MONITORAMENTO DE AMBIENTE ATRAVÉS DE UMA REDE OPORTUNÍSTICA CENTRADA EM INTERESSE

Lucas Severiano dos Santos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Claudio Luis de Amorim

Rio de Janeiro
Fevereiro de 2020

MONITORAMENTO DE AMBIENTE ATRAVÉS DE UMA REDE OPORTUNÍSTICA CENTRADA EM INTERESSE

Lucas Severiano dos Santos

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Claudio Luis de Amorim

Aprovada por: Prof. Claudio Luis de Amorim
Prof. Felipe Maia Galvão França
Profa. Maria Clicia Stelling de Castro
Prof. Raphael Carlos Santos Machado

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2020

Santos, Lucas Severiano dos

Monitoramento de ambiente através de uma rede oportunística centrada em interesse/Lucas Severiano dos Santos. – Rio de Janeiro: UFRJ/COPPE, 2020.

XII, 70 p.: il.; 29, 7cm.

Orientador: Claudio Luis de Amorim

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2020.

Referências Bibliográficas: p. 58 – 65.

1. Redes Oportunísticas. 2. Redes Centradas em Interesse. 3. Redes Centradas no Conteúdo. 4. Internet das Coisas. 5. Monitoramento de ambiente. I. Amorim, Claudio Luis de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus pais

Agradecimentos

Em primeiro lugar agradeço a Deus, Criador do céu e da terra e Senhor de todo conhecimento, por ter me guardado e sustentado durante toda minha jornada acadêmica, me proporcionando perseverança, força e paz nos momentos mais difíceis e obscuros.

Aos meus pais Edilson e Lindaura, por terem acreditado no meu potencial, por cada palavra de motivação e incentivo, por todo investimento e por cada pérola de sabedoria investida em mim.

Ao meu orientador, o Professor Claudio Luis de Amorim, por ter acredito no meu trabalho, por toda paciência, orientação acadêmica e ajuda na elaboração desta dissertação.

Ao Inmetro e ao projeto SHCDiber por ter proporcionado o recurso financeiro do meu projeto.

Ao meu amigo, Paulo Roberto, por toda discussão técnica, ideias, submissões de artigos e projetos, ajuda no texto e por todo apoio durante a realização deste trabalho.

Ao Raphael Machado pela confiança depositada em mim e por, principalmente, me motivar e incentivar a continuar com o desenvolvimento do meu projeto.

Ao Jaci Júnior e ao Carlos Augusto pela ajuda na execução dos testes em campo e por terem sido os melhores copilotos de drone.

À Lucila Bento pelo apoio, orientação, instruções para participação do evento SBSeg e pela ajuda na revisão do texto.

Ao Fernando Alves Rodrigues, do laboratório de telecomunicações, pela ajuda nas análises dos espectros Wi-Fi e plotagem dos gráficos.

Aos meus colegas de trabalho Luiz Tarelho, Guilherme Garcia, João Alfredo, Simony Monteiro e Ewerton Madruga pelas conversas descontraídas durante o horário de almoço.

Ao meu colega Michael Douglas pelas diversas vezes que sanou as minhas dúvidas sobre o protocolo RadNet.

A minha família e amigos, pela compreensão durante os diversos momentos em que estive ausente devido minha dedicação acadêmica.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MONITORAMENTO DE AMBIENTE ATRAVÉS DE UMA REDE OPORTUNÍSTICA CENTRADA EM INTERESSE

Lucas Severiano dos Santos

Fevereiro/2020

Orientador: Claudio Luis de Amorim

Programa: Engenharia de Sistemas e Computação

A Internet das Coisas vem tornando a Internet cada vez mais presente no cotidiano das pessoas ao permitir a integração de uma variedade de objetos físicos e dispositivos na rede. Este novo paradigma pode ser encontrado nos mais diversos campos, desde da implementação de sistemas de segurança física até o monitoramento de incêndios florestais. Porém, quando a rede de comunicação não está disponível — o que é frequentemente o caso de aplicativos de sensoriamento de lugares amplos e remotos — ter acesso às informações nos sensores pode ser um desafio. Esta dissertação apresenta o desenvolvimento e avaliação experimental de uma aplicação de sensoriamento remoto a qual é baseada nos conceitos de redes oportunísticas e redes centradas em interesse, como mecanismo de transmissão de dados em áreas remotas. A aplicação de monitoramento foi implementada sobre o protocolo RadNet, um protocolo de comunicação que implementa o conceito de redes centrada em interesse. Um agente mensageiro móvel foi implantado utilizando um drone, modelo Phaton 4 Pro+, para prover conectividade entre os dispositivos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ENVIRONMENT MONITORING THROUGH AN INTEREST-CENTRIC OPPORTUNISTIC NETWORK

Lucas Severiano dos Santos

February/2020

Advisor: Claudio Luis de Amorim

Department: Systems Engineering and Computer Science

The Internet of Things has been making the Internet even more present in people's routine by allowing the integration of a variety of physical objects and devices in the network. This new paradigm can be found in most diverse fields, ranging from physical security systems to forest fire monitoring. However, when the communication network is not available — which is frequently the case for remote-sites wide-area sensing applications — having access to information on sensors can be a challenge. This work presents the development and experimental evaluation of a remote sensing application, based on the concepts of opportunistic networks and interest-centric networks, as data transmission mechanism in remote area. The monitoring application was implemented and built over the RadNet protocol, a communication protocol that implements the concept of interest-centric networking. A mobile messenger was deployed on a drone, model Phantom 4 Pro+, to provide connectivity between devices.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A proliferação de objetos físicos interconectados vem crescendo em função do avanço de diversas áreas como Sistemas Embarcados, Microeletrônica, Comunicação e Sensoriamento [Al-Fuqaha et al. 2015] [Santos et al. 2016] [Atzori et al. 2010]. Neste sentido, Internet das Coisas (*Internet of Things* – IoT) emerge como solução para interconectar tais objetos à Internet e promover comunicação entre dispositivos e pessoas. Assim, IoT é um paradigma da Tecnologia da Informação e Comunicação (TIC) no qual um grande número de dispositivos, como Medidores Inteligentes, sensores e atuadores são interconectados em rede de modo a cooperar uns com os outros com a finalidade de alcançar um objetivo em comum [Atzori et al. 2010].

O conceito de Internet das Coisas torna a Internet ainda mais presente no cotidiano das pessoas ao permitir acesso e interação a uma variedade de dispositivos, os quais geram dados abertos para consumo por terceiros [Zanella et al. 2014]. Este paradigma pode ser empregado em diversos ambientes e aplicações como automação residencial [Mandula et al. 2015] [Piyare 2013], assistência médica [Gia et al. 2015] [Mohapatra and Rekha 2012], sistemas de transporte [Zhou et al. 2012] e resposta de emergência a catástrofes [Bellavista et al. 2013]. Notavelmente, o paradigma de Internet das Coisas oferece um extraordinário potencial para alavancar novos serviços de TIC, porém introduz novos desafios aos desenvolvedores de sistemas IoT como os descritos a seguir [Amadeo et al. 2016] [Li et al. 2014]:

- **Escalabilidade:** Com o avanço das tecnologias móveis foi estimado que em 2020 o número de dispositivos conectados à Internet será de 30 a 50 bilhões [Statista 2019] [Swan 2012] , chegando ao valor de 75 bilhões em 2025 [Statista 2019]. Assim, as plataformas desenvolvidas para Internet das Coisas devem ser escaláveis no que diz respeito a métricas como tempo de resposta e tráfego de dados;

- **Mobilidade:** Os dispositivos no contexto de Internet das Coisas podem se locomover embarcados em veículos ou serem transportados por pessoas. Neste

sentido, soluções para aplicações de Internet das Coisas devem levar em consideração o suporte a mobilidade considerando uma variedade de dispositivos;

- **Heterogeneidade:** Sistemas de Internet das Coisas são implementados em ambientes altamente heterogêneos com uma variedade de dispositivos, tecnologias e serviços envolvendo diferentes *stakeholders*[1] e fabricantes. Neste sentido, fabricantes fornecem dispositivos com suas próprias tecnologias e serviços que podem não ser acessados por outros equipamentos ou pessoas por conta da incompatibilidade tecnológica. Sendo assim, torna-se necessária a busca por uma padronização a fim de fornecer melhor interoperabilidade entre os dispositivos e serviços[Khan et al. 2012];

- **Eficiência energética:** Dispositivos de Internet das Coisas são tipicamente limitados no tocante aos recursos de energia, capacidade computacional e funcionalidades de comunicação. Portanto, é imprescindível que as arquiteturas projetadas para sistemas de Internet das Coisas sejam adequadas às limitações destes dispositivos, proporcionando economia de recursos de modo a aumentar o tempo de operação destes equipamentos;

- **Segurança:** A Internet das Coisas apresenta, também, novas ameaças e desafios relacionados à privacidade, confiabilidade, confidencialidade, integridade e disponibilidade, visto que este cenário é composto por diversos dispositivos com capacidade de sensoriamento de ambiente e controle de objetos físicos [Al-Fuqaha et al. 2015]. Portanto, é necessário que os dispositivos presentes na rede atendam a uma série de requisitos de segurança especificados de acordo com o risco da aplicação.

Além das deficiências expostas, Internet das Coisas introduz novos desafios em relação às tecnologias de comunicação, visto que, o desenvolvimento e implantação em larga escala de arquiteturas de IoT são baseadas em IP. As limitações do endereçamento baseado no protocolo de comunicação IP, que funciona tanto como localizador quanto como identificador da informação, e a necessidade de um sistema de resolução, suporte de mobilidade complexo e acesso massivo, tornam a pilha de protocolos TCP/IP ineficiente diante do cenário de Internet das Coisas [Amadeo et al. 2016] [Arshad et al. 2018].

[1]Stakeholder é o termo utilizado para identificar as parte interessadas num determinado projeto de *software*.

1.1 Motivação

Embora o protocolo de IP seja amplamente utilizado para comunicação na Internet, existem outros protocolos e arquiteturas que não se baseiam em endereçamento para prover conectividade entre dispositivos. Uma das arquiteturas deste tipo mais estudadas atualmente são as Redes Centradas em Interesse onde o paradigma tradicional de conexão fim-a-fim é substituído pelo roteamento baseado no conteúdo que se deseja disponibilizar e no interesse neste conteúdo [Dutra et al. 2012] [Gonçalves et al. 2016]. Nestes modelos de comunicação, a rede é capaz de se adaptar a muitos dispositivos com alta heterogeneidade, fornecendo rápido acesso e distribuição, segurança e escalabilidade. Outro conceito relevante em redes de comunicação são as Redes Oportunísticas (OppNet) [Trifunovic et al. 2017]. Uma OppNet é um tipo de rede *ad-hoc* capaz de realizar o encaminhamento de informações no ambiente de rede conectado de maneira intermitente, através da oportunidade de contato trazida pelo movimento mútuo entre os nós. Estas redes são capazes de fornecer comunicação entre dispositivos em ambientes onde não há infraestrutura de rede previamente estabelecida entre os dispositivos [Trifunovic et al. 2017].

Diante deste cenário, este trabalho apresenta o desenvolvimento do Sistema Hermes, uma aplicação de monitoramento de ambiente baseada no paradigma de rede oportunísticas (OppNet) e que utiliza o protocolo RadNet [Dutra et al. 2012] como mecanismo de transmissão de dados. O protocolo RadNet é um protocolo de rede oportunística cujo foco da comunicação entre os nós está no interesse do conteúdo da informação e não no dispositivo no qual a informação está armazenada. O Sistema Hermes é baseado no conceito de implementação de três agentes: o Coletor, o Mensageiro e o Servidor. O agente Coletor executa nos dispositivos que possuem os sensores e são responsáveis por armazenar os dados coletados e transmiti-los ao Mensageiro. O Mensageiro, por sua vez, é responsável por capturar os dados transmitidos pelos coletores e levá-los até o servidor, no qual os dados serão armazenados em definitivo. Embora o agente Mensageiro possa ser implementado junto a qualquer dispositivo com capacidade de mobilidade, neste trabalho é considerado o uso de um veículo aéreo não tripulado (Drone) que possui a finalidade de transportar os dados gerados pelos agentes Coletores. Por fim, o agente Servidor é responsável por armazenar os dados recebidos pelo agente Mensageiro e disponibilizá-los para outros serviços e aplicações disponíveis na rede.

