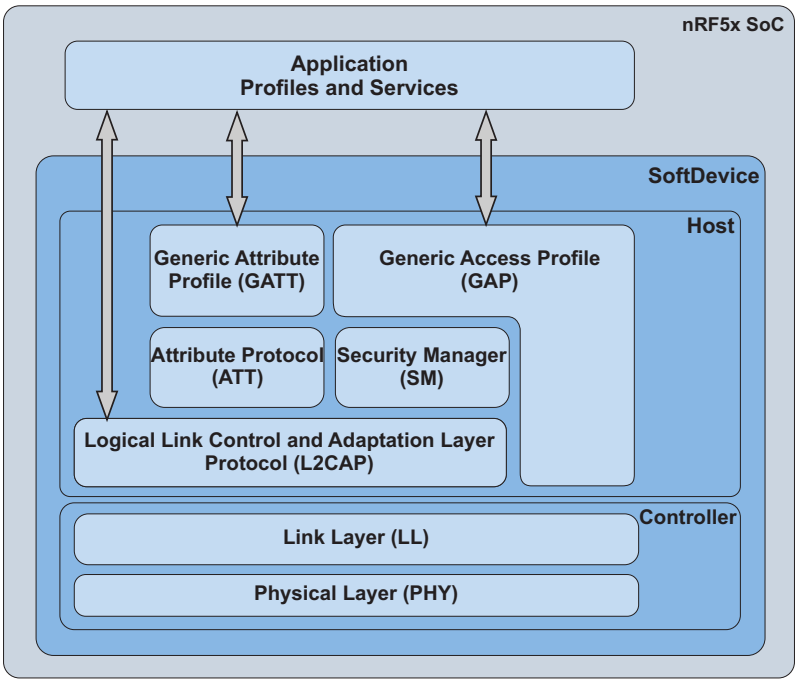

Figura 11: Pilha de protocolo do nRF52 [73].

3.7 Sensores

3.7.1 Sensor de Gás BME680

O BME 680 é um sensor digital compacto (3x3x1 mm) do segmento de sensores da empresa BOSCH. O sensor realiza medições de gás, umidade, pressão e temperatura. Além disso, possui baixo consumo, visando dispositivos energizados por bateria (para monitoramento de pressão, umidade e temperatura). O consumo é de 3,7 μA a 1 kHz de *refresh rate*. As principais aplicações do *chip* é o monitoramento da qualidade do ar, IoT, aplicações de navegação e esporte. A interface de comunicação se dá por meio dos protocolos *Serial Peripheral Interface* (SPI) e *Inter-Integrated Circuit* I^2C [75].

3.7.2 Sensor de umidade do solo

Um sensor de umidade do solo avalia a resistência elétrica entre seus eletrodos, resistência esta que se altera conforme a quantidade de água presente no

solo, devido à capacidade da água de conduzir eletricidade. A resistência elétrica medida pelo sensor é traduzida em uma leitura de umidade do solo por meio de um microcontrolador. O sensor conta com um trimpot ajustável que permite definir os pontos de ativação ou desativação do sensor e calibrar sua sensibilidade conforme condições específicas, como tipos de plantas ou solos.

Capítulo 4

Metodologia da proposta

Neste capítulo, será abordado a proposta do *FloatingBlue*, uma arquitetura de comunicação de RSSF tolerante a atraso e desconexões, baseada nos protocolos de comunicação *Bluetooth Low Energy* e *Bundle Protocol 7*. A proposta foi concebida, inicialmente, na crescente utilização de nós concentradores móveis (*Gateway* móvel) em arquiteturas de redes de sensores, com o objetivo de reduzir o consumo dos nós de terminação (*End Nodes* - EN). Dessa forma, os *gateways* móveis se tornam uma solução alternativa ou complementar às LPWAN que normalmente utilizam *gateways* ou estações bases estáticas como elementos concentradores. Por meio dos nós concentradores móveis (CM) é possível atingir uma área de cobertura elevada e minimizar o consumo de energia dos nós de terminação que possuem maior restrição de energia.

Entretanto, as redes com nós CMs estão sujeito a diversas desconexões devido a sua trajetória. Assim, faz-se necessário a utilização de protocolos de comunicação sujeitos a atrasos e desconexões. Com isso, o *FloatingBlue* é uma arquitetura de comunicação de RSSF que possui como elemento principal os nós CMs que se comunicam entre si por meio de uma DTN, enquanto se comunicam com os nós ENs por meio do protocolo BLE. A Figura 12 ilustra a arquitetura.

O CM1 (3) é um dispositivo microcontrolador móvel que pode estar acoplado a algum veículo, tripulado ou não, conforme a aplicação, sendo cada CM responsável por uma sub-rede. Uma sub-rede possui, além dos CMs, nós sensores estáticos, EN (4, 5). O nó CM realiza uma rotina de deslocamento para captura das informações dos ENs ao entrar na zona de alcance de comunicação dos ENs. Os dados dos ENs

são transmitidos e ficam salvos no CM, sendo em seguida encaminhados para outros nós CMs (1) até serem descarregados em uma estação central de processamento (2).

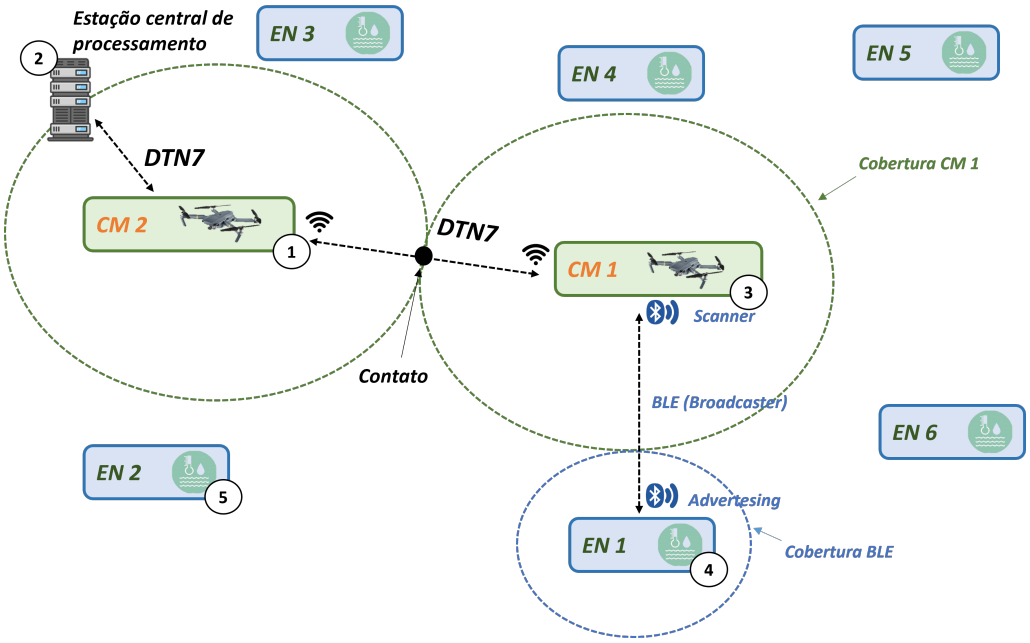

Figura 12: Proposta da arquitetura de RSSF *FloatingBlue*

A comunicação entre os ENs e CMs é realizada por meio do protocolo BLE no modo *broadcasting*. Cada CM realiza rotinas de varredura periódicas, enquanto o EN realiza transmissões dos pacotes de *advertising* com os dados coletados pelos sensores integrados. Já os nós CMs se comunicam entre si utilizando o protocolo DTN 7, e nas as camadas inferiores o TCP-IP (TCPLC e IPv6/IPv4) e o IEEE 802.11. A Figura 13 ilustra as camadas dos protocolos adotados na arquitetura *FloatingBlue*, apresentado anteriormente na Figura 12.

Na Figura 13, um EN transmite uma mensagem BLE em *broadcast*, que é recebida pelo nó CM 1 (3) e encaminha a mensagem para o CM2 (1), até chegar no servidor de aplicação (2). Os detalhes de implementação das camadas são comentados nos próximos tópicos.

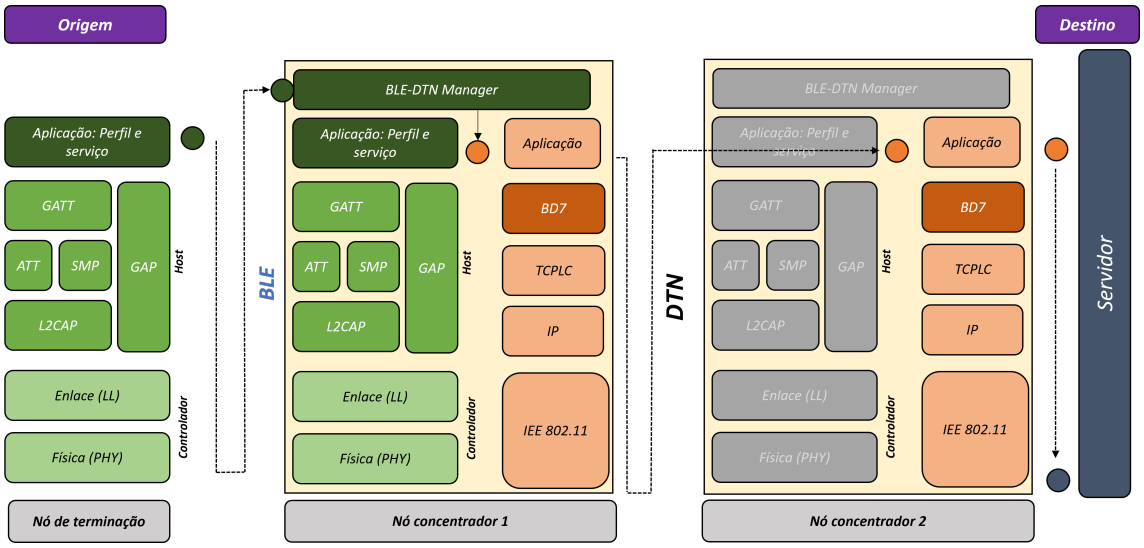

Figura 13: Pilhas de protocolo do *FloatingBlue*.