Para validação do sistema proposto neste trabalho, é abordado o problema de sensoriamento remoto num cenário onde os sensores estão dispersos geograficamente e com um ponto de concentração para armazenamento de dados. O estudo considera um cenário real localizado no campus do Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro). Os laboratórios, situados no Instituto, manipulam

materiais de referência[1], sendo assim, é imprescindível o sensoriamento do ambiente visto que há exigência que sejam mantidas determinadas condições ambientais dentro das instalações. Os laboratórios estão situados em diversos prédios dispostos numa área com mais de 100 mil m^2, sendo a distância entre os prédios não inferior a 100 metros. Cada laboratório possui um dispositivo com um conjunto de sensores (umidade, temperatura, fumaça e movimento) fixado em um ponto e os dados coletados por estes sensores precisam ser transportados para um servidor central que é responsável pelo armazenamento e análise dos dados. Portanto, o Sistema Hermes, além de proporcionar recursos para o monitoramento do ambiente, se apresenta como uma medida alternativa de transmissão de dados de sensoriamento diante da indisponibilidade da rede local no campus.

1.2 Contribuições

Como principais contribuições destacamos:

- Desenvolvimento de uma aplicação de sensoriamento remoto através de uma rede oportunística centrada em interesse;
- Implantação do sistema de monitoramento de ambiente num cenário real;
- Avaliação experimental da aplicação em cenário onde a infraestrutura de rede é escassa ou inexistente;
- Documentação do processo de desenvolvimento do sistema de sensoriamento;
- Avaliação experimental do protocolo RadNet num cenário de redes oportunísticas e que utiliza um veículo aéreo não tripulado como mecanismo de transmissão.

1.3 Objetivo

Este trabalho tem por objetivo desenvolver e implantar um sistema de sensoriamento remoto de ambiente que utiliza uma Rede Oportunística Centrada em Interesse como mecanismo de transmissão de dados. Esta aplicação irá coletar grandezas físicas como temperatura, umidade e gases e, através de um veículo aéreo não tripulado, transportar estas informações até um servidor responsável pelo armazenamento e disponibilização das informações para outros serviços. Tal sistema corresponde a

[1]Segundo o ABNT ISO Guia 30:2016 [ISO 2016], material de referência é um material suficientemente homogêneo e estável com respeito a uma ou mais propriedades especificadas, que foi estabelecido como sendo adequado para o seu uso pretendido em um processo de medição.

uma solução alternativa ao sistema de monitoramento de ambiente local utilizado no campus do Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro) o qual utiliza a infraestrutura de redes local.

Como objetivos específicos, pretende-se: **(i)** Desenvolver o sistema de sensoriamento remoto, **(ii)** Implantar o sistema desenvolvido no campus do Inmetro. **(iii)** avaliar experimentalmente a aplicação de monitoramento considerando o cenário proposto, e **(iv)** documentar o processo de desenvolvimento e implantação do sistema.

1.4 Trabalhos relacionados

Sistemas de monitoramento ambiental, em geral, controlam e monitoram grandezas físicas como temperatura, umidade, luz e pressão. Um número considerável de estudos vêm sendo conduzidos a fim de solucionar problemas referentes ao monitoramento ambiental [Othman and Shazali 2012]. Em [Hudhajanto et al. 2018] foi implementada uma Rede de Sensores sem Fio para monitoramento ambiental em tempo real. Cada nó sensor é composto por sensores de temperatura, umidade e CO2 conectados a um arduino. Na topologia proposta, os nós sensores enviam os dados, provenientes dos sensores, para um *gateway* através do padrão IEEE 802.15.4 (Xbee). O *gateway*, por sua vez, consiste de um Rasperry o qual é responsável por sincronizar os dados coletados e armazenados (numa base de dados MySQL) com um serviço em nuvem através da pilha de protocolos TCP/IP.

O trabalho desenvolvido em [Shkurti et al. 2017] apresenta um sistema de monitoramento baseado em Web utilizando tecnologia de Redes de Sensores sem Fio. Os dispositivos, acoplados com sensores, coletam dados e os enviam, através de um roteador Wi-Fi, para uma aplicação em nuvem onde os dados serão armazenados numa base de dados. Este sistema permite que o usuário monitore os dados coletados por intermédio de uma aplicação Web. Além disso, o sistema Web notifica o usuário, através do envio de mensagens (*e-mails*), sobre as alterações nas condições climáticas, caso esteja fora do intervalo previamente configurado.

Em [Del Campo et al. 2016] foi realizada uma análise sobre a aplicabilidade do protocolo *Message Queuing Telemetry transport* (MQTT) [mqt 2019] em ambientes de vida assistida (*Ambient-Assisted Living* – AAL) [Montanini et al. 2016]. Para tal, foi selecionado um cenário específico, baseado em tecnologias assistivas, projetado para monitorar pessoas com demência em ambiente domiciliar ou hospitalar. Ainda, são apresentadas três arquiteturas de comunicação implementando o MQTT em diferentes níveis: (1) No Servidor; (2) No *gateway*; (3) Por fim, nos dispositivos com sensores. Os resultados obtidos, e a estimativa de consumo de energia fornecida para cada caso, mostram que o MQTT foi efetivamente adotado para

um distribuição rápida e confiável de mensagens de notificação entre os diferentes agentes envolvidos na plataforma.

Os trabalhos apresentaram resultados promissores no tocante ao desenvolvimento de sistemas de Internet das Coisas. Contudo, tais soluções ainda são baseadas na pilha de protocolos TCP/IP. Portanto, outros trabalhos apresentam soluções para IoT que não se baseiam no endereçamento IP. Em [Abidy et al. 2014] foi implementado e testado uma solução que utiliza o conceito Rede Centrada em Conteúdo (*Content-Centric networking* – CCN). Este conceito é semelhante às Redes Centradas em interesse, porém, contêm diferenças em suas arquiteturas. Neste trabalho, o CCN foi implementado na camada de comunicação do Contiki[2]. O trabalho foi avaliado por intermédio do recurso de simulação e de nós físicos. Os resultados demonstraram pouco *overhead* e bom desempenho em termos de troca de mensagens e atrasos na recepção dos dados. Em [Saadallah et al. 2012] são apresentados os detalhes da implementação.

Os autores em [Rodriguez et al. 2017] desenvolveram um sistema de sensoriamento para monitorar as condições das colônias de abelhas, denominado *MyBee*. Este sistema utiliza o conceito de Redes Centradas em Interesses, através do protocolo de comunicação RadNet, para implementar as funcionalidades de transmissão de dados. Os resultados do estudo de caso mostram que o *MyBee* é uma solução eficiente para apoio aos apicultores na manutenção dos apiários.

1.5 Organização da Dissertação

Esta dissertação de mestrado está divida em cinco capítulos, onde o Capítulo 1 apresenta a introdução deste trabalho, descrevendo de forma geral o problema abordado, ressaltando a sua motivação, as contribuições, trabalhos relacionados e os objetivos.

O Capítulo 2 apresenta o referencial teórico deste trabalho. Descrições sobre Internet das Coisas, Redes Centradas no Conteúdo, Redes Oportunísticas e Redes Centradas em Interesse são discutidas neste capítulo.

No Capítulo 3 é apresentado o desenvolvimento da aplicação de monitoramento onde foram implementados os conceitos apresentados no capítulo anterior. Ainda são apresentados os dispositivos, ferramentas e equipamentos utilizados para o desenvolvimento do sistema.

O Capítulo 4 apresenta uma análise experimental sobre o sistema proposto, apresentando e discutindo os resultados obtidos. Por fim, o Capítulo 5 apresenta as considerações e conclusões deste trabalho.

[2]O Contiki é um sistema operacional para sistemas em rede e com restrição de memória, com foco em dispositivos sem fio da Internet das Coisas

Capítulo 2

Fundamentação teórica

Neste capítulo é discutida a fundamentação teórica deste trabalho por meio de cinco seções. A Seção 2.1 apresenta conceitos fundamentais sobre Internet das Coisas. A Seção 2.2 descreve os conceitos sobre Redes Centradas no Conteúdo. A Seção 2.3 introduz conceitos referentes às Redes Oportunísticas. A Seção 2.4 apresenta o protocolo oportunístico centrado em interesse, denominado RadNet. Por fim, a Seção 2.5 apresenta as considerações sobre o protocolo RadNet.

2.1 Internet das Coisas

O termo "Internet das Coisas"(*Internet of Things* – IoT) foi empregado pela primeira vez em 1999 pelo pesquisador britânico, Kevin Ashton, do Instituto de Tecnologia de Massachusetts. Para Ashton, este novo paradigma tinha o mesmo potencial de disruptura que a Internet [Ashton et al. 2009]. Desde então, Internet das Coisas vem se tornando foco de vários esforços do setor industrial e de instituições acadêmicas, incluindo a União Internacional de Telecomunicaçoes (UIT). O significado de IoT vem sendo descrito de diversas formas por diferentes autores. A seguir, são apresentados algumas definições encontradas na literatura:

- Para [Al-Fuqaha et al. 2015], Internet das Coisas permite aos objetos físicos coletar dados de ambiente, compartilhar informações e coordenar decisões. Portanto, objetos tradicionais são convertidos em objetos inteligentes através de tecnologias como computação ubíqua e pervasiva, dispositivos embarcados, tecnologias de comunicação, rede de sensores, protocolos de Internet e aplicações;
- Segundo [Bassi and Horn 2008], a origem semântica da expressão "Internet das Coisas"é apresentada como uma rede mundial de objetos interconectados e endereçáveis, baseada em protocolos de comunicação padrão;

- De acordo com [Vermesan et al. 2011], IoT é um paradigma que permite pessoas e objetos estarem conectados em qualquer momento e em qualquer lugar usando diferentes redes, tecnologias e serviços disponíveis.

2.1.1 Elementos da Internet das Coisas

Um sistema de Internet das Coisas pode ser definido, também, como a combinação de diversas tecnologias, as quais são complementares, para prover integração dos objetos do ambiente físico com o virtual. Este sistema é composto por blocos básicos, como exibido na Figura 2.1, sendo eles [Santos et al. 2016]:

- **Identificação:** Este é um bloco primordial, visto a necessidade de conectar os objetos à Internet através de uma identificação única. Para este fim, são utilizadas tecnologias como RFID, NFC (*Near Filed Communication*), e endereçamento IP para identificar unicamente os objetos;

- **Sensores/Atuadores:** Sensores são responsáveis por coletar os dados de ambiente onde os objetos estão inseridos. Os sensores são acoplados a dispositivos, como Raspberry Pi e arduino, que podem armazenar/encaminhar os dados coletados para centros de armazenamento e/ou serviços em nuvem, visto que estes possuem maior capacidade computacional. Enquanto os atuadores são dispositivos que produzem alguma ação, atendendo a comandos que podem ser manuais, elétricos ou mecânicos;

- **Comunicação**: Este bloco é responsável por realizar a comunicação entre os dispositivos e servidores remotos. Assim sendo, tecnologias de comunicação para IoT são empregadas para conectar objetos heterogêneos com a finalidade de fornecer serviços para outras aplicações;

- **Serviços**: A Internet das Coisas pode prover diversas classes de serviços como: Serviços de Monitoramento, que é responsável por mapear as grandezas físicas em grandezas virtuais (temperatura de um local físico em seu valor digital, por exemplo); Serviços de Agregação de Dados, que reúne os dados coletados pelos dispositivos/objetos inteligentes; Serviços de Colaboração e Inteligência, que são responsáveis por tomar decisões com base nos dados coletados; e Serviço de Ubiquidade, que visam prover serviços de colaboração e inteligência em qualquer momento e lugar;

- **Semântica**: Refere-se a capacidade de extração de conhecimento a partir dos objetos inteligentes distribuídos na rede. Em outras palavras, trata-se do uso eficiente dos recursos existentes na rede para prover determinado serviço;

- **Computação**: Se refere a capacidade de processamento embarcada no dispositivo de Internet das Coisas como, por exemplo, Raspberry Pi e FPGAs que possuem capacidade de executar algoritmos locais nos dispositivos inteligentes.

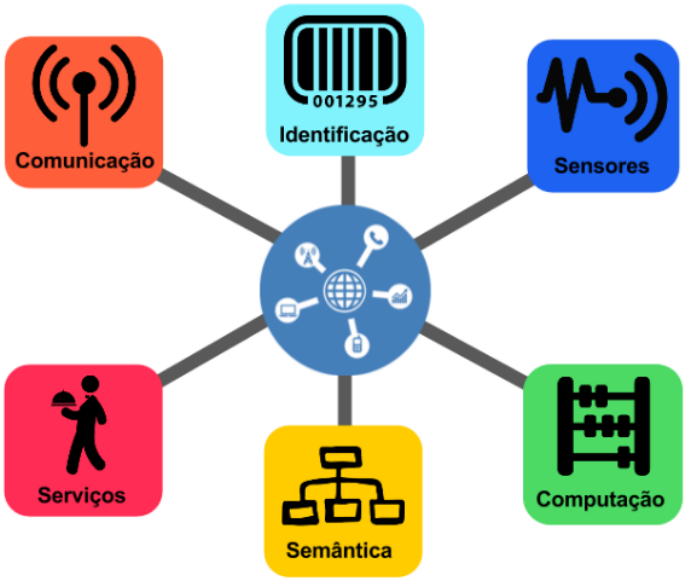

Figura 2.1: Blocos básicos IoT [Santos et al. 2016].

Um sistema de Internet das Coisas é composto por dispositivos que fornecem capacidade de sensoriamento, atuação, controle e monitoramento [Ray 2018]. Estes dispositivos incluem unidades de processamento/memória, comunicação, energia e sensores/atuadores. A Figura 2.2 apresenta os elementos que compõem a arquitetura dos dispositivos de Internet das Coisas. Nesta arquitetura a unidade **(iii) Fonte de energia** é primordial para funcionamento do dispositivo, visto que esta é responsável por fornecer energia aos componentes embarcados no dispositivo. Esta fonte de energia pode ser composta por uma bateria, que é mais indicado para objetos móveis, e um conversor AC-DC. A energia elétrica também pode ser utilizada como fonte de alimentação, sendo mais indicada para dispositivos estacionários [Santos et al. 2016].

2.1.2 Aplicações de Internet das Coisas

Como já citado, Internet das Coisas engloba uma diversidade de áreas de aplicações que inclui monitoramento de riscos à saúde das pessoas, transporte público, automação industrial e resposta de emergência a catástrofes [Al-Fuqaha et al. 2015]. No âmbito da saúde, tecnologias de Internet das Coisas desempenham um papel importante nas aplicações de assistência à saúde através da integração de sensores e

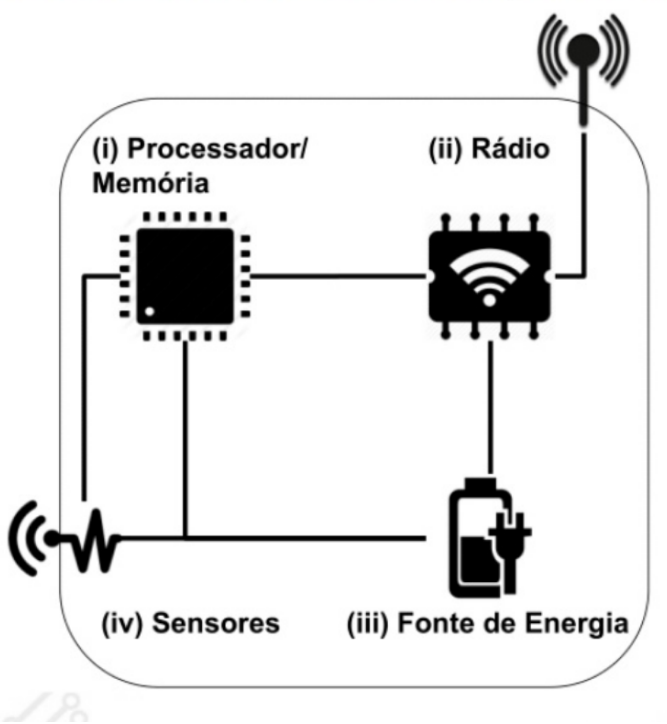

Figura 2.2: Arquitetura dos dispositivos IoT [Santos et al. 2016].

atuadores que permitem monitorar e analisar remotamente o estado fisiológico dos pacientes [Perera et al. 2014]. Em [Hande et al. 2006] é apresentado um sistema que permite monitorar sinais como eletrocardiograma, oximetria e pressão arterial sem a necessidade de ter um médico fisicamente presente para realizar as medições.

No tocante ao transporte público, carros, táxis, ônibus e trens, assim como suas respectivas vias, são equipados com sensores, atuadores e capacidade de processamento com a finalidade de fornecer informações relevantes para motoristas e/ou passageiros. Estas informações oferecem melhores condições de condução e segurança através de sistemas de prevenção de colisão e monitoramento de transporte de materiais perigosos, por exemplo [Al-Fuqaha et al. 2015]. Em [Li et al. 2006] é apresentada a aplicação *Traffic Information Grid (TIG)*, a qual proporciona às pessoas informações e orientações sobre o trânsito. Esta aplicação integra informações de tráfego coletadas dos sensores e compartilha dados em relação ao tráfego com a finalidade de fornecer melhores serviços aos participantes removendo e/ou reduzindo o congestionamento da via.

No que se refere à resposta de emergência a catástrofes, Internet das Coisas oferece medidas fundamentais em casos de desastres naturais, como inundações, deslizamentos e incêndios, através da análise sobre a evolução de uma catástrofe que permite ao governo e às equipes de emergência oferecerem serviços de resgate mais adequados e eficientes [Castillo-Effer et al. 2004].

2.2 Redes Centradas no Conteúdo

Os princípios de engenharia e arquitetura da Internet foram criados com a finalidade de solucionar o problema do compartilhamento de recursos. Portanto, o objetivo era estabelecer uma comunicação eficiente entre duas estações. Para tal, foi estabelecido um mecanismo de comunicação que utiliza endereçamento para identificar as estações de origem e de destino na comunicação [Jacobson et al. 2009] [Xylomenos et al. 2013]. Entretanto, a distribuição de recursos na Internet passou por um processo de evolução ao longo dos anos, afastando-se de um sistema de informação textual para um sistema de informação multimídia onde dados, serviços e aplicações são consumidos como conteúdo. Neste contexto, os conteúdos podem ser vídeos, áudios, documentos, imagens e páginas web, por exemplo. Este modelo enfatiza o interesse pelo conteúdo e não a sua localização física ou lógica [Plagemann et al. 2006] . Assim, a comunicação pode se tornar mais eficaz à medida que os consumidores de informação especificam o conteúdo que pretendem consumir ao invés de especificar sua localização [Carzaniga et al. 2011] [Oh et al. 2010] [Amadeo and Molinaro 2011] [Amadeo et al. 2013]. Para tal finalidade, o paradigma de Redes Centradas no Conteúdo (*Information Centric Network* – ICN), também referenciado como Redes Centradas em Informação, foi proposto.

Em contraste com o paradigma tradicional da Internet que é baseado na pilha de protocolos TCP/IP, Redes Centradas no Conteúdo oferecem um modelo onde o conteúdo do objeto é foco da comunicação ao invés do endereço, de modo que, a informação chega até o solicitante sem que haja menção explícita de um endereço de origem. Este paradigma utiliza em sua arquitetura os conceitos de conteúdo nomeado, roteamento baseado em nomes, segurança aplicada ao conteúdo, *cache* nos elementos do núcleo da rede e modelos de interação que desacoplam o emissor do receptor [Jacobson et al. 2009] [Ahlgren et al. 2012]. Estes elementos permitem o desenvolvimento de uma arquitetura mais adequada para distribuição de conteúdo.

O problema de distribuição de conteúdo já vem sendo explorado através de propostas como as Redes de Distribuição de Conteúdo – *Content Distribution Network* (CDN) e redes par-a-par (*peer-to-peer* – P2P). Uma CDN é formada por um conjunto de servidores distribuídos geograficamente e interconectados através da Internet com a finalidade de cooperar na distribuição de conteúdo. Uma arquitetura típica de CDN é composta por duas classes de servidores: O servidor de origem e o servidor de réplica. O servidor de origem é responsável pelo armazenamento, atribuição de identificadores e divulgação do conteúdo. Por outro lado, o servidor de réplica é responsável por encaminhar o conteúdo para um determinado cliente. Assim, as requisições enviadas a um servidor de origem são redirecionadas para um servidor de réplica que está geograficamente mais próximo ao cliente que efetuou

a requisição [de Brito et al. 2012]. Sob outra perspectiva, uma arquitetura P2P é formada por nós distribuindo e consumindo conteúdo (ou compartilhando recursos computacionais) de maneira que a informação passa a ser difundida de forma descentralizada [de Brito et al. 2012]. BitTorrent e Kazaa são exemplos de duas aplicações que utilizam esta arquitetura. Questões relacionadas à segurança, interoperabilidade, custo computacional e financeiro fazem com que estas soluções não sejam as mais adequadas ao cenário atual da Internet.

Existem na literatura diversas projetos que implementam o conceito de Redes Centradas no Conteúdo [Jacobson et al. 2009] [Carzaniga et al. 2011] [Ain et al. 2009] [Koponen et al. 2007]. Embora estas abordagens apresentem diferenças umas das outras, estes projetos possuem premissas, objetivos e propriedades arquiteturais em comum [Ahlgren et al. 2012]. Nomeação de conteúdo (*Named Data Object* – NDO) é o conceito mais abstrato e comum a todas as implementações de Redes Centradas no Conteúdo. Um NDO é capaz de identificar semanticamente um único objeto independente da localização, método de armazenamento e mecanismo de comunicação. A seguir são descritos dois esquemas de nomeação mais utilizados nas Redes Centradas no Conteúdo [de Brito et al. 2012] [Ahlgren et al. 2012].

- **Nomeação Plana:** Os nomes planos podem ser definidos como cadeias de *bits* utilizados na identificação do conteúdo. O esquema de nomenclatura plana mais comum em Redes Centradas em Conteúdo se dá através da utilização de funções *hash* de criptografia. Neste sentido, um *hash* pode ser gerado a partir dos *bits* de entrada do conteúdo o que permite a autocertificação do mesmo. Assim, através da utilização de pares de *hashes* criptográficos no formato: *P:L*, sendo *P* o *hash* criptográfico do conteúdo e *L* o rótulo escolhido pelo publicador, os usuários possuem a capacidade de validar o conteúdo através do seu vínculo com o nome. Isso pode ser alcançado embarcando o *hash* do conteúdo no nome do objeto;

- **Nomeação Hierárquica:** O esquema de nomes hierárquicos contém uma estrutura similar aos Identificadores Uniformes de Recurso (*Uniform Resource Identifiers* – URI). Em oposição à nomenclatura plana, os nomes hierárquicos possuem uma característica semântica visto que sua estrutura pode ser composta por informações que dizem respeito ao conteúdo em questão: propriedade, versão e formato, por exemplo.