4.1 Nó concentrador móvel

Conforme mostrado na Figura 14, o nó CM foi desenvolvido usando um Raspberry Pi 3 Model B, que foi instalada em um Drone de modelo DJI Mavic 2 Zoom. O Raspberry Pi 3 tem capacidades de comunicação IEEE 802.11ac (WLAN) e Bluetooth 4.1, permitindo a utilização dos protocolos BLE e DTN7 sobre Wi-Fi na arquitetura proposta

Figura 14: Nó coordenador móvel composto de uma Raspberry Pi 3 Model B afixada em um Drone modelo DJI Mavic 2 Zoom.

A pilha de protocolos que implementam o protocolo DTN 7 (modelo de referência TCP-IP) são as camadas de transporte, rede e acesso a rede respectivamente os protocolos TCPLC, IP e IEE 802.11. Enquanto a pilha de protocolos do BLE segue o apresentado na arquitetura da Figura 6. O DTN 7 que é implementado no

CM é o dtn7-go. As camadas de ambos os protocolos DTN 7 e BLE implementados no nó CM podem ser visualizadas na Figura 13.

O nó CM opera de duas formas: função coleta e função de encaminhamento. Na função de coleta o CM implementa ambos os protocolos, utilizando o BLE para receber os dados transmitidos pelos EN e o protocolo DTN 7 para codificar os dados, em formato de *bundle*, a fim de que sejam transmitidos até o nó destino. Já na função nó de reencaminhamento o nó implementa o *bundle protocol 7*, realizando as aplicações armazenamento persistente (com ou sem transferência de custódia) do *bundle*.

Na camada de aplicação foram implementados no trabalho serviços que são inicializados do módulo computacional raspiberry pi. Esses serviços foram desenvolvidos na camada nomeada *BLE-DTN Manager*, conforme apresentado na Figura 13. Dois serviços estão relacionados com a inicialização do protocolo DTN 7, os demais serviços são divididos em *BLE manager* e *DTN manager* e estão relacionados com a integração das pilhas de protocolos DTN 7 e BLE. A Figura 15 ilustra o fluxo de mensagens na camada de aplicação de um CM.

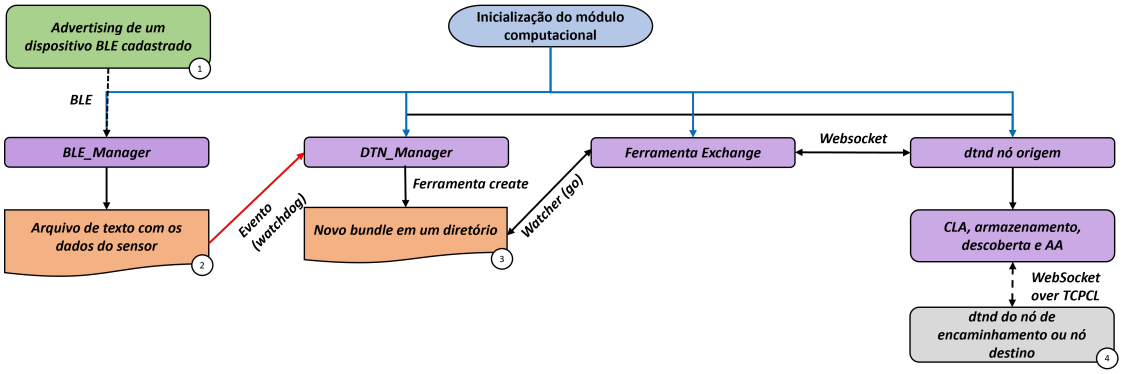

Figura 15: Fluxo de mensagem do nó concentrador.

O ciclo inicia primeiramente ao receber o *advertising* (1) de um EN cadastrado no dispositivo CM. Em caso de EN cadastrado, os dados dos EN são coletados, formatados e escritos em um arquivo de texto (2), por meio do serviço *BLE_Manager*. Posteriormente o serviço *DTN_Manager* possui a função de transformar esse arquivo em um *Bundle* DTN 7 (3). Paralelamente a ferramenta *exchange* é inicializada e monitora o diretório, onde é salvo os *bundles*, a partir de um módulo *watcher* desenvolvido em linguagem GoLand. Na ocorrência de um *bundle* do diretório, envia o pacote para o *daemon dtnd*. A própria ferramenta *exchange* também inicializa um

agente *Websocket* no canal do nó destino (4), encapsulado pelo protocolo TCPLC, a fim de encapsular/encaminhar o *bundle* até o destino.

4.1.1 Inicialização e configuração do DTN 7

Conforme ilustrado na Figura 15, dois serviços são inicializados juntamente com o módulo computacional. Esses serviços são, primeiramente, para ativar o *daemon dtnd* passando como parâmetro o arquivo de configuração, enquanto o segundo serviço inicializa a dtn-tool na função *exchange* passando como atributos do ip *websocket* do nó destino, EID do remetente e diretório observado.

```
./dtn-tool exchange websocket endpoint-id directory
```

Já as configurações relacionados ao dtn7-go são editadas no arquivo TOML, com destaque para as seguintes configurações: tipo de protocolo da camada de rede e transporte, *node id* e algoritmo de roteamento. Como será utilizado comunicação *websocket*, adota-se na camada de transporte o TCPLC e na camada de rede ambos os protocolos de internet (IPV4 e IPV6). Quanto ao algoritmo de roteamento emprega-se o algoritmo epidêmico devido a tipologia da rede com poucos nós concentradores.

4.1.2 Serviço *BLE Manager*

Na Figura 15, o *BLE_ manager.service* é um serviço que implementa um *script* Python na versão 3.9, utilizando a biblioteca bluepy, a fim de coletar os parâmetros mensurados pelo EN's. A Figura 16 ilustra o fluxograma do algoritmo implementado no serviço *BLE_ manager*. O serviço possui em formato de dicionário, alguns endereço de dispositivos BLE (PDA), previamente cadastrados. Então o serviço realiza rotinas de varredura em laço com um intervalo definido visando encontrar dispositivos BLE que estão ao alcance do nó CM. Após realizar o procedimento de varredura os dados dos ENs encontrados pelo CM é gerado uma lista de objetos (classe *ScanEntry* da biblioteca bluepy) que são comparados são comparados com o dicionário de dispositivos BLE cadastrados. Caso o serviço encontre o endereço de um dos nós cadastrados, o valor do *data type* BLE 0x2D (*Broadcast Code*) é extraído e salvo em um arquivo de texto, juntamente com o PDA, visando distinguir

os ENs, e o instante que ocorreu a escrita. Com isso, é possível obter dados para eventuais análises de série temporal dos parâmetros coletados pelos diversos ENs. O arquivo de texto é salvo em um diretório destinado apenas para coletar os dados dos ENs.

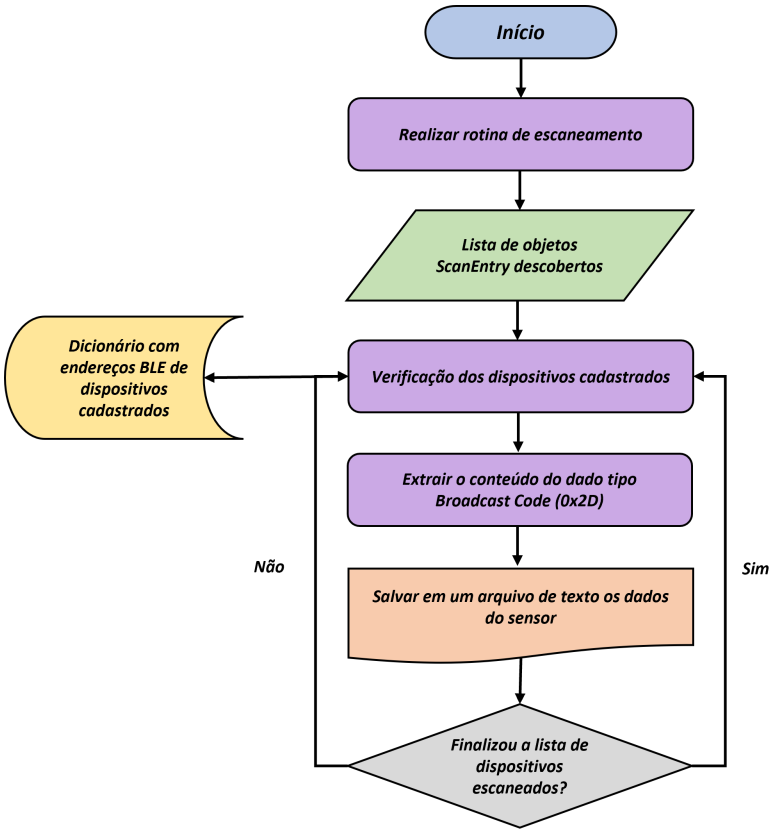

Figura 16: Fluxograma do algoritmo *BLE Manager*.