Redes Centradas no Conteúdo permitem a obtenção do conteúdo independente da especificação da localização do servidor, para tal foram desenvolvidos mecanismos de *caching* nos nós intermediários na rede. Deste modo, qualquer elemento na rede que contenha uma cópia do conteúdo tem a capacidade de responder a uma

requisição. Ademais, as requisições e entregas de conteúdo são realizadas através de APIs (*Application Programming Interface*) onde um produtor, ou origem, publica um conteúdo através de métodos como *publish* ou *register*. Por outro lado, os clientes consomem o conteúdo por meio de métodos como *get*, *interest*, *request*, *find* ou *subscribe*. A nomenclatura dos métodos pode variar de acordo com a implementação de uma ICN [Ahlgren et al. 2012]. A Figura 2.3 apresenta um modelo de uma Rede Centrada no Conteúdo.

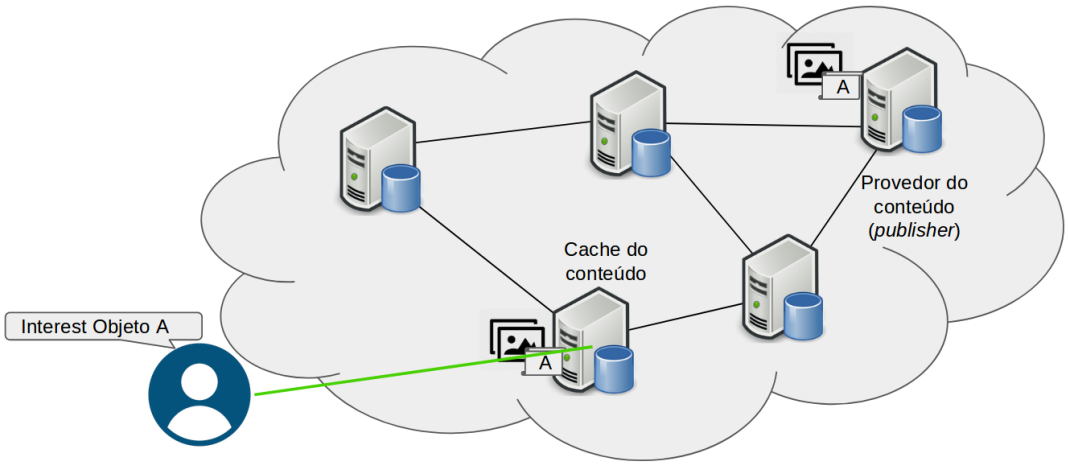

Figura 2.3: Modelo de uma Rede Centrada no Conteúdo.

Existem diversas implementações de Redes Centradas no Conteúdo disponibilizadas na literatura. Para melhor compreensão, a seguir são apresentadas duas implementações clássicas de Redes Centradas no Conteúdo.

2.2.1 Data-Oriented Network Architecture - DONA

Data-Oriented Network Architecture - DONA [Koponen et al. 2007] é uma das arquiteturas pioneiras no que se refere a implementação dos conceitos de Redes Centradas no Conteúdo. Na arquitetura DONA é utilizado o termo "orientado a dados"(*data-oriented*) ao invés do termo "centrado no conteúdo", portanto estes termos possuem o mesmo significado [Ahlgren et al. 2012]. DONA proporciona melhora na recuperação de conteúdo e acesso a serviços através de um suporte mais coerente para fornecer persistência, autenticidade e disponibilidade. A persistência e autenticidade é concedida através do uso de nomes planos e autocertificadores. Por fim, a alta disponibilidade é conferida através de um mecanismo de localização de conteúdo que é responsável por redirecionar as requisições dos usuário para os nós que contêm cópias do conteúdo original [de Brito et al. 2012] [Koponen et al. 2007].

Na arquitetura DONA todo nome é gerado por um outorgante (*principal*). Um outorgante é uma entidade associada a um par de chaves criptográficas que são utilizadas na identificação do conteúdo. Cada dado ou serviço é associado a um outorgante. Os nomes apresentam o formato P:L, sendo P o *hash* criptográfico da chave pública do outorgante e L um rótulo escolhido pelo mesmo, de modo que o nome de cada conteúdo seja único. Assim, os outorgantes são responsáveis pela publicação e administração de conteúdo de modo que apenas *hosts* autorizados pela chave do outorgante P podem prover acesso a objetos nomeados do tipo P:L. Ademais, cada dado é composto pelo conteúdo, chave pública e rótulo do outorgante, metadados e por uma assinatura do conteúdo [de Brito et al. 2012] [Koponen et al. 2007].

DONA utiliza o esquema de nomenclatura plana o que dificulta a associação dos nomes de conteúdo através dos usuários. Portanto, a arquitetura considera a implementação de mecanismos externos à rede, como sistemas de busca, comunicação privada e serviços de recomendação [de Brito et al. 2012].

De maneira oposta às redes tradicionais, DONA não utiliza servidores DNS, portanto, os mecanismos de resolução de nomes são implementados através de manipuladores de registro (*Resolution Handlers* – RHs). A resolução de nomes é implementada por meio de duas primitivas: *FIND(P:L)* e *REGISTER(P:L)*. Portanto, um cliente emite pacotes *FIND(P:L)* para o *RH* local a fim de localizar determinado objeto (P:L) que por sua vez encaminha a requisição para as cópias mais próximas do cliente que realizou a requisição. Em contrapartida, mensagens *REGISTER(P:L)* estabelecem o estado necessário para que *RHs* encaminhem os pacotes FINDs de forma eficaz [de Brito et al. 2012] [Koponen et al. 2007].

A Figura 2.4 apresenta um exemplo de funcionamento de uma arquitetura DONA. As requisições *FIND* são roteadas pelo nome em direção ao *RH* apropriado, como indicado nas etapas 1-4. Os dados são enviados de volta, em resposta à solicitação, por meio de um caminho reverso de RH (etapas 5-8), sendo possível realizar o cache da informação. Ademais, a informação pode ser enviada através de uma rota mais direta (etapa 9).

2.2.2 Content-Centric Networking - CCN

CCN *Content-Centric Networking* [Jacobson et al. 2009] é uma arquitetura de Rede Centrada no Conteúdo que utiliza o conteúdo como objeto elementar da rede. Na arquitetura CCN é utilizado o termo "centrado no conteúdo"(*content-centric*). Segundo [Jacobson et al. 2009] a proposta das CCNs é reutilizar os elementos bem sucedidos do TCP/IP, como encaminhamento de pacotes e *cache* no núcleo da rede, com a finalidade de construir uma nova rede baseada no conceito de Rede Centrada

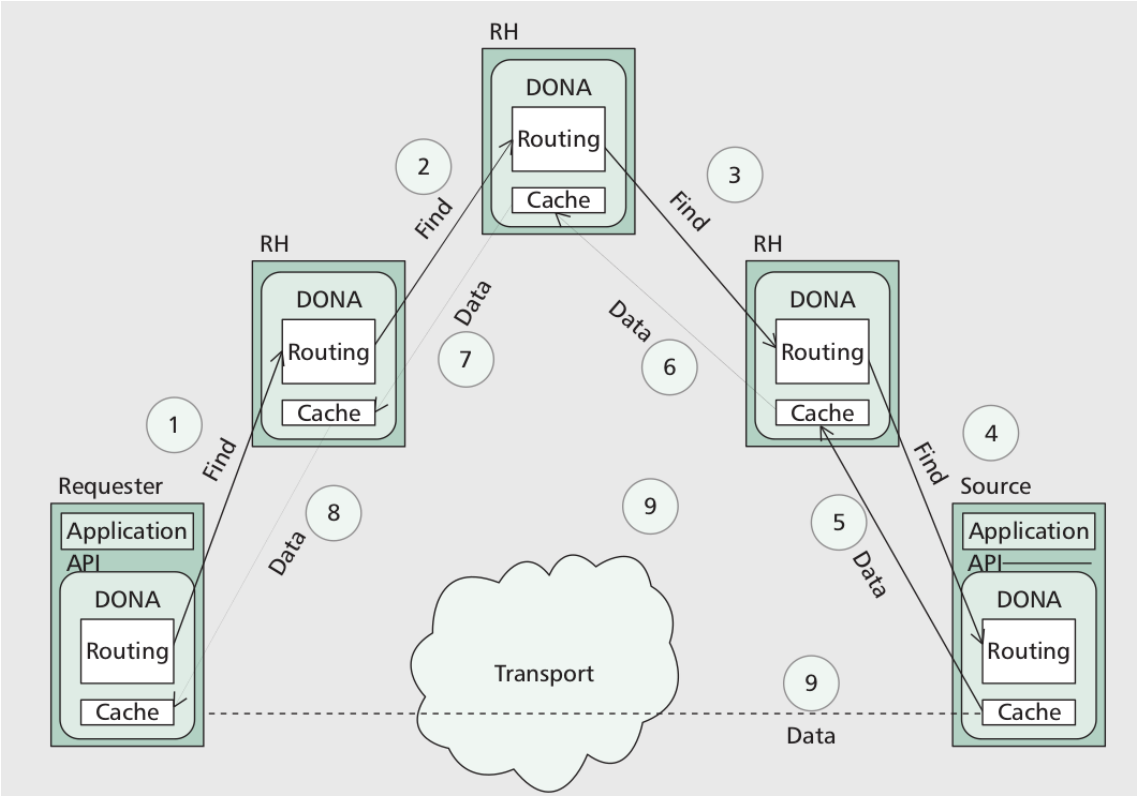

Figura 2.4: Arquitetura DONA [Ahlgren et al. 2012].

no Conteúdo.

A comunicação no CCN é realizada basicamente através de dois pacotes básicos: Interesses e Dados (Figura 2.5). Os pacotes de interesse são transmitidos pelos usuários que desejam receber determinado conteúdo. Estes pacotes contêm o nome do conteúdo solicitado através de um identificador hierárquico estruturado como uma URI, por exemplo, "/inmetro.gov.br/video/aula.avi". Os pacotes de interesses são enviados através de todas as interfaces disponíveis para todos os nós vizinhos. Caso um nó vizinho contenha o dado armazenado em memória é enviado um pacote de dados como resposta ao interesse. Caso contrário, o pacote de interesse é repassado para os nós vizinhos até, que eventualmente, chegue a um nó que contenha o dado armazenado. Resumidamente, os pacotes de dados são enviados apenas em resposta aos pacotes de interesse [Jacobson et al. 2009]

No CCN o encaminhamento dos pacotes é realizado através do mapeamento entre nome do conteúdo e a interface associada à arvore de distribuição de pacotes [de Brito et al. 2012]. Cada roteador CCN utiliza três estruturas de dados distintas em suas operações de encaminhamento de pacotes: Base de Informações de Encaminhamento (*Forwarding Information Base* – FIB), Armazém de conteúdos (*Content Store* – CS) e Tabela de interesses Pendentes (*Pending Interest Table* – PIT). A Figura 2.6 apresenta o esquemático do mecanismo de encaminhamento de pacotes no CCN [Jacobson et al. 2009].

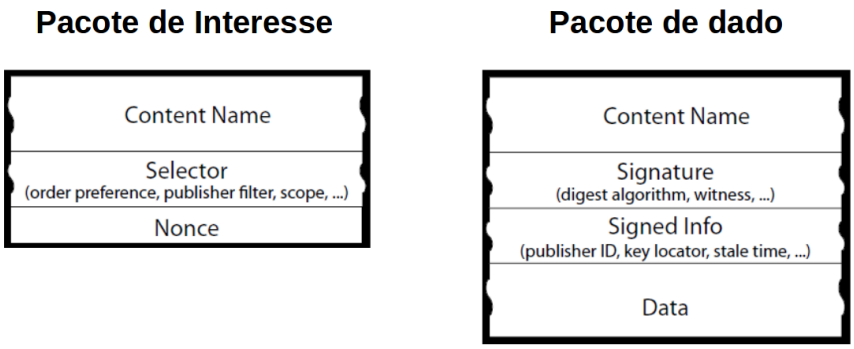

Figura 2.5: Tipos de pacotes no CCN [Jacobson et al. 2009].

A FIB é uma estrutura de dados utilizada para persistência de informações de encaminhamento de pacotes através do mapeamento entre o nome do conteúdo e uma, ou mais, interfaces de encaminhamento. Desta forma, o pacote de interesse é direcionado para os nós que, potencialmente, possuam os dados que foram requisitados. O CS é uma estrutura de *cache* do roteador no CCN, ou seja, a réplica dos dados são armazenados nos elementos de núcleo da rede. Além disso, esta estrutura utiliza políticas de atualização de *cache* como *Least Frequantly Used* (LFU) ou *Least recently used* (LRU). Finalmente, PIT mantém uma tabela de interesses pendentes, que correlaciona os interesses encaminhados com as interfaces de origem que os solicitou. Portanto, para que uma potencial fonte consiga responder a uma determinada solicitação de conteúdo, a tabela PIT possibilita aos pacotes percorrer o caminho, no sentido reverso, até o solicitante da informação.

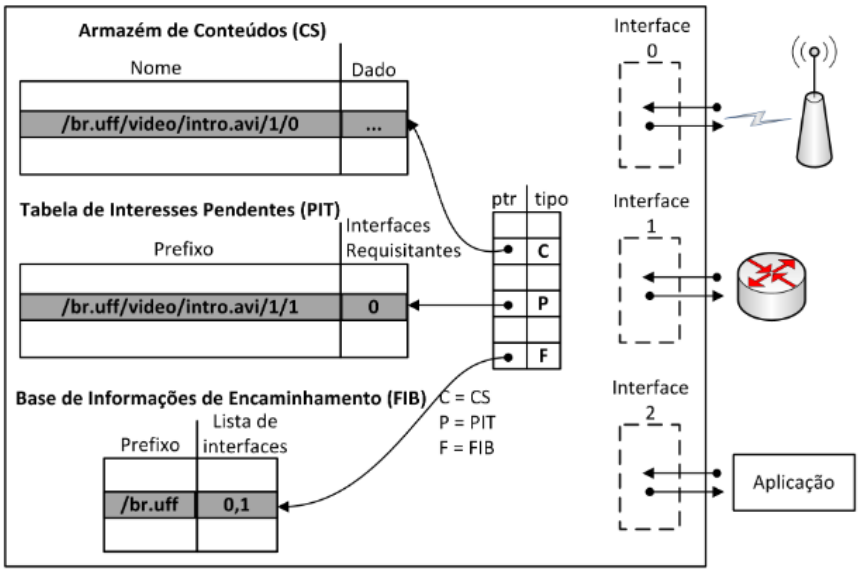

Figura 2.6: Estrutura do mecanismos de encaminhamento do CCN extraído de [de Brito et al. 2012].

2.3 Redes oportunísticas

Desde seu surgimento na década de 1970, redes sem fio vem se tornando cada vez mais populares, desde então, esta tecnologia vem sendo adaptada para prover mobilidade [Royer and Toh 1999]. Conseguinte, foram definidos dois principais modos de operação destas redes, descritos a seguir [Kurose and Ross]:

- **Infraestrutura:** Os dispositivos móveis, também chamados de estacionários, estão associados a uma estação base e os serviços tradicionais da rede, como endereçamento e roteamento, são definidos e proporcionados pela infraestrutura da rede na qual os dispositivos estão conectados. Neste cenário, quando um dispositivo móvel se transporta para fora do raio de alcance da estação base e entra em outro raio, o dispositivo altera seu ponto de conexão para continuar a comunicação em outra estação base. Este processo é denominado transparência (*handoff*);

- ***Ad hoc:*** Neste modo os dispositivos móveis não possuem infraestrutura básica de redes como estação base e/ou roteadores fixos. Assim, os próprios dispositivos conectados na rede devem prover serviços de endereçamento e roteamento.

De acordo com [Loo et al. 2016], em uma rede *ad hoc* sem fio móvel (*Mobile Ad Hoc Network* – MANET) todos os nós sem fio podem se mover e manter a conexão através da comunicação *multhop* sem a necessidade de infraestrutura de rede, com um ponto de acesso, previamente estabelecida. Assim, o objetivo destas redes é suportar operações robustas e eficientes através da implementação de funcionalidades de roteamento nos dispositivos móveis.

Durante um longo período de tempo, os esforços de pesquisa e desenvolvimento em redes sem fio móveis permaneceram no âmbito militar. No entanto, com o advento das tecnologias de comunicação sem fio comerciais, a comunidade de pesquisa em redes sem fio notou o potencial e as vantagens da mobilidade das MANETs [Agrawal and Chauhan 2015]. Deste modo, MANETs possuem uma vasta gama de aplicações em diversos campos como no domínio militar [Oh et al. 2010] [Etefia and Zhang 2012], serviços de emergência e redes domésticas [Loo et al. 2016]. Portanto, são propostos novos protocolos e soluções à medida que surgem novas demandas e aplicações para redes sem fio móveis. Uma das demandas mais pertinentes são as redes desafiadoras (*challenged networks*) [Fall 2003].

Em ambientes desafiadores, a comunicação entre os dispositivos da rede é bastante restrita devido à conectividade intermitente, longos atrasos, limitação na largura de banda e alta taxa de erro. A mobilidade dos nós é um fator de extrema influência nestes ambientes, pois é uma das principais causas de desconexão entre

os nós causando grande imprevisibilidade na topologia da rede. Ademais, os protocolos de roteamentos tradicionais não são adequados a este novo cenário devido à dificuldade no estabelecimento de conexão entre os dispositivos de origem e destino. Portanto, para estes cenários foi proposto uma arquitetura de sobrecamada (*overlay*) chamada Rede Tolerante a Atraso (*Delay-Tolerant Networking* – DTN) [Fall 2003]. Uma DTN é uma arquitetura que consiste de uma rede independente de conectividade com a Internet onde a comunicação entre os dispositivos ocorre eventualmente ou de modo oportunístico. Neste cenário, cada comunicação oportunística pode ocorrer de forma agendada ou não determinística [Pelusi et al. 2006]. Entre as características de uma DTN está a adoção do paradigma Armazena-Carrega-Encaminha (*Store-Carry and Forward*) no qual os nós armazenam cópias das mensagens e as encaminham, de modo oportunista, à medida que uma nova conexão é estabelecida. Os encaminhamentos oportunistas caracterizam as chamadas Redes Oportunísticas.

Redes oportunísticas (*Opportunistic networks* – OppNets) são um tipo especial de MANET e uma subcategoria das DTNs. As redes oportunísticas herdam as características das redes DTN, porém, estas exploram a mobilidade e o encontro dos dispositivos com a finalidade de proporcionar a entrega dos dados. Um encontro, neste sentido, ocorre quando dois nós estão no mesmo alcance de raio um do outro possibilitando a troca de dados entre si [Pelusi et al. 2006] [Trifunovic et al. 2017].

Nas redes oportunísticas, os dispositivos móveis armazenam as informações e as carregam de acordo com a mobilidade do dispositivo até que surja uma oportunidade de comunicação para encaminhar os dados armazenados [Trifunovic et al. 2017]. Com isso, nesta categoria de redes, não há o pressuposto que exista um caminho estabelecido entre dois dispositivos, ou nós, que se comunicam. Deste modo, os nós de origem e destino podem nunca estar conectados à mesma rede e ao mesmo tempo [Pelusi et al. 2006]. Portanto, questões relacionadas ao roteamento e encaminhamento são primordiais, visto que, estabelecer uma rota entre o solicitante e o destinatário é considerado um desafio devido às características de um ambiente altamente desconectado.

De acordo com este paradigma, os dados são "movidos"pela rede, não apenas através do encaminhamento de mensagens entre os nós, mas também pela mobilidade dos próprios dispositivos que transportam mensagens enquanto esperam entrar na faixa de rádio dos nós intermediários ou do nó destinatário. Neste sentido, redes oportunísticas são tradicionalmente consideradas como principal mecanismo de prover serviços de comunicação entre dispositivos portáteis onde não há uma infraestrutura de rede estabelecida [Conti et al. 2015].

Diferente das DTNs, redes oportunísticas não possuem conhecimento sobre a topologia da rede, cada nó atua como um roteador ou *gateway*. Além disso, redes

oportunísticas são centradas em seres humanos (*human-centric*) pois exploram o contato entre as pessoas para realizar a comunicação. Assim, estas redes estão profundamente ligadas às redes sociais e exploram as relações humanas para desenvolver protocolos de comunicação mais eficientes, adequados e confiáveis.

2.3.1 Exemplo de uma Rede Oportunística

A Figura 2.7 apresenta um exemplo de comunicação através de redes oportunísticas onde uma mulher, em seu computador (*desktop*), transfere oportunisticamente uma mensagem para um amigo. A mensagem é inicialmente transferida, através de uma rede Wi-fi, até um ônibus que irá transportar a mensagem para mais próximo do destinatário da mensagem. O ônibus se move pelo tráfego e envia a mensagem, através do *bluetooth*, para uma mulher que está desembarcando em um ponto de ônibus. Ela caminha até a universidade passando por um parque onde seu celular envia a mensagem para um ciclista. Seguindo esta lógica, mais adiante, a mensagem é finalmente entregue ao destinatário.

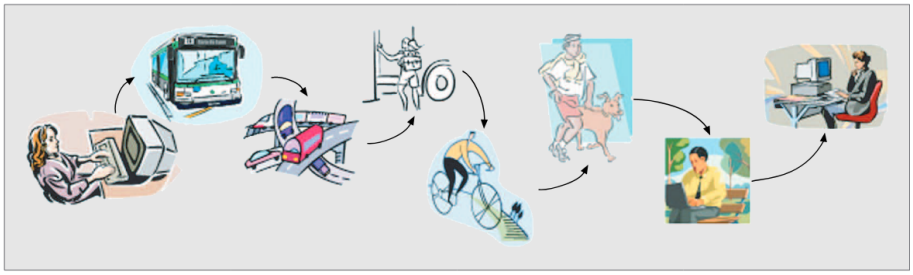

Figura 2.7: Exemplo de uma rede oportunística [Pelusi et al. 2006].

Em [Juang et al. 2002] é apresentado o projeto ZebraNet, um projeto de monitoramento da vida selvagem desenvolvido pela Universidade Princeton sob a orientação do Centro de Pesquisa de Mpala[1]. O cenário de implantação desta aplicação é a vasta área de savana do Quênia onde o objetivo é monitorar e rastrear atividades de animais selvagens em seus habitats naturais, no caso, zebras. No cenário desta aplicação, a estação base consiste de um veículo móvel, utilizado pelos pesquisadores, que se movimenta periodicamente pela savana coletando dados das zebras encontradas. Os resultados dos experimentos reais são disponibilizados e utilizados para definir um modelo de mobilidade usado para testes de técnicas de encaminhamento em redes oportunísticas.

[1]http://www.mpala.org/

2.3.2 Protocolos de roteamento/encaminhamento

Considerando os exemplos citados na Seção anterior é possível notar que roteamento e encaminhamento de mensagens é uma das questões mais importantes em redes oportunísticas. Portanto, diversos protocolos de roteamento vem sendo estudados e propostos [Sobin et al. 2016]. Nesta Seção são relatadas duas abordagens fundamentais para o roteamento em redes oportunísticas.

Em [Vahdat et al. 2000] é proposto o protocolo Roteamento Epidêmico (*Epidemic routing*). Neste protocolo as mensagens são disseminadas na rede por meio de contatos entre os nós usando o conceito de inundação, ou *flooding* em inglês. Cada nó contém dois *buffers*. O primeiro *buffer* é usado para armazenar as mensagens geradas pelo próprio nó, enquanto que, o segundo *buffer* é utilizado para armazenar as mensagens recebidas por outros nós da rede. Ademais, cada mensagem contém uma identificação única (ID) e cada nó integra uma lista das mensagens armazenadas, denominada *Summary Vectors*. Assim, quando ocorre um contato entre os nós, eles comparam seus *Summary Vectors* e requisitam as mensagens que não possuem em comum em seus *buffers*. Por fim, a mensagem é entregue ao seu destinatário quando este recebe a mensagem que contém o seu endereço. A Figura 2.8 ilustra a troca de mensagens entre os nós.