4.1.3 *Serviço DTN Manager*

Na Figura 15, o serviço *DTN_manager.service* possui a funcionalidade de transformar o arquivo de texto da saída do serviço *BLE Manager* em um *bundle*. A Figura 17 ilustra o fluxograma das etapas envolvidas. O serviço também foi implementado na linguagem Python versão 3.9 e possui como biblioteca principal a *Watchdog 0.8.2*, uma biblioteca para monitoramento de eventos de arquivos de sistemas. No *script* é definido o caminho do diretório para monitoramento e o tipo de evento monitorado no diretório. Foi definido o evento do tipo criação de arquivo (método *on_created* da biblioteca *Watchdog*) para estabelecer a sinalização do evento. O diretório adotado para ser monitorado é o diretório que salva os dados de saída do *BLE_manager.service*. Na ocorrência de um evento de criação a

informação do evento fica salvo em um objeto. Após isso, é extraído do objeto de sinalização o caminho do arquivo que foi criado no diretório monitorado.

Em seguida utiliza-se o *dtn-tool* para transformar o arquivo de texto da informação do sensor em um *bundle*, por meio de biblioteca *subprocess* que executa o comando *dtn-tool create* passando o caminho do arquivo, o EIR do nó destino e o EIR do remetente.

```
./dtn-tool create sender receiver -|filename
```

Os *bundles* criados ficam salvos na pasta onde encontra-se a aplicação do *dtn-tool* e é monitorada pela função *exchange* do *dtn-tool*.

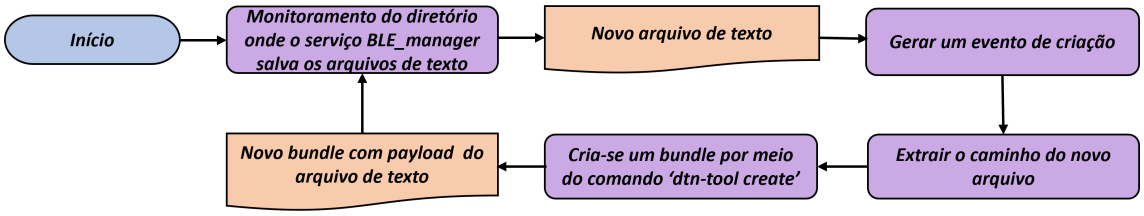

Figura 17: Fluxograma do algoritmo *DTN Manager*.

4.2 Nó de terminação

O nó de terminação (*end node* - EN), apresentado na Figura 12, foi construído usando o dispositivo nRF52840, juntamente com o sensor de gás BME680 e um sensor de umidade do solo. A comunicação entre o nRF52840 e o BME680 foi feita através de uma interface I^2C, usando os pinos 0.31 e 0.29 do nRF52840 para sincronização e transmissão de dados, respectivamente. O valor analógico do sensor de umidade do solo foi lido pelo pino ADC 0.02 do nRF52840. Os sensores de gás e umidade consomem 2 uA e 2,5 mA de energia, respectivamente.

O dispositivo nRF52840 foi conectado ao sensor de gás BME680 e ao sensor de umidade do solo através de uma Placa de Circuito Impresso (PCI), projetada usando a plataforma EasyEDA. Após o *design*, a PCI foi fabricada usando uma fresadora CNC (*Computer Numerical Control*) e o invólucro do EN foi criado com uma impressora de manufatura aditiva. A Figura 18 ilustra o dispositivo EN.

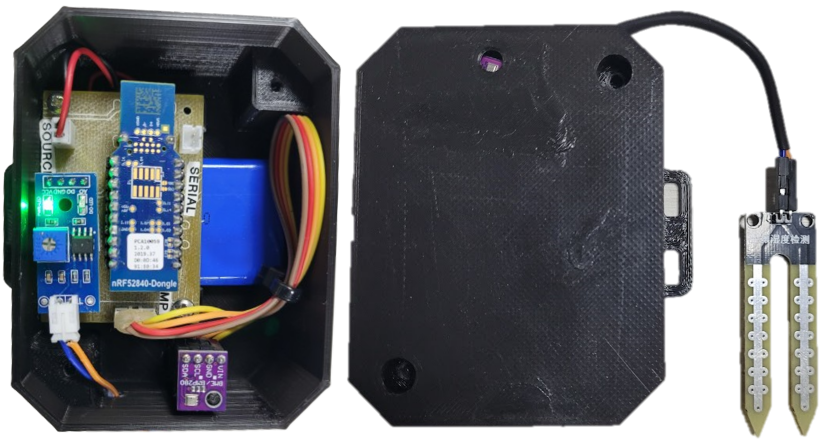

Figura 18: Fluxograma do algoritmo *Nó de terminação - EN*.

O software embarcado é baseado SDK nRF *Connect*, versão 2.3, utilizando a IDE *Visual Studio Code*. O método de programação do EN foi realizado por meio de *Device Firmware Update* (DFU) presente no SDK. O código foi desenvolvido sobre as funções do Zephyr. As características que foram admitidas para o EN foram de um *beacon* com pacotes de *advertising* não conectáveis e não direcionados.

A arquitetura proposta transmite dados de temperatura e umidade do ar e do solo via mensagens de *advertising* BLE originadas nos dispositivos EN e armazenadas no campo *Broadcast Code* do BLE (hexadecimal 0x2D). Especificamente, são usados 16 bytes para a temperatura do ar, 8 bytes para a umidade do ar e 8 bytes para a umidade do solo. A Figura 19 ilustra a rotina do *firmware* no dispositivo EN.

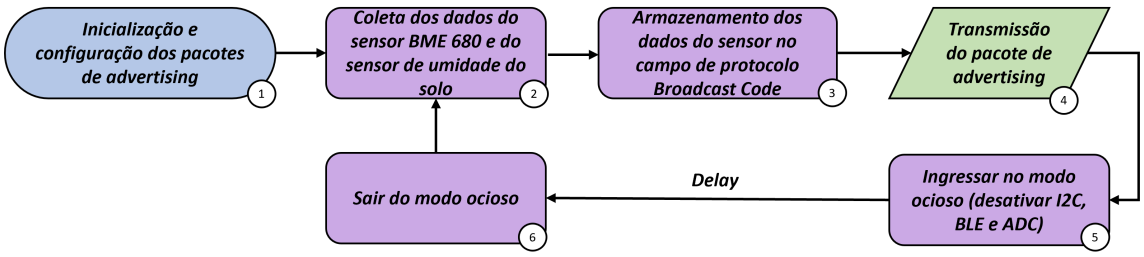

Figura 19: Rotina do software embarcado do nó de terminação (EN).

A rotina do *firmware* do nó de terminação (EN), segue processo, conforme a seguir: (1) Configuração do cabeçalho de *advertising*: Inicialmente, o *firmware* estabelece o cabeçalho dos pacotes de *advertising*. Nesta fase, são definidas as variáveis que compõem a mensagem de *advertising*, com foco no quadro *broadcast code*. Também ocorre a inicialização do Bluetooth e a definição de sua transmissão no modo

broadcast, incluindo a configuração do intervalo entre as mensagens. (2) Inicialização do periférico I²C e coleta de dados: Em seguida, o *firmware* ativa o periférico I²C do microcontrolador. Utilizando uma função do sistema operacional Zephyr para o sensor BME680, o *firmware* coleta dados deste sensor. A coleta requer a especificação dos registradores de endereço do sensor na árvore de dispositivos do sistema. Além disso, dados do sensor de umidade são coletados através do uso de funções de captura do conversor analógico-digital (ADC). (3) Armazenamento dos dados: Os dados coletados são salvos em um *buffer* referenciado pelo *broadcast code* e utilizado para transmitir as informações coletadas. Em seguida (4), é chamado uma função para atualizar a mensagem de *advertising*. Por fim, (5) é desativado o rádio BLE e os periféricos I²C e ADC, o sensor entra no estado ocioso, com o objetivo de reduzir o consumo do dispositivo EN. Após um intervalo de tempo programável o dispositivo sai do modo ocioso, acionando os periféricos desativados e retornando, em uma lógica de laço, para o passo 2.

Capítulo 5

Resultados

A proposta *FloatingBlue* foi testada em um ambiente agrícola para verificar sua eficácia. Durante os testes, avaliou-se não apenas a funcionalidade geral do sistema, mas também o consumo de energia e o alcance de comunicação dos nós EN, que foram projetados para serem eficientes em termos energéticos em áreas remotas. Além disso, observou-se como o alcance de comunicação dos ENs influencia a movimentação dos nós CM, uma vez que estes precisam permanecer dentro da área de cobertura dos ENs para a coleta efetiva de dados.

5.1 Avaliação do estudo de caso

5.1.1 Cenário de teste em um ambiente agrícola

Os testes foram realizados em uma área rural, no km 40 da BR-174, próximo de Manaus, Amazonas, em uma plantação de cítricos, incluindo laranjas e limões. A área, sem acesso à internet, é ilustrada na Figura 20.

Os testes ocorreram em condições climáticas claras, com temperatura média de 33,1 °C e 56% de umidade relativa do ar, dados obtidos do Instituto Nacional de Meteorologia. A Figura 21 mostra uma das três estações (ENs) colocadas no solo, com os endereços Bluetooth E5:E1:C3:96:ED:D4, D0:0D:46:91:E0:34 e CB:A3:E6:EE:17:06.

Figura 20: Ambiente para implementação e avaliação da proposta

Figura 21: EN instalado no ambiente de teste

Dois nós CM foram instalados em veículos para coleta e descarregamento de dados dos dispositivos ENs. O CM 1 foi colocado em um drone DJI Mavic 2 Zoom e o CM 2, em um trator agrícola. A Figura 22 apresenta uma visualização desses dois nós CM no ambiente de testes.

Figura 22: Nó CM1 instalado em um drone e nó CM2 instalado em trator rural.