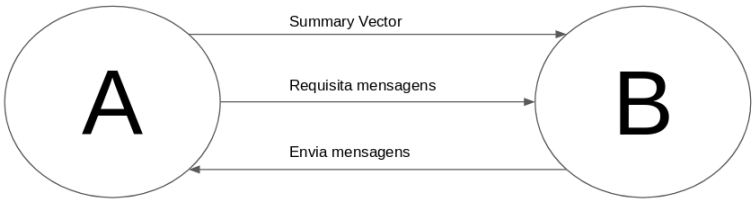

Figura 2.8: Troca de mensagens entre nós usando *Epidemic Routing*.

Este protocolo atua de forma similar a uma epidemia de uma doença ou virose, por exemplo. Um nó é "infectado"quando produz a mensagem que será disseminada ou quando recebe uma mensagem de um outro nó da rede. Por outro lado, um nó é considerado "suscetível"quando ainda não contém a mensagem armazenada em seu *buffer*. Por fim, um nó "infectado"se torna "recuperado"(curado da doença) quando a mensagem é entregue ao destinatário e, como resultado, se torna imune à mesma doença pois não fornece mais a mesma mensagem. Devido ao grande número de mensagens redundantes este protocolo demanda de capacidade de armazenamento dos nós e largura de banda [Dhurandher et al. 2013].

O protocolo de roteamento probabilístico utilizando históricos de encontros e transitividade (*Probabilistic Routing Protocol using History of Encounters and Transitivity* – PROPHET) [Lindgren et al. 2003] é uma evolução do Roteamento

Epidêmico. Neste protocolo existe uma informação extra para cada mensagem na lista. Esta informação é referente à probabilidade de cada nó entregar mensagens para um destino baseada no histórico de encontros entre os nós.

No PROPHET, quando um nó recebe a lista de mensagens (*Summary Vectors*) do vizinho, é calculada a probabilidade de entrega para cada uma das mensagens que ainda não possui armazenada em seu *buffer*. Em seguida, para cada mensagem da lista, o nó compara a probabilidade representada na sua lista com a probabilidade representada na lista recebida do vizinho. Essa comparação é realizada com a finalidade de verificar qual dos dois nós apresenta a maior probabilidade na entrega da mensagem em questão. Quando esta comparação é finalizada são realizados os seguintes procedimentos: o nó solicita as mensagens não armazenadas que possuem uma probabilidade maior de serem entregues através dele; recebe a solicitação de mensagens do vizinho e as envia; finalmente, remove do *buffer* todas as mensagens na qual o vizinho apresenta maior probabilidade de entrega. Ao final do processo, cada nó armazena apenas as mensagem que possuem probabilidades de sucesso de entrega através dele.

2.4 Redes Ad Hoc Centrada em Interesse - RadNet

Rede Ad Hoc Centrada em Interesse (RadNet) é um protocolo de comunicação, proposto para redes móveis, o qual se enquadra no modelo Publicador / Subscritor (*Publisher / Subscriber* – Pub/Sub). Este modelo utiliza uma fila de mensagens assíncronas onde as mensagens são armazenadas pela aplicação até serem utilizadas pelo usuário. Um nó publica na rede uma mensagem com um interesse registrado e o nó que possui o mesmo interesse registrado em sua aplicação recebe a mensagem publicada. Neste sentido, interesse é definido como qualquer termo que tenha significado para o usuário e para aplicação. Por exemplo, em um aplicativo de *streaming* um interesse pode se referir a uma informação genérica como "filmes"ou a uma informação específica como "filmes/Keanu Reeves". O conceito de interesse da RadNet compreende um mecanismo geral de comunicação que pode ser estendido tanto para aplicações tradicionais, incluindo comércio eletrônico e jogos, quanto para aplicações em áreas desafiadores como emergência e segurança. Deste modo, o foco da comunicação é o conteúdo da informação de interesse e não o equipamento na qual a informação está contida, como nas redes convencionais [de Castro Dutra 2012]. O protocolo RadNet foi originalmente proposto para uso em redes móveis *ad hoc* (*Mobile Ad-hoc NETworkS* – MANETS) [Dutra et al. 2012]. Em uma RadNet os nós participantes utilizam de um mecanismo de nomeação chamado Prefixo Ativo (PA)

que tem a finalidade de compensar a falta de infraestrutura básica de uma MANET. O PA (Figura 2.9) é uma estrutura de dados implementada na camada de rede do dispositivo e é composto pelo prefixo do dispositivo e pelo interesse da aplicação. Nesta perspectiva, o prefixo é construído de modo a permitir identificação do nó, encaminhamento probabilístico de mensagem e endereçamento. Complementarmente, o interesse da aplicação é utilizado para identificação do conteúdo que se deseja compartilhar [Dutra et al. 2012].

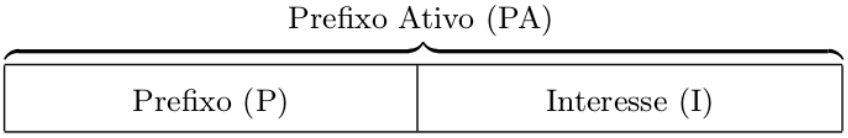

Figura 2.9: Prefixo Ativo (PA) [Silva et al. 2013].

Nesta perspectiva, o interesse possui duas funções: descarte da mensagem, caso não haja correspondência de interesse entre o nó receptor e a mensagem; e repasse da mensagem da camada de rede para a camada de aplicação (*cross-layer*). Ademais, a implementação do interesse no cabeçalho da mensagem possibilita que o nó descarte a mensagem diretamente na camada de rede de modo a reduzir o tempo de processamento em consequência da mensagem não ser repassada para a camada mais alta [de Castro Dutra 2012].

2.4.1 Estrutura da mensagem

Assim como nas mensagens dos protocolos de redes convencionais, a mensagem da RadNet é composta por um cabeçalho e um *payload* [de Castro Dutra 2012]. A Figura 2.10 apresenta a estrutura da mensagem utilizada pelo protocolo RadNet. A Figura 2.10 (a) apresenta o Prefixo Ativo (PA) que é composto pelo prefixo do nó (Pi) e pelo interesse da aplicação (I). A Figura 2.10 (b) mostra o cabeçalho da mensagem que contém a versão do protocolo (Ver), o número de saltos da mensagem (HTL), o comprimento do cabeçalho (CdC), o identificador da mensagem (ID), prefixo de origem, prefixo de destino e um campo de interesse da aplicação [Dutra et al. 2012].

Cada nó concebe seu prefixo com n campos, cada qual com m bits, assim, nxm bits provê a identificação do nó para fins de endereçamento e filtro de casamento [Dutra et al. 2012]. Dependendo do número de campos no prefixo, os dispositivos na rede podem estar completamente conectados ou não. Deste modo, utilizar somente dois campos no prefixo fará com que os nós tenham apenas duas possibilidades de endereços, em contrapartida, utilizar um prefixo complexo e maior poderá impedir duplicatas, aumentar a probabilidade de encaminhamento e conexão entre os

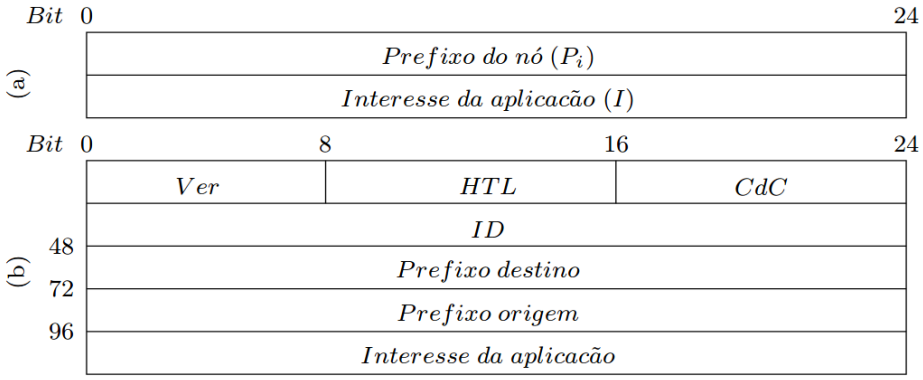

Figura 2.10: Elementos da RadNet: (a) Prefixo Ativo e (b) Cabeçalho da mensagem [de Castro Dutra 2012].

dispositivos. O protocolo de comunicação implementa o encaminhamento de mensagem aplicando o filtro de casamento através do uso do campo do prefixo do nó [Dutra et al. 2012]. A RadNet implementa dois tipos de endereçamento com base no Prefixo Ativo:

- ***Source-to-destination* (S2D):** Neste modo de endereçamento o nó de origem descreve o Prefixo de destino (Figura 2.10 (b)) no cabeçalho de sua mensagem. Assim, a mensagem enviada é dirigida a um nó específico da rede;
- ***Interest-Group* (IG):** Em contrapartida, neste modo de endereçamento o nó de origem informa apenas o campo de interesse da aplicação, assim, a mensagem é destinada para um grupo de nós que correspondem ao interesse indicado na mensagem. Ademais, o campo Prefixo de destino pode ser especificado como *null* de modo a indicar que é uma mensagem *broadcast* e não possui um destino específico.

A mensagem é encaminhada na rede, através dos nós intermediários, com a finalidade de encontrar um nó com um interesse em comum, isto é, com o mesmo interesse registado. Para esta finalidade, o Prefixo Ativo é utilizado como "filtro de casamento". Caso ocorra correspondência de interesse entre os dispositivos intermediários a mensagem é copiada e entregue para a aplicação local. Esta comunicação configura o modo de endereçamento IG. Quando um nó recebe uma mensagem da rede, este passa a conhecer a origem da mensagem recebida, assim sendo, a comunicação pode ser continuada através do modo S2D, onde o nó de origem informa o prefixo de destino no cabeçalho da mensagem.

2.4.2 Desacoplamento na RadNet

Conforme já citado, o protocolo RadNet se enquadra no modelo Publicador / Subscritor (*Publisher / Subscriber*). Neste paradigma de comunicação, os produtores de informação submetem dados como publicações e os consumidores de informação indicam seu interesse através da submissão da assinatura [Fidler et al. 2005]. Neste sentido, um assinante, ou *Susbscriber*, possui a capacidade de expressar um interesse em determinado evento ou conjunto de eventos. Por outro lado, um publicador, ou *Publisher*, é responsável por gerar o evento que será consumido pelo assinante de acordo com o interesse informado. Assim, um evento é propagado de forma assíncrona para todos os assinantes que registraram interesse nesse determinado evento. Aplicações de larga escala se beneficiam deste modelo de comunicação através da redução do acoplamento entre os assinantes e publicadores de eventos [Eugster et al. 2003].

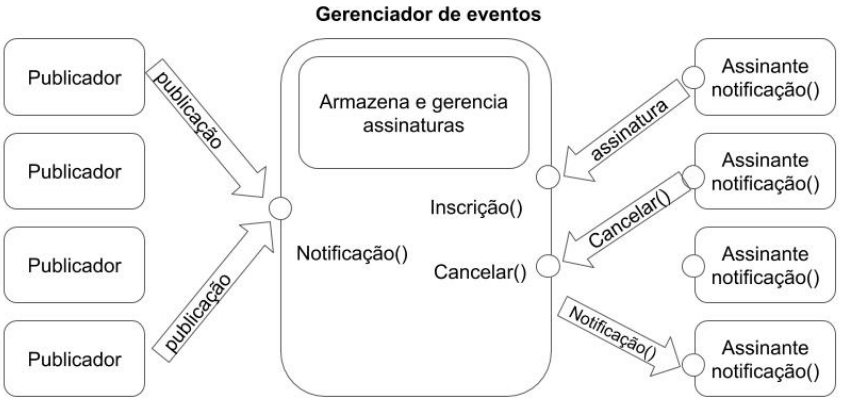

Figura 2.11: Modelo Pub/Sub.