A Tabela 4 detalha as características dos nós EN e CMs utilizados no cenário de teste. Um aspecto importante é o intervalo do modo ocioso do EN, fixado em 5 segundos. Foi observado que este intervalo não pode exceder 5,65 segundos devido a vários fatores: a altitude na qual os testes foram realizados, a distância máxima de comunicação eficaz do EN quando há linha de visão direta, e o padrão de voo do VANT que precisa passar na localização geográfica específica do EN. Para compreender melhor como o intervalo de tempo em modo ocioso do EN influencia a transmissão de dados ao nó CM, a Equação 2 fornece uma formulação matemática deste intervalo. Salienta-se que o raio de alcance do EN adotado na Tabela 4 foi avaliado para uma altitude de 30m para o drone.

$$T_{adv} < 2xRaioAlcance_{EN}/Velocidade_{VANT} \tag{2}$$

Tabela 4: Características do EN e do VANT para o cenário de teste

Descrição	**Valor**
Potência TX do EN	0 dBm
Raio de alcance do EN com visada	56,5 m
Velocidade máxima do VANT	20 m/s
Altitude do teste	30 m
Autonomia de voo	31 min
Intervalo do modo ocioso	5 s

No nó CM, foram configurados parâmetros para os protocolos BLE e dtn7-go. Para o BLE, a janela de tempo do *scanner* foi ajustada para 5 segundos, um tempo superior ao intervalo de *advertising* do EN, conforme a instrução comentada na sessão 3.4.2.2 (camada de enlace lógico do BLE). Os parâmetros para o dtn7 estão explicitados na Tabela 5.

Tabela 5: Configuração do dtn7

Descrição	Parâmetro
Protocolo de descoberta	IPV4 e IPV6
Websocket	ws://0.0.0.0:8080/ws
Protocolo CLA	TCPLC
Algoritmo de roteamento	Epidêmico
Node-id	dtn://node-name/

Durante os experimentos, um ponto de acesso Wi-Fi foi utilizado para coletar resultados via SSH e criar um canal WebSocket com o dispositivo CM2. Vale ressaltar que o ponto de acesso é um elemento opcional na arquitetura *FloatingBlue*. A Figura 23 oferece uma visão completa do cenário.

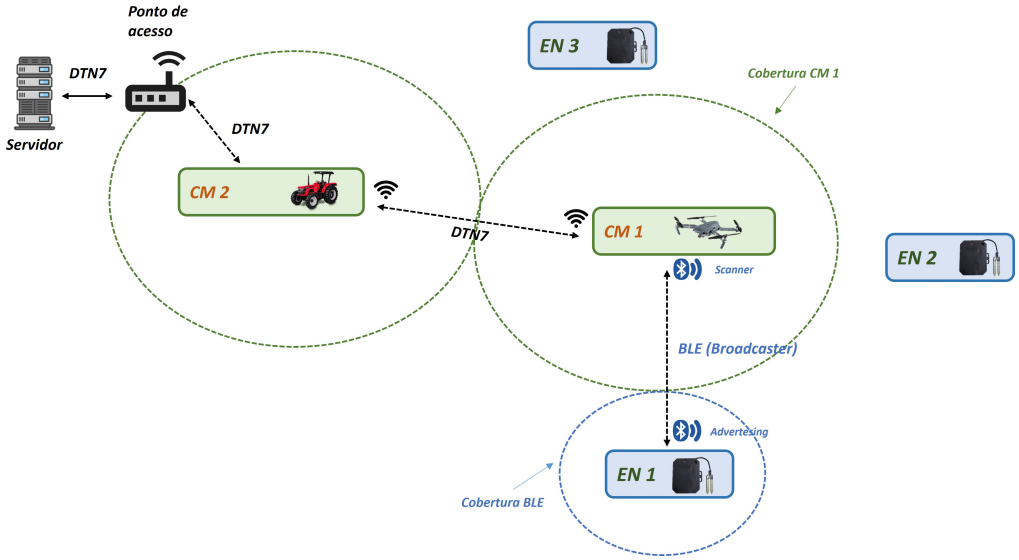

Figura 23: Cenário de teste completo do *FloatingBlue*

A Figura 24 mostra a disposição dos ENs e dos nós CMs, utilizando imagens de satélite obtidas através do software Google Earth. As coordenadas geográficas dos ENs e CMs foram adquiridas usando o relógio GPS *Garmin Forerunner 55*.

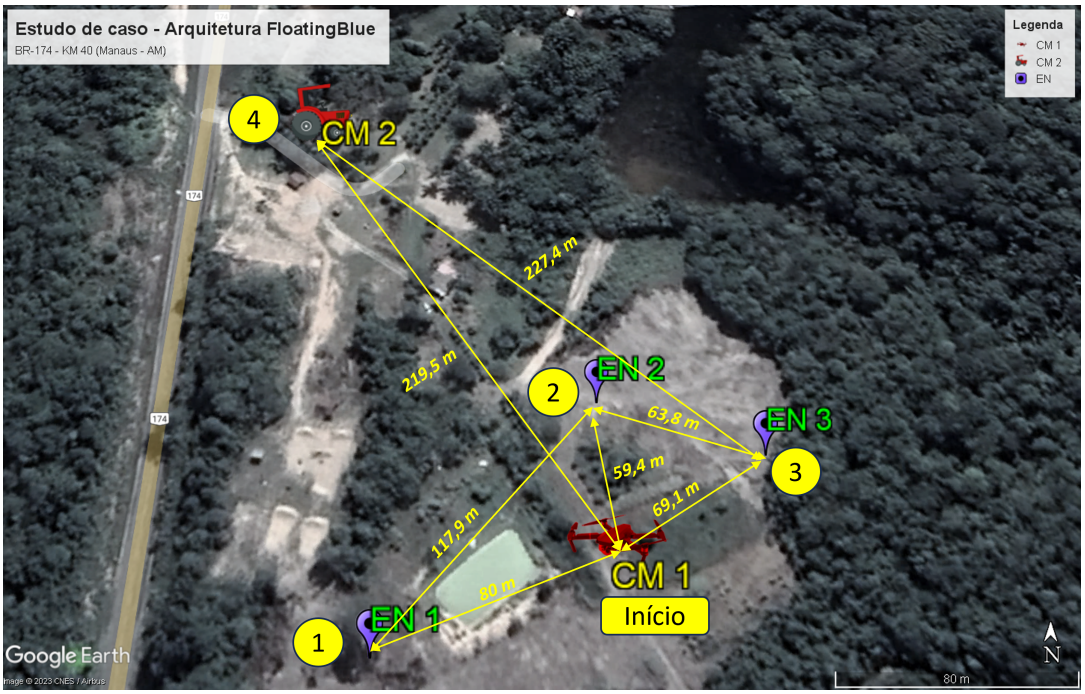

Figura 24: Localização das entidades da arquitetura e trajetória do CM 1

5.1.2 Implementação do estudo de caso

Conforme a Figura 24, o teste iniciou com a decolagem do nó CM 1 do ponto denominado "Início". O nó CM 2 estava a 219,5 metros do ponto Início e fora do alcance de comunicação do nó CM 1. Durante o teste, o trator com o nó CM 2 permaneceu parado. O VANT do nó CM 1 voou a uma altitude constante de 30 metros para evitar interferência com a vegetação local.

Após decolar, o nó CM 1 seguiu as coordenadas GPS, sobrevoando sequencialmente os pontos EN 1, EN 2 e EN 3. Em cada EN, ele permaneceu parado por 3 a 5 minutos para coletar ao menos 45 mensagens de *advertising*. Depois, sobrevoou o trator do nó CM 2 e retornou ao ponto 'Início'. A Figura 24 mostra essa trajetória. Para confirmar o processo de desconexão, o ponto de acesso Wi-Fi utilizado para coletar resultados via dispositivo CM2, foi inicialmente desativado. Posteriormente, ele foi reativado apenas quando o nó CM1 estava sobrevoando diretamente o nó CM2.

5.1.3 Resultados

Nesta seção, analisa-se a arquitetura *FloatingBlue* com um nó EN transmitindo mensagens de advertising. Para a transmissão de mensagens destaca-se o

campo de Código de Transmissão (0x2D). A Figura 26 exibe um pacote de *advertising* transmitido pelo EN 3 e, observado através do aplicativo nRF Connect.

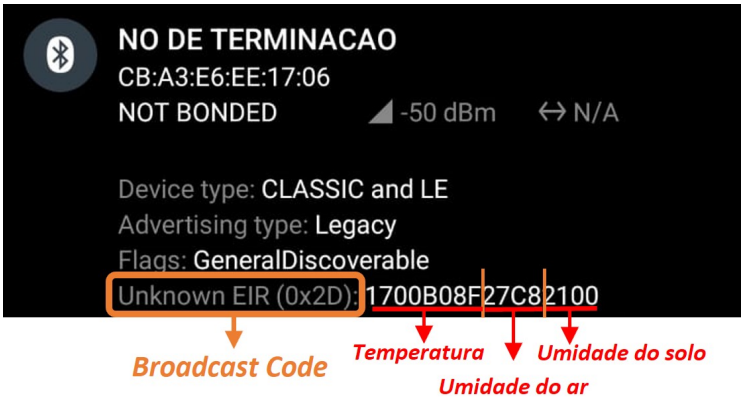

Figura 25: Pacote de advertising transmitido pelo EN 3 e, observado através do aplicativo nRF Connect.

O serviço *BLE_manager.service* instalado no nó CM 1 também teve seu funcionamento avaliado. Na Figura 26 pode-se visualizar o arquivo de texto "beacon.txt" gerado por esse serviço quando o nó CM 1 recebe pacote de *advertising* transmitido pelo EN de endereço PDA d0:0d:46:91:e0:34.

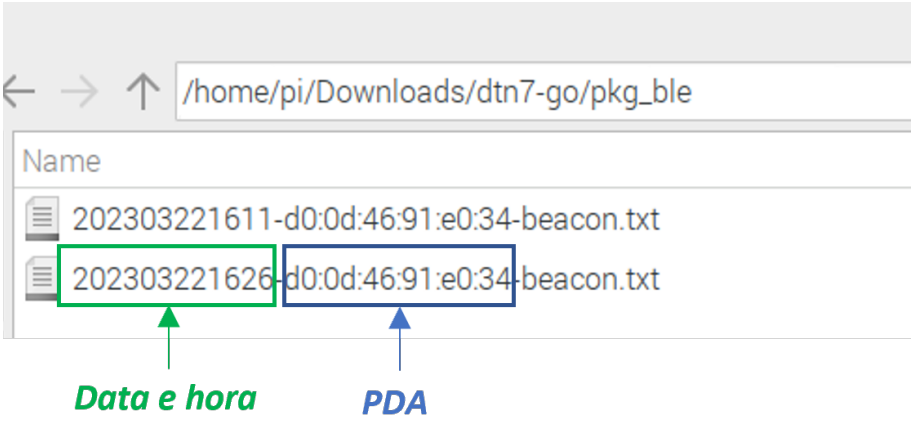

Figura 26: Arquivo de texto "beacon.txt" gerado pelo serviço *BLE_manager.service* quando o nó CM 1 recebe pacote de advertising transmitido pelo EN.

Em seguida, o serviço *DTN_manager.service* dos nós CM1 e CM2 foram analisados. Na Figura 27, pode-se ver os *bundles* transmitidos do nó CM1 para o nó CM 2. Estes *bundles* incluem o conteúdo referente ao arquivo de texto criado pelo serviço *BLE_manager.service*. Para conferir se o conteúdo estava íntegro, utilizou-se a ferramenta *dtn_show* do DTN7.

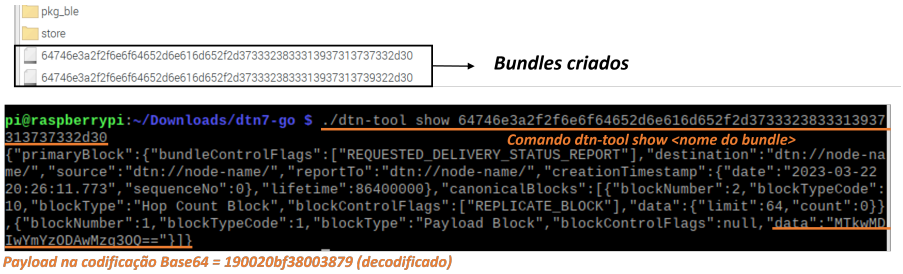

Figura 27: *Bundle* transmitido pelo nó CM1 foi recebido e validado pelo nó CM2.

Os nós CM1 e CM2 ficaram desconectados por 51 minutos. Nesse período, o CM1 produziu 437 bundles. Quando a conexão entre CM1 e CM2 foi reestabelecida, todos os bundles do CM1 foram transferidos para o CM2 com taxa de entrega (Packet Delivery Ratio - PDR) de 100%.

5.2 Avaliação do consumo energético

5.2.1 Cenário do teste

Para avaliar o consumo energético, utilizou-se o analisador de perfil de potência *Power Profile Kit II* da *Nordic Semiconductor* para medir a corrente elétrica consumida pelos nós EN. Os testes de consumo energético dos nós EN foram conduzidos utilizando tensão de alimentação de 1,8 V e, inicialmente, potência de transmissão de 0 dBm. O analisador de perfil de potência é um micro amperímetro de alta precisão conectado em série com a fonte de alimentação e o EN. O *Power Profile Kit II* também é conectado a um computador por meio de uma porta USB e o resultado das medições são visualizados por meio de uma interface gráfica disponibilizada pela *Nordic Semiconductor*, conforme visualizado na Figura 28.

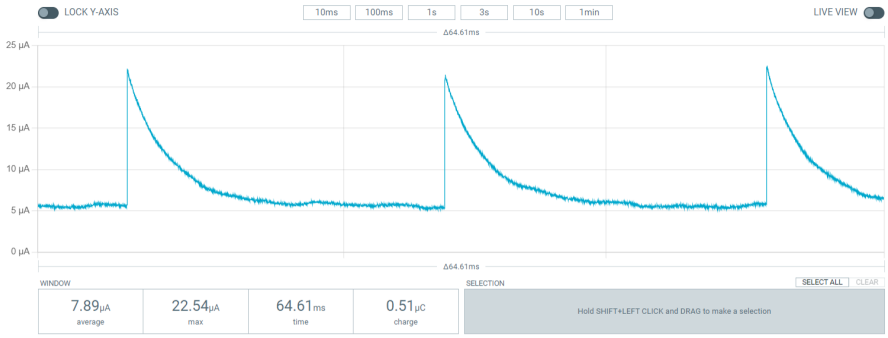

Figura 28: Interface do *Power Profile Kit* II

Os testes de consumo energético foram executados sem os sensores BME-680 e de umidade do solo a fim de avaliar apenas no consumo do *beacon* BLE, tornando a comparação mais próxima da literatura. Os resultados de cada ensaio foram determinados calculando a média de dez medições individuais. Primeiro, mediu-se o consumo do EN em modo ocioso e, em seguida, avaliou-se o consumo ao transmitir mensagens de *Advertising* em três cenários. No primeiro cenário, o intervalo de transmissão foi ajustado para 30 ms, 100 ms e 1000 ms, seguindo as opções no SDK BLE. Nesse cenário foi avaliado uma série temporal de 1 segundo para o teste de intervalo de *advertising* de 30 ms e 100 ms e, posteriormente, uma série temporal de 10 segundos para a configuração de intervalo de *advertising* de 1000 ms.

No segundo cenário de teste, o foco foi analisar como o consumo energético é afetado pela variação do tamanho do *payload* da PDU (*Packet Data Unit*) do BLE. É importante destacar que o *payload* máximo permitido é de 31 bytes, divididos entre 3 bytes para os campos de *Flags* e 28 bytes para os dados. O teste foi realizado com mensagens de *advertising* sendo transmitidas a cada 1 segundo de intervalo.

A última avaliação do consumo de energia envolveu a análise da relação entre a potência de transmissão (TX) do nó EN e o seu consumo energético. O módulo Nordic nRF52840 possibilita ajustar a potência TX do transceptor BLE em uma faixa de -40 dBm a 8 dBm.

5.2.2 Resultados

Nesta seção apresenta-se os resultados de avaliação de consumo energético do nó EN. A primeira métrica avaliada diz respeito ao consumo de energia do EN em modo ocioso. Nos três cenários de teste descritos na seção anterior foram observados o mesmo comportamento, um consumo de 13 μJ.

Em seguida, foram realizados testes específicos para medir o consumo energético do nó EN em função de diferentes intervalos de transmissão das mensagens de *advertising*. Os intervalos testados foram de 30, 100 e 1000 ms. Os resultados destes testes, mostram o consumo de energia do nó EN, em μJ, para cada um destes intervalos, estão representados na Figura 29. Verifica-se que à medida que se aumenta o intervalo de transmissão de mensagens de *advertising*, reduz-se o consumo energético do nó EN.

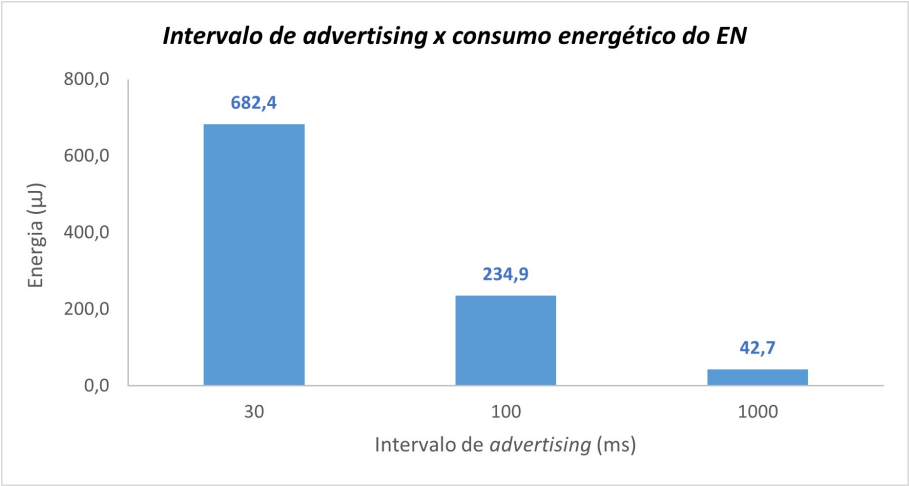

Figura 29: Relação do consumo energético de transmissão do EN com intervalo de transmissão de mensagem *advertising*.

De acordo com a Figura 30, foram realizados experimentos para analisar como o consumo energético do EN se altera quando o tamanho da mensagem PDU de *advertising* varia entre 0 e 31 bytes (veja a Figura 10, nos campos Flags e Data da PDU BLE de advertising). Posteriormente, como mostrado na Figura 31, foi examinado o consumo energético do EN com base na potência de transmissão do rádio Bluetooth no EN.