Conforme ilustrado na Figura 2.11 o modelo Pub/Sub é composto pelo Publicador, pelo Assinante e pelo gerenciador de eventos.

De acordo com [Eugster et al. 2003] o esquema de comunicação Pub/Sub provê três dimensões de desacoplamento, descritos a seguir:

- **Espacial:** O desacoplamento espacial provê desobrigação de conhecer a localização entre o provedor e consumidor da informação. Deste modo, o publicador publica eventos através do gerenciador de eventos e os assinantes recebem tais eventos, indiretamente, através de notificações provenientes do gerenciador de eventos;

- **Sincronismo:** No tocante ao processamento, o desacoplamento de sincronismo possibilita que os produtores não fiquem bloqueados durante a

produção/submissão de eventos, da mesma forma, os assinantes podem ser notificados de forma assíncrona (por meio de *call-back*, por exemplo) enquanto executam outras funções;

- **Temporal:** O desacoplamento temporal possibilita que as partes interessadas se comuniquem independente de estarem ativas no mesmo instante de tempo. Assim, os eventos e notificações podem ser gerados e enviados para o gerenciador de eventos que os armazena até os respectivos assinantes estarem disponíveis.

À vista disso, o protocolo RadNet originalmente implementa parcialmente os desacoplamentos propostos pelo modelo Pub/Sub. O desacoplamento espacial é atendido por padrão, visto que a rede é distribuída. O desacoplamento de sincronismo é atendido no nível da aplicação desenvolvida através do uso de *call-backs* e *threads*, por exemplo. Por fim, o desacoplamento temporal não foi implementado originalmente no protocolo RadNet. Assim, foi proposto em [Silva et al. 2013] um mecanismo para adicionar persistência de dados para que as aplicações possam receber mensagens de seus interesses posteriormente, quando não estiverem ativas.

2.4.3 Exemplo de comunicação na RadNet

O exemplo da Figura 2.12 ilustra a transmissão de dados entre quatro dispositivos na RadNet, onde o raio de transmissão sem fio é delimitado pela circunferência tracejada. Nesse cenário, cada nó contém um PA com dois campos numéricos, e um interesse registrado na camada de rede. A comunicação inicia através do envio da mensagem do nó A com prefixo [1;5] e interesse [Futebol], i.e., PA:[1;5;Futebol]. O nó B, no raio de alcance da transmissão de A, recebe o pacote proveniente de A, e encaminha a mensagem de A por haver casamento de prefixos de A com B (ex. critério de casamento: ambos PAs tem o mesmo valor 5 no 2º campo). O nó A recebe o pacote de volta encaminhado por B, porém detecta que o pacote já foi processado anteriormente e o descarta. O Nó C, recebe e encaminha a mensagem de A por haver casamento de prefixos no 1º campo(=1). O nó D, ao receber o pacote, detecta que possui o mesmo interesse (Futebol) e o repassa para a aplicação local, mas não há casamento de prefixos, logo descarta a mensagem.

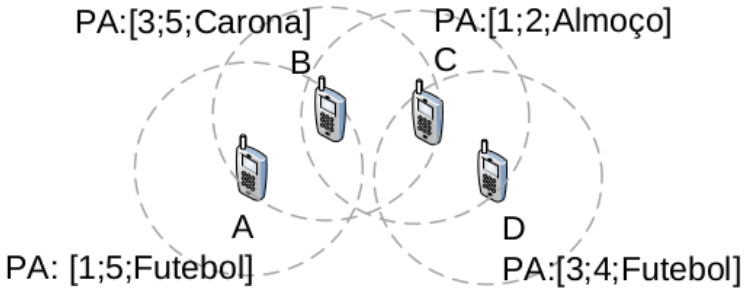

Figura 2.12: Exemplo de comunicação usando RadNet - Extraído de [Salles 2014].

2.5 Considerações sobre a RadNet

Nesta dissertação a RadNet é utilizada como uma Rede Oportunística Centrada em Interesse. Portanto, assumiu-se que a RadNet integra algumas características de Redes Centradas no Conteúdo e Redes Oportunísticas para a finalidade da aplicação proposta neste trabalho. De acordo com [de Castro Dutra 2012] a RadNet pode ser definida de acordo com a programação da aplicação em questão.

De acordo com as definições dos tipos de redes, a RadNet contém características em comum com uma Rede Centrada no Conteúdo, visto que na RadNet não existe uma dependência da implementação da pilha de protocolos TCP/IP para realizar a comunicação. Porém, numa RadNet o roteamento e encaminhamento são baseados no prefixo do dispositivo e não no conteúdo da informação. Em contrapartida, a RadNet não implementa o conceito de *cache* nos elementos de núcleo da rede.

A RadNet possui, também, características de uma Rede Oportunística visto que este protocolo foi originalmente proposto para redes sem fio móveis Ad Hoc (MANETs). A falta de hierarquia na comunicação entre os nós e implementação de mobilidade permitem ao protocolo atuar em ambientes onde não existe infraestrutura de redes disponível para realizar a comunicação entre os dispositivos, como é proposto em uma Rede Oportunística. Porém, na RadNet, como já citado, não implementa *cache* para armazenamento de mensagens, impossibilitando a adoção do paradigma Armazena-Carrega-Encaminha. Para tal, neste trabalho foi implementada uma solução para armazenamento de mensagens, temporariamente em disco, para possibilitar carregar as informações até os pontos de interesse.

Capítulo 3

Sistema Hermes

Este capítulo apresenta o Sistema Hermes, uma aplicação de monitoramento de ambiente baseada em uma rede oportunística centrada em interesses, para tal, está divido em quatro seções. A Seção 3.1 apresenta o conceito da aplicação. A Seção 3.2 apresenta o procedimento técnico de desenvolvimento do Projeto Hermes. Finalmente, a Seção 3.3 ilustra o funcionamento da aplicação proposta neste trabalho.

3.1 Conceito da aplicação

O Sistema Hermes é um aplicação de monitoramento remoto de ambiente que utiliza uma rede oportunística centrada em interesse como mecanismo de transmissão de dados entre os dispositivos. Entre as características da abordagem adotada para este sistema é destacada sua aplicabilidade em ambientes onde não há disponibilidade de infraestrutura de redes, além de garantir a disponibilidade do serviço de monitoramento mesmo quando a rede local, baseada na camada de redes da pilha de protocolos TCP/IP, está indisponível.

Como indicado na Figura 3.1, o sistema de sensoriamento proposto é composto por três categorias de agentes: Coletor, Mensageiro e Servidor. Portanto, estes agentes possuem como finalidade coletar, transportar e armazenar informações de ambiente, respectivamente. Neste cenário foram utilizados dispositivos Raspberry Pi[1], modelo 3 B+, como agentes Coletores, os quais foram conectados a sensores para realizar a coleta e registro de dados de sensoriamento, armazenando temporariamente as informações geradas. No escopo deste trabalho, o agente Mensageiro é composto por um dispositivo Raspberry PI acoplado a um veículo aéreo não tripulado (Drone), modelo Phantom 4 Pro plus[2] do fabricante DJI, para realizar o

[1]https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf

[2]https://www.dji.com/br/phantom-4-pro/info

transporte dos dados até um servidor. O agente Servidor é responsável por armazenar os dados recebidos pelo agente Mensageiro, disponibilizá-los para outros serviços e aplicações disponíveis na rede e gerenciar o banco de dados.

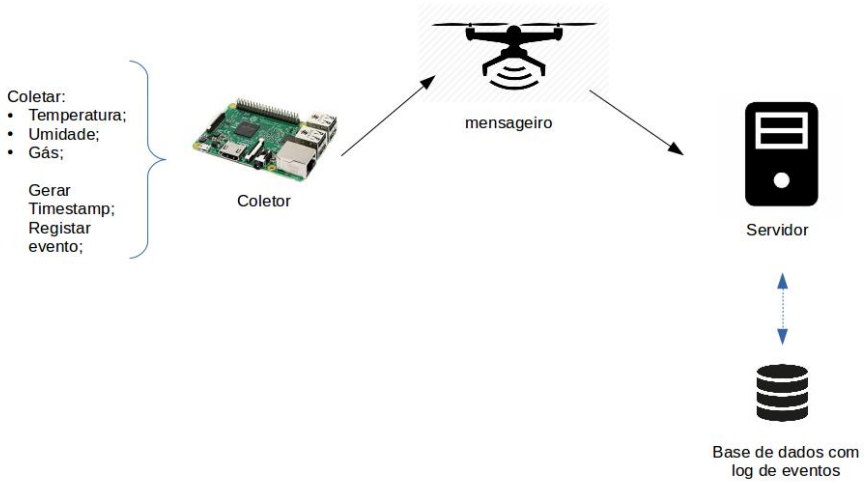

Figura 3.1: Representação em alto nível de aplicação de monitoramento.

Neste projeto, o agente Mensageiro é composto por um drone, o qual é utilizado para realizar o transporte oportunístico das informações geradas pelos agentes Coletores. Porém, vale ressaltar que o agente Mensageiro pode ser qualquer dispositivo que possua capacidade de mobilidade, como o *smartphone* de uma pessoa que percorre um campus universitário ou um *laptop* em uso sendo transportado por um aluno dentro de um ônibus, por exemplo.

3.2 Desenvolvimento

A arquitetura de software base do Sistema Hermes é formado pelas classes *Device*, *Collector*, *Messenger*, *Server* e *Sensor*, como indicado na Figura 3.2. Nesta arquitetura, todos os dispositivos possuem uma identificação, um interesse e uma localização. A classe *Messenger* não implementa o atributo *location*, pois esta será implementada num dispositivo móvel (*smartphone*, *tablets* e drone, por exemplo) e não terá uma localização fixa. Por fim, a classe *Sensor* representa os sensores a serem implementados pelo projeto, por exemplo, temperatura e umidade (DHT22),

gás (MQx) e proximidade (PIR).

O projeto foi inteiramente desenvolvido através da linguagem de programação Python na versão 2.7. No tocante à transmissão de informação foi utilizado o protocolo de redes oportunísticas Centrada em Interesse, denominado RadNet [Dutra et al. 2012]. O Python foi adotado neste projeto por se tratar de uma linguagem de programação multiplataforma: em geral, o Python pode ser executado em sistemas Windows e derivados do Unix, como Linux e Mac OS X. Portanto, a aplicação de sensoriamento desenvolvida para uma arquitetura Desktop (x86/x64) pode ser executada, com poucas ou nenhuma alteração, em arquitetura ARM como no caso do Raspberry Pi. Ademais, a biblioteca-padrão do Python é bastante completa e possui uma série de recursos disponíveis, como leitura e serialização de arquivos em formato JSON. Além disso, o protocolo RadNet possui uma API para desenvolvimento de aplicações em Python o que viabiliza a implementação do protocolo de comunicação junto à aplicação de sensoriamento.