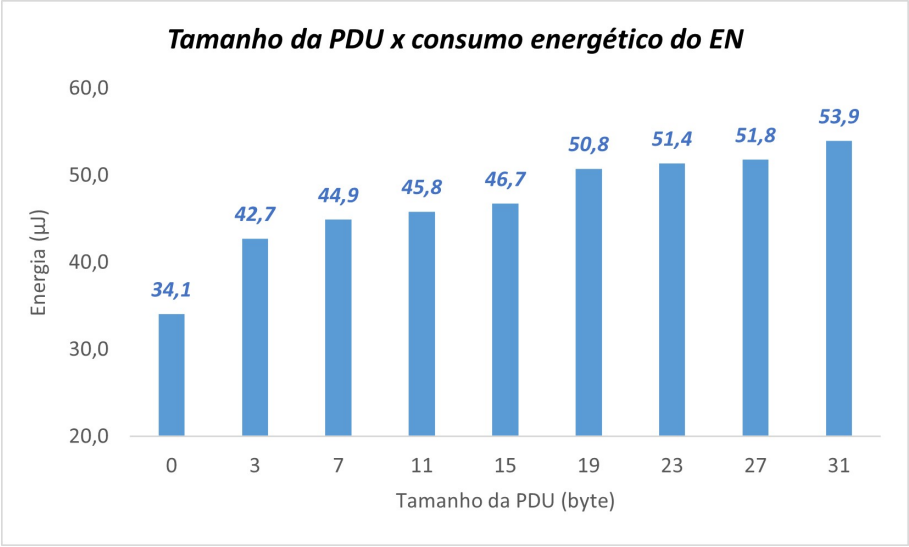

Figura 30: Relação do consumo de transmissão energético do EN com o tamanho do pacotes da PDU.

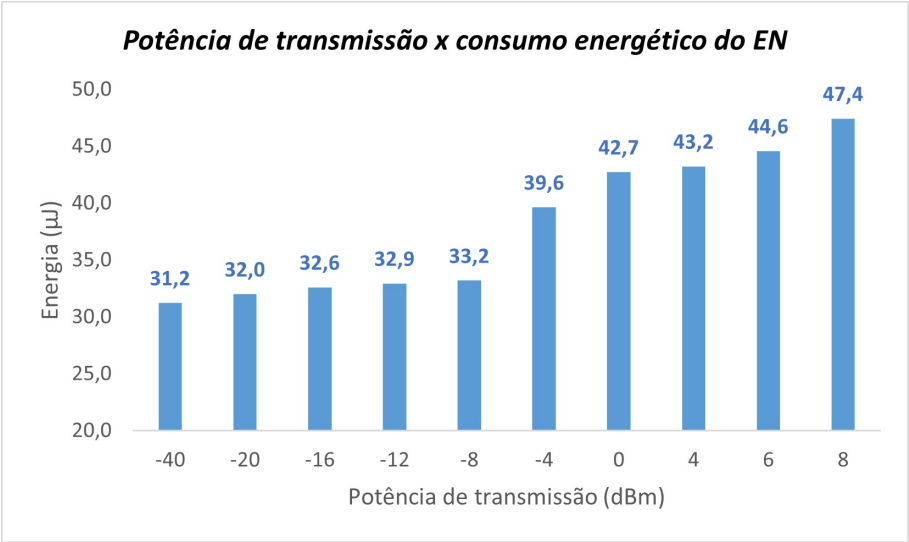

Figura 31: Relação do consumo energético de transmissão do EN com a potência TX do EN.

5.3 Avaliação do alcance da comunicação do EN

5.3.1 Cenário do teste

A avaliação do alcance da comunicação entre um nó EN e um nó CM foi conduzida em condições de visada direta, sem obstruções. Para determinar a distância entre o nó CM 1 (VANT) e um local de coordenadas GPS conhecidas, que coincidia com a localização do nó EN, foram utilizadas informações de geolocalização fornecidas pelo controle remoto do VANT.

Durante o teste de alcance, o VANT foi mantido na altitude de 30 metros, conforme estabelecido no teste. Após atingir a altitude de 30 m, o VANT foi deslocado horizontalmente no sentido norte, afastando o nó CM móvel do nó EN fixo, até que não houvesse mais comunicação entre os nós por um período igual ou maior a 5 segundos.

Foram realizados testes de alcance de transmissão usando mensagens de *advertising* de 3 bytes enviadas a cada segundo. O alcance foi avaliado com diferentes potências de transmissão (TX) do nó EN.

5.3.2 Resultados

A avaliação final focou no alcance de comunicação entre o EN e o CM. Embora o EN normalmente opere com potência de transmissão de 0 dBm, foi avaliado seu

desempenho em vários níveis de potência do rádio BLE, conforme Figura 32,.

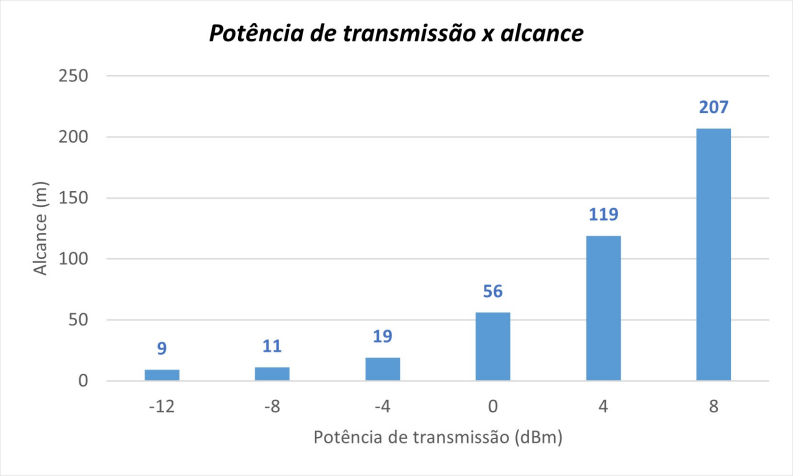

Figura 32: Relação do alcance de comunicação com a potência TX do EN.

5.4 Discussões

Os experimentos permitiram comparar o consumo de energia dos ENs mencionados na literatura científica, que usam nós concentradores móveis (também conhecidos como nós relés ou mulas de dados), com o EN deste estudo. A Tabela 6 mostra, para cada pesquisa, as tecnologias de comunicação utilizada nas RSSFs, a presença ou ausência do DTN, e os consumos energéticos dos ENs em modo ocioso e modo de transmissão.

Tabela 6: Análise comparativa de consumo energético

Trabalho	Ano	Tecnologia	DTN	Consumo Ocioso	Consumo Transmissão
[51]	2011	DIVERSAS	SIM	633 nJ	100 mW
[12]	2020	LORA	SIM	72 uW	648 uW
[11]	2020	LORA	SIM	231 mW	660 mW
[46]	2019	ZIGBEE	NÃO	50 uW	15 mW
[47]	2022	IEEE 802.11	NÃO	9 mJ	337 mJ
[48]	2020	IEEE 802.11	NÃO	12 uW	494 uJ
[49]	2016	BLE	NÃO	NA	57 mW
[13]	2018	LoRa	SIM	24 mW	27 mW
[43]	2007	ZIGBEE	SIM	80 mW	NA
[52]	2018	ZIGBEE	SIM	181 mW	825 mW
FloatingBlue	**2024**	**BLE**	**SIM**	**13 uJ**	**42, 7 uJ**

Os testes realizados validaram a arquitetura *FloatingBlue*. Em uma vasta área remota, os três nós ENs coletaram dados ambientais e os enviaram ao CM 1 usando pacotes BLE. O nó CM 1 formatou essa informação em *bundles* DTN. Após cerca de uma hora sem conexão, os nós CM1 e CM2 se comunicaram com sucesso, trocando 437 *bundles* quando reconectados.

Até o momento, a literatura científica não apresenta uma implementação prática da arquitetura *FloatingBlue*, nem relata resultados de desempenho dessa arquitetura. Embora o estudo [51] ofereça a possibilidade de simular a arquitetura *FloatingBlue*, ele não conduziu ensaios específicos que simulam o uso de DTN e do BLE juntos.

A proposta *FloatingBlue* obteve sucesso devido aos serviços de software desenvolvidos na camada de aplicação do nó CM e ao sistema embarcado desenvolvido no nó EN. A Figura 29 mostra que variar o intervalo de transmissão das mensagens de *advertising* altera o consumo de energia do EN. Para aplicações que não precisam de transmissões rápidas, enviar um pacote por segundo é mais eficiente em termos energéticos. Já a Figura 30 destaca a necessidade de otimizar o tamanho da mensagem da PDU para economizar energia na transmissão de *advertising*. A Figura 32 indica que a potência de transmissão do Bluetooth afeta o alcance da comunicação. Especificamente, o alcance aumenta 370% ao comparar potências de 8 dBm com 0 dBm, em contrapartida, o consumo energético aumenta apenas 11%, como mostra a Figura 31.

Conforme indicado na Tabela 6, a arquitetura proposta destaca-se por apresentar um dos menores consumos de transmissão, devido a otimização das mensagens de *advertising*. Em modo ocioso, observa-se um consumo favorável do EN que, apesar de ser superior ou comparável aos de trabalhos que utilizam a tecnologia WuR [48,51], apresenta uma vantagem significativa no alcance. Enquanto em trabalhos com a tecnologia WuR se verifica alcance de recepção entre 5 e 25 metros, neste estudo, alcança-se até 207 metros.

Capítulo 6

Conclusões

Neste trabalho, foi desenvolvido o *FloatingBlue*, uma arquitetura para RSSF que não depende de infraestrutura fixa. Esta arquitetura combina as tecnologias de Redes Tolerantes a Atrasos e Desconexões (DTN) com Bluetooth Low Energy (BLE). Os resultados obtidos demonstram que a *FloatingBlue* otimiza a manutenção e operação de RSSF em áreas remotas e extensas, representando uma contribuição valiosa em comparação com as soluções existentes na literatura científica.

A otimização alcançada neste trabalho foi realizada através da redução do consumo energético dos ENs, utilizando a função de *broadcast* do BLE. Essa abordagem contribui para prolongar a vida útil da bateria dos sensores. Além disso, a otimização inclui a diminuição da necessidade de infraestrutura de rede, que é uma característica comum em redes de *Low-Power Wide-Area Network* (LPWAN). Para isso, implementamos nós CM que atuam como gateways móveis dos ENs. A contribuição científica deste estudo se destaca por criar uma arquitetura de rede que possui as seguintes características inovadoras: i) o uso de ENs de baixo consumo energético na borda da rede através do BLE; ii) a dispensa de infraestrutura de rede tradicional; iii) e a capacidade de tolerar atrasos e desconexões.

A fim de validar a arquitetura *FloatingBlue*, foi realizado um estudo de caso em um ambiente rural. O estudo de caso envolveu o desenvolvimento de software embarcado para os nós EN e nós CMs, que foram instalados em veículos, tanto tripulados quanto não tripulados. O principal objetivo da implementação desse estudo de caso foi estabelecer a comunicação entre os ENs e os CMs utilizando a tecnologia BLE. Os nós CM foram responsáveis por coletar dados dos ENs e, em

conjunto com a estação de processamento e controle, formar uma rede capaz de tolerar atrasos e desconexões. Esta capacidade foi viabilizada pelo uso do protocolo *Bundle Protocol* 7.

O desempenho da arquitetura *FloatingBlue* foi avaliado como satisfatório, especialmente pela capacidade de transferir centenas de pacotes dos ENs para a estação de processamento. Essa transferência de mensagens ocorreu mesmo em cenários onde havia longas desconexões entre os nós concentradores. É importante destacar que, segundo o estado da arte, não existem trabalhos que tenham implementado uma arquitetura similar à *FloatingBlue* sem o uso de simuladores. Outro ponto a ser ressaltado é que o uso do modo *broadcast* do BLE resultou em um consumo energético menor durante a transmissão em comparação com outras propostas. Quanto ao consumo energético no modo ocioso, a *FloatingBlue* apresenta resultados similares aos encontrados no estado da arte, especialmente em arquiteturas que utilizam *Wake-up Radios*. No entanto, o EN desenvolvido para a presente proposta consome um pouco mais de energia em modo ocioso, mas compensa com um alcance de comunicação superior e a vantagem de não necessitar de transceptores adicionais para um rádio secundário.

Para trabalhos futuros, propõe-se as seguintes investigações:

1. Incorporar a tecnologia de rádio LoRa na camada física do nó CM, com o objetivo de aumentar o alcance de comunicação entre os nós CMs;

2. Adotar um protocolo que permita o armazenamento e a transmissão de mensagens dos nós EN em múltiplos pacotes de *advertising*. Para otimizar este protocolo, recomenda-se a implementação de um algoritmo de compressão de mensagens;

3. Desenvolver um protocolo de segurança a nível de aplicação, visando proteger os dados transmitidos pelos nós EN;

4. Realização de experimentos com a arquitetura em ambientes florestais. Neste cenário, os ENs seriam posicionados abaixo da copa das árvores, enquanto o nó CM se deslocaria acima da copa. O objetivo seria testar a eficácia da arquitetura na coleta de dados dos ENs nestas condições específicas de ambiente florestal.

Referências Bibliográficas

[1] KANDRIS, D., NAKAS, C., VOMVAS, D., KOULOURAS, G., "Applications of wireless sensor networks: an up-to-date survey", *Applied System Innovation*, v. 3, n. 1, pp. 14, 2020.

[2] MUTEBA, F., DJOUANI, K., OLWAL, T., "A comparative survey study on LPWA IoT technologies: Design, considerations, challenges and solutions", *Procedia Computer Science*, v. 155, pp. 636–641, 2019.

[3] PIYARE, R., MURPHY, A. L., MAGNO, M., BENINI, L., "On-demand LoRa: Asynchronous TDMA for energy efficient and low latency communication in IoT", *Sensors*, v. 18, n. 11, pp. 3718, 2018.

[4] DJIDI, N. E. H., GAUTIER, M., COURTAY, A., BERDER, O., MAGNO, M., "On-demand LoRa: Asynchronous TDMA for energy efficient and low latency communication in IoT", *Sensors*, v. 21, n. 3, pp. 733, 2021.

[5] ZHANG, X., ZHANG, M., MENG, F., QIAO, Y., XU, S., HOUR, S., "A low-power wide-area network information monitoring system by combining NB-IoT and LoRa", *IEEE Internet of things Journal*, v. 6, n. 1, pp. 590–598, 2018.

[6] FU, X., LI, W., MING, H., FORTINO, G., "A framework for WSN-based opportunistic networks". In: *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 343–348, 2015.

[7] FINNEGAN, J., BROWN, S., "A comparative survey of LPWA networking", *arXiv preprint arXiv:1802.04222*, 2018.

[8] PARK, S., YUN, S., KIM, H., KWON, R., GANSER, J., "Forestry monitoring system using lora and drone". In: *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, pp. 1–8, 2018.

[9] SOLPICO, D., TAN, M., MANALANSAN, E., ZAGALA, F., LECETA, J., LANUZA, D., BERNAL, J., RAMOS, R., VILLAREAL, R., CRUZ, X., OTHERS, "Application of the V-HUB standard using LoRa beacons, mobile cloud, UAVs, and DTN for disaster-resilient communications". In: *2019 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–8, 2019.

[10] BUJARI, A., CALAFATE, C. T., CANO, J.-C., MANZONI, P., PALAZZI, C. E., RONZANI, D., "A location-aware waypoint-based routing protocol for airborne DTNs in search and rescue scenarios", *Sensors*, v. 18, n. 11, pp. 3758, 2018.

[11] BAUMGÄRTNER, L., LIESER, P., ZOBEL, J., BLOESSL, B., STEINMETZ, R., MEZINI, M., "LoRAgent: A DTN-based Location-aware Communication System using LoRa". In: *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–8, 2020.

[12] HÖCHST, J., BAUMGÄRTNER, L., KUNTKE, F., PENNING, A., STERZ, A., FREISLEBEN, B., "Lora-based device-to-device smartphone communication for crisis scenarios". In: *Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, 2020.

[13] BAUMGÄRTNER, L., PENNING, A., LAMPE, P., RICHERZHAGEN, B., STEINMETZ, R., FREISLEBEN, B., "Environmental monitoring using low-cost hardware and infrastructureless wireless communication". In: *2018 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–8, 2018.

[14] JEON, K. E., SHE, J., SOONSAWAD, P., NG, P. C., "Ble beacons for internet of things applications: Survey, challenges, and opportunities", *IEEE Internet of Things Journal*, v. 5, n. 2, pp. 811–828, 2018.

[15] GOMEZ-DE GABRIEL, J. M., FERNÁNDEZ-MADRIGAL, J. A., LOPEZ-ARQUILLOS, A., RUBIO-ROMERO, J. C., "Monitoring harness use in construction with BLE beacons", *Measurement*, v. 131, pp. 329–340, 2019.

[16] HEYDARISHAHREZA, N., EBADOLLAHI, S., VAHIDNIA, R., DIAN, F. J., "Wireless sensor networks fundamentals: A review". In: *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0001–0007, 2020.

[17] BUI, N. M., NGUYEN, Q. C., LE, M. T., OTHERS, "Indoor Localization Based on BLE Wireless Sensor Network". In: *2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, pp. 110–115, 2022.

[18] JONDHALE, S. R., SHARMA, M., MAHESWAR, R., SHUBAIR, R., SHELKE, A., "comparison of neural network training functions for rssi based indoor localization problem in WSN", *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's*, pp. 112–133, 2020.

[19] WANG, W., OTHERS, "Performance Evaluation of Deployable Bluetooth Low Energy Mesh Network for Monitoring System", 2022.

[20] DIAN, F. J., "An analytical scheme for power consumption of battery-operated peripheral BLE nodes". In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1021–1026, 2019.

[21] DE RAEVE, N., VERHAEVERT, J., VAN TORRE, P., RONSE, F., ROGIER, H., "BLE-based Power Efficient WSN for Industrial IoT Train Integrity Monitoring". In: *2022 7th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–6, 2022.

[22] SOMARATNE, K., DIAN, F. J., YOUSEFI, A., "Accuracy analysis of time synchronization using current consumption pattern of BLE devices". In: *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 841–844, 2018.

[23] DIAN, F. J., YOUSEFI, A., SOMARATNE, K., "A study in accuracy of time synchronization of BLE devices using connection-based event". In: *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 595–601, 2017.

[24] DIAN, F. J., YOUSEFI, A., LIM, S., "A practical study on Bluetooth Low Energy (BLE) throughput". In: *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 768–771, 2018.

[25] DIAN, F. J., "Low-power synchronized multi-channel data acquisition communication system". In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1027–1031, 2019.

[26] WANG, P.-H. P., MERCIER, P. P., "A dual-mode Wi-Fi/BLE wake-up receiver", *IEEE Journal of Solid-State Circuits*, v. 56, n. 4, pp. 1288–1298, 2021.

[27] GIOVANELLI, D., MILOSEVIC, B., BRUNELLI, D., FARELLA, E., "Enhancing Bluetooth Low Energy with wake-up radios for IoT applications". In: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1622–1627, 2017.

[28] MIKHAYLOV, K., KARVONEN, H., "Enhancing Bluetooth Low Energy with wake-up radios for IoT applications". In: *2020 14th International Symposium on Medical Information Communication Technology (ISMICT)*, pp. 1–5, 2020.

[29] SULTANIA, A. K., DELGADO, C., FAMAEY, J., "Enabling low-latency Bluetooth low energy on energy harvesting batteryless devices using wake-up radios", *Sensors*, v. 20, n. 18, pp. 5196, 2020.

[30] PAOLINI, G., MURILLO, Y., CLAESSENS, S., MASOTTI, D., POLLIN, S., COSTANZO, A., SCHREURS, D., "Rf energy harvesting from gfsk-modulated ble signals". In: *2021 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNeT)*, pp. 27–29, 2021.

[31] LUO, B., XIANG, F., SUN, Z., YAO, Y., "BLE neighbor discovery parameter configuration for IoT applications", *IEEE Access*, v. 7, pp. 54097–54105, 2019.

[32] CHEN, B.-R., CHENG, S.-M., LIN, J.-J., "Energy-efficient BLE device discovery for Internet of Things". In: *2017 Fifth International Symposium on Computing and Networking (CANDAR)*, pp. 75–79, 2017.

[33] SPACHOS, P., PLATANIOTIS, K. N., "BLE beacons for indoor positioning at an interactive IoT-based smart museum", *IEEE Systems Journal*, v. 14, n. 3, pp. 3483–3493, 2020.

[34] SOUBELET, A., PATHINARUPOTHI, R. K., RANGAN, E. S., DURGA, P., MENON, K. U., "Internet-of-things based respiratory rate monitoring for early detection of cardiovascular and pulmonary diseases". In: *5th EAI International Conference on IoT Technologies for HealthCare*, pp. 97–106, 2020.

[35] ALFIAN, G., SYAFRUDIN, M., IJAZ, M. F., SYAEKHONI, M. A., FITRIYANI, N. L., RHEE, J., "A personalized healthcare monitoring system for diabetic patients by utilizing BLE-based sensors and real-time data processing", *Sensors*, v. 18, n. 7, pp. 2183, 2018.

[36] CERF, V., BURLEIGH, S., HOOKE, A., TORGERSON, L., DURST, R., SCOTT, K., FALL, K., WEISS, H., *Delay-tolerant networking architecture*, Tech. rep., 2007.

[37] SUGIURA, S., YAMADA, Y., YOSHIZAKI, T., NAITOT, K., "Proposal of service framework for information sharing based on delay tolerant networks". In: *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–2, 2020.

[38] MALLORQUÍ, A., ZABALLOS, A., SERRA, D., "The Antarctic Delay Tolerant Network". In: *2022 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–7, 2022.

[39] KULLA, E., SHINTANI, K., MATSUO, K., "Mobility-Aware Narrow Routing Protocol for Underwater Wireless Sensor Networks". In: *Advances in Internet, Data & Web Technologies: The 10th International Conference on Emerging Internet, Data and Web Technologies (EIDWT-2022)*, pp. 245–253, 2022.

[40] OCHIAI, H., ISHIZUKA, H., KAWAKAMI, Y., ESAKI, H., "A DTN-based sensor data gathering for agricultural applications", *IEEE Sensors Journal*, v. 11, n. 11, pp. 2861–2868, 2011.

[41] FALL, K., "A delay-tolerant network architecture for challenged internets". In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27–34, 2003.

[42] LIU, X., LI, B., CHEN, M., WANG, X., ZOU, Q., "Research on DTN Based Interconnecting ZigBee Network Techniques". In: *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, 2011.

[43] BANERJEE, N., CORNER, M. D., LEVINE, B. N., "An energy-efficient architecture for DTN throwboxes". In: *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pp. 776–784, 2007.

[44] RATHJE, P., LANDSIEDEL, O., "DisruptaBLE: Opportunistic BLE Networking". In: *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, pp. 165–172, 2022.

[45] BAKER, C., ALMODOVAR-FARIA, J., JUSTE, P. S., MCNAIR, J., "Low energy socially cognizant routing for delay tolerant mobile networks". In: *MILCOM 2013-2013 IEEE Military Communications Conference*, pp. 299–304, 2013.

[46] CHEN, J., DAI, Z., CHEN, Z., "Development of radio-frequency sensor wake-up with unmanned aerial vehicles as an aerial gateway", *Sensors*, v. 19, n. 5, pp. 1047, 2019.

[47] SHESHASHAYEE, A. V., BUCZEK, J., PETRIOLI, C., BASAGNI, S., "Experimental Evaluation of Wake-up Radio Ranges for UAV-assisted Mobile Data Collection". In: *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 716–721, 2022.

[48] TROTTA, A., DI FELICE, M., BONONI, L., PERILLI, L., SCARSELLI, E. F., CINOTTI, T. S., "Throughput Enhancement in UAV-aided Wireless Sensor Networks via Wake-Up Radio Technology and Priority-based MAC Scheme". In: *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pp. 1–6, 2020.

[49] KOMAROV, M., MOLTCHANOV, D., "System design and analysis of UAV-assisted BLE Wireless Sensor Systems". In: *Wired/Wireless Internet Communications: 14th IFIP WG 6.2 International Conference, WWIC 2016, Thessaloniki, Greece, May 25-27, 2016, Proceedings 14* , pp. 284–296, 2016.

[50] RAJAKARUNA, A., MANZOOR, A., PORAMBAGE, P., LIYANAGE, M., YLIANTTILA, M., GURTOV, A., "Enabling end-to-end secure connectivity for low-power iot devices with uavs". In: *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, pp. 1–6, 2019.

[51] TROTTA, A., DI FELICE, M., PERILLI, L., SCARSELLI, E. F., CINOTTI, T. S., "BEE-DRONES: Ultra low-power monitoring systems based on unmanned aerial vehicles and wake-up radio ground sensors", *Computer Networks*, v. 180, pp. 107425, 2020.

[52] RAHMADHANI, M. A., YOVITA, L. V., MAYASARI, R., "Energy Consumption and Packet Loss Analysis of LEACH Routing Protocol on WSN Over DTN". In: *2018 4th International Conference on Wireless and Telematics (ICWT)*, pp. 1–5, 2018.

[53] MARCHESE, M., MOHEDDINE, A., PATRONE, F., "Towards increasing the LoRa network coverage: A flying gateway". In: *2019 International Sympo-*

sium on Advanced Electrical and Communication Technologies (ISAECT), pp. 1–4, 2019.

[54] ROCHOL, J., *Sistemas de comunicação sem fio: conceitos e aplicações*. Bookman Editora, 2018.

[55] HANES, D., SALGUEIRO, G., GROSSETETE, P., BARTON, R., HENRY, J., *IoT fundamentals: Networking technologies, protocols, and use cases for the internet of things*. Cisco Press, 2017.

[56] VASILAKOS, A., ZHANG, Y., SPYROPOULOS, T., *Delay Tolerant Networks*. CRC Press, 2016.

[57] CERF, V., BURLEIGH, S., HOOKE, A., TORGERSON, L., DURST, R., SCOTT, K., FALL, K., WEISS, H., "RFC 4838: Delay-tolerant networking architecture", 2007.

[58] WARTHMAN, F., OTHERS, "Delay-and disruption-tolerant networks (DTNs)", *A Tutorial. V.. 0, Interplanetary Internet Special Interest Group*, pp. 5–9, 2012.

[59] SCOTT, K., "Request for comments 5050: Bundle Protocol Specification", *http://www. ietf. org/rfc/rfc5050. txt*, 2007.

[60] BURLEIGH, S., FALL, K., BIRRANE, E. J., "Bundle Protocol Version 7", RFC 9171, Jan. 2022.

[61] PENNING, A., BAUMGÄRTNER, L., HÖCHST, J., STERZ, A., MEZINI, M., FREISLEBEN, B., "Dtn7: An open-source disruption-tolerant networking implementation of bundle protocol 7". In: *Ad-Hoc, Mobile, and Wireless Networks: 18th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2019, Luxembourg, Luxembourg, October 1–3, 2019, Proceedings 18*, pp. 196–209, 2019.

[62] BHATTACHARJEE, S., ROY, S., BANDYOPADHYAY, S., "Exploring an energy-efficient DTN framework supporting disaster management services in post disaster relief operation", *Wireless Networks*, v. 21, pp. 1033–1046, 2015.

[63] SOMMER, M., HÖCHST, J., STERZ, A., PENNING, A., FREISLEBEN, B., "ProgDTN: Programmable Disruption-tolerant Networking". In: *International Conference on Networked Systems (NETYS)*, May 2022.

[64] *Module python subprocess*, 2023.

[65] SPACHOS, P., PLATANIOTIS, K., "BLE beacons in the smart city: Applications, challenges, and research opportunities", *IEEE Internet of Things Magazine*, v. 3, n. 1, pp. 14–18, 2020.

[66] TOWNSEND, K., CUFÍ, C., DAVIDSON, R., OTHERS, *Getting started with Bluetooth low energy: tools and techniques for low-power networking*. "O'Reilly Media, Inc.", 2014.

[67] Bluetooth SIG, *Bluetooth Specification Version 4.2*, 2014.

[68] HEYDON, R., *Bluetooth Low Energy: The Developer's Handbook*. *Pearson Always Learning*, Prentice Hall, 2012.

[69] HARVEY, I., "bluepy: Python interface to Bluetooth LE on Linux", 2021.

[70] Bluetooth SIG, *Assigned Numbers*, 2014.

[71] Zephyr Project, *Zephyr RTOS*, 2023.

[72] Nordic Semiconductor, *nRF5340 SoC Product Brie*, 2020, V 2.0.

[73] Nordic Semiconductor, *S132 SoftDevice*, 2019, V 7.1.

[74] Nordic Semiconductor, *nRF Connect Bluetooth Low Energy*, 2022, V 4.0.

[75] Bosch Sensortec, *BME680 Datasheet*, 2022, V 1.8.