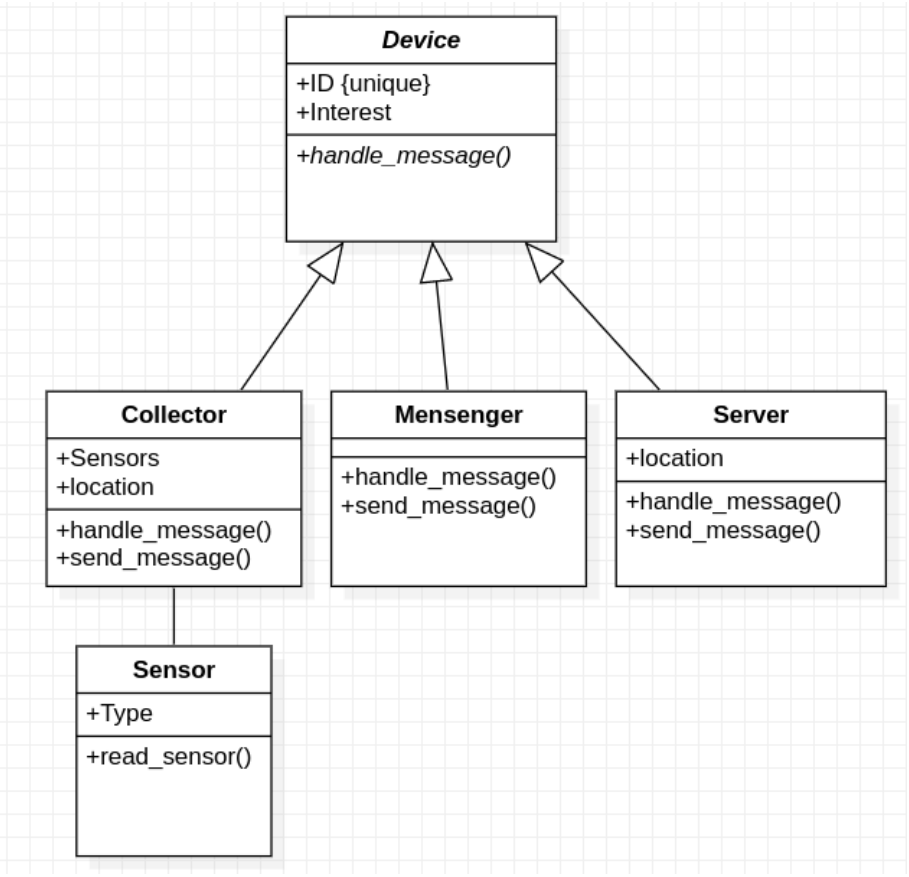

Figura 3.2: Arquitetura do Projeto Hermes.

3.2.1 Protocolo de comunicação

O protocolo Rede Oportunística Centrada em Interesses, denominado RadNet, introduzido na Seção 2.4, foi selecionado como protocolo de comunicação para aplicação de monitoramento remoto proposta neste trabalho. A RadNet foi implementada em sistema Linux (Desktop/ARM) como um serviço de comunicação entre os dispositivos de modo a ser responsável pelo envio, recebimento e encaminhamento da mensagem para aplicação local interessada. Este serviço é executado em segundo plano, podendo ser acessado, também, através de uma API com suporte às linguagens de programação C/C++, Java e Python. A Seção 2.4 apresentou detalhes sobre o funcionamento do protocolo.

3.2.2 Computador Raspberry Pi

Raspberry Pi (Figura 3.3) é um computador de placa única e de tamanho reduzido, desenvolvido no Reino Unido pela Fundação Raspberry Pi. A fundação fornece a distribuição Raspbian, que é baseada no Debian, e possui o Python como a principal linguagem de programação. A placa Raspberry Pi contém um processador, *chip* gráfico, memória RAM (*Random-Access Memory*) e diversas interfaces e conectores para dispositivos externos [Upton and Halfacree 2014].

Figura 3.3: Dispositivo Raspberry Pi.

O Raspberry Pi foi escolhido para este trabalho devido a sua simplicidade de instalação, implementação, programação e, principalmente, compatibilidade com a RadNet, visto que este protocolo é executado sobre a plataforma Linux. Ademais, o Raspberry pode ser configurado no modo *ad hoc* que possibilita a comunicação entre os dispositivos sem a necessidade de um ponto de acesso ou infraestrutura de

rede previamente estabelecida. Os procedimentos de instalação e configuração estão descritos no Apêndice A.

3.2.3 Agente Coletor

O agente Coletor é responsável pela leitura de informações de ambiente (temperatura, umidade, presença e gases), armazenamento temporário e transferência de dados. À vista disso, este agente exerce a função de produtor de informação. O agente Coletor é representado pela classe *Collector*, como indicado na Figura 3.2. A Classe *Sensor* corresponde ao sensor acoplado ao dispositivo, por exemplo, o DHT22. A Figura 3.4 apresenta a implementação física do agente coletor.

O agente Coletor é composto pelos seguintes elementos:

- Placa Raspberry Pi 3, modelo B+, 1GB de RAM;
- Sensor de temperatura e umidade DHT22;
- Sensor de movimento PIR;
- Sensor de Gas MQ-2.

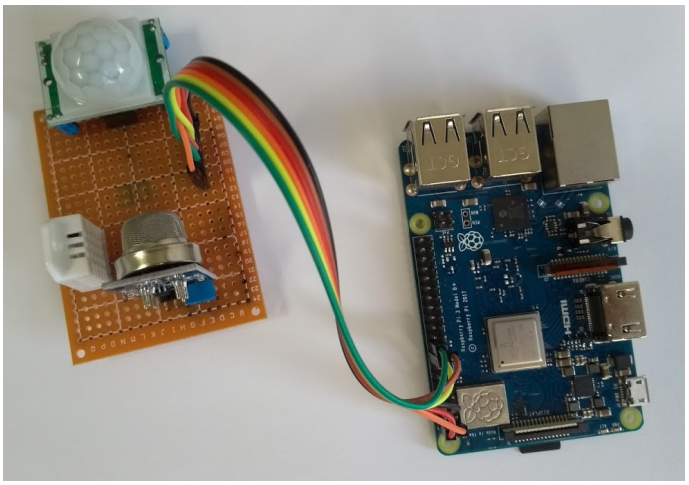

Figura 3.4: Dispositivo Raspberry com sensores - Coletor.

Sensor de temperatura e umidade - DHT22

O sensor DHT22 (Figura 3.5) é um módulo utilizado para realizar leituras de temperatura, entre -40 e 80 graus Celsius, e umidade, entre 0 e 100%, em tempo real. A temperatura possui uma precisão de 0,5 °C e umidade de 2%. A saída do DHT22 é um sinal digital calibrado que pode ser conectado diretamente ao pino de porta do Raspberry Pi [Bogdan 2016].

O DHT22 consiste em uma medição de temperatura do termistor e um sensor capacitivo para determinar a umidade.

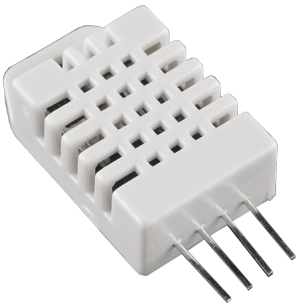

Figura 3.5: Sensor DHT.

Sensor de movimento PIR

Sensores PIR, apresentado na Figura 3.6, permitem detectar movimentação de pessoas, animais e objetos através da mudança no padrão de energia térmica. Devido seu baixo custo, facilidade de uso e resistência, estes sensores são frequentemente utilizados em aplicações de tecnologia como domótica e sistemas de intrusão.

Figura 3.6: Sensor PIR.

O sensor possui um campo de visão, como exibido na Figura 3.7. Este campo é a zona em que as alterações na radiação infravermelho podem ser detectadas. O campo de visão destes sensores varia de acordo com o fabricante, temperatura ambiente e tamanho da fonte de calor. Assim, quando uma fonte de calor se aproxima do campo de visão, o nível de radiação infravermelha nessa área aumenta. Essa alteração é detectada e processada pelo sensor, iniciando o processo de *Time*. Enquanto a fonte de calor estiver no campo de visão, o sensor permanece detectando e processando as mudanças de energia. Quando a fonte de calor se move para fora da zona, a detecção e processamento de *Time* são encerrados [Osman et al. 2009].

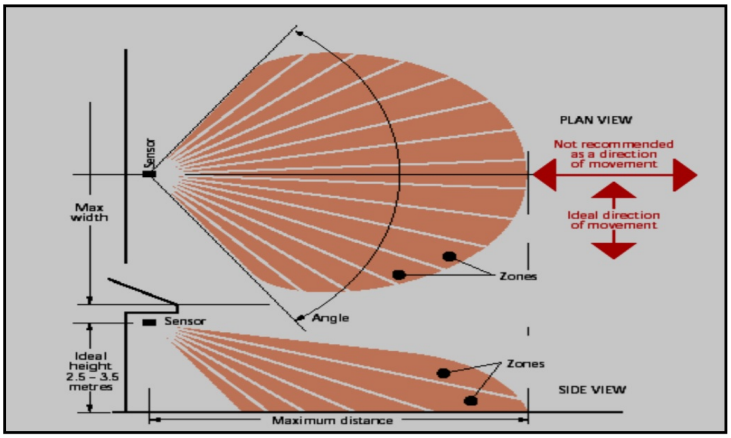

Figura 3.7: Campo de visão

Sensor de gás - MQ-2

O Sensor de Gás (MQ2) é utilizado para detecção de vazamento de gás (residencial e industrial). É apropriado para detectar H2, GLP, CH4, CO, álcool, fumo ou propano. Devido à alta sensibilidade e curto tempo de resposta, a medição pode ser realizada de maneira quase imediata. Este sensor apresenta baixo custo e é adequado para diferentes aplicações de monitoramento, incluindo a aplicação proposta por este projeto. O MQ-2 é exibido na Figura 3.8.

Figura 3.8: Sensor de gás MQ-2.

Pseudocódigo

A seguir é apresentado o pseudocódigo implantado no agente coletor.

Algoritmo 1: Agente Coletor

```
Entrada: ID, Interesse
início
    registra (Interesse);
    repita
        coleta e armazena dados de sensores;
        recebe Mensagem da rede;
        if Interesse == Mensagem(Interesse) then
            Envia dados armazenados temporariamente;
            Limpa o cache;
    até;
```

O agente coletor registra um interesse que o caracteriza como um dispositivo provedor de dados. Em seguida, o dispositivo fica constantemente colhendo dados dos sensores e os armazenando num *buffer* temporário. Simultaneamente, o agente fica aguardando uma solicitação para realizar o envio das informações armazenadas neste *buffer*. Os dados são removidos do *buffer* do dispositivo logo após terem sido enviados ao solicitante.

O tempo de coleta de informação, através dos sensores, pode ser diferente para cada sensor instalado no dispositivo Raspberry. Estas configurações são especificadas pelo desenvolvedor no momento da implementação do sistema. Por exemplo, o sensor de temperatura e umidade podem conter um intervalo de cinco minutos entre cada leitura, enquanto que, o sensor de fumaça fica constantemente realizando leituras com finalidade de identificar alguma anomalia no ambiente.

Estrutura de dados

As leituras provenientes dos sensores são armazenadas temporariamente no próprio dispositivo para posteriormente serem enviadas através da rede no momento de uma requisição do agente Mensageiro. A RadNet opera sobre o protocolo Ethernet que apresenta uma Unidade Máxima de Transmissão (*Maximum Transmission Unit* – MTU) de 1500 bytes, de modo, a limitar o tamanho da mensagem a ser transferida. Assim, foi necessário implementar um mecanismo para armazenar leituras em arquivos separados e de tamanho limitado. Cada leitura possui um tamanho de 200 *bytes*.

Para armazenamento e troca de mensagens foi adotado o formato JSON, que possui como principais vantagens sintaxe simples e de fácil compreensão. O formato de registro de medições é gerado a partir da estrutura de dicionários do Python

onde a chave e valor representam respectivamente o atributo e conteúdo do sensor. A seguir é exibido o formato das leituras utilizadas pelo Sistema Hermes.

```
{
  "value": 30,
  "location": [-22.588484, -43.284468],
  "date": 1561192101.66541,
  "sensor_model": "DHT22",
  "id": "id://inmetro/dmtic/lainf/escritorio/001",
  "physical_quantity": "TEMPERATURA"
}
```

Nesta estrutura de dados, o atributo *value* representa o valor de leitura gerado pelo sensor. O atributo *location* é referente à geolocalização de onde o dispositivo está alocado. O *date* registra o tempo em que a determinada leitura foi gerada. O *sensor_model* é referente ao modelo do sensor que gerou a leitura. O id identifica o dispositivo. Por fim, o atributo *physical_quantity* diz respeito a qual propriedade física está sendo gerada (ex. temperatura).

3.2.4 Agente Mensageiro

O agente Mensageiro é responsável por requisitar e coletar os dados produzidos pelo gente Coletor. O Mensageiro é representado pela classe *Messenger*, como foi apresentado na Figura 3.2. Esta classe é implementada num dispositivo *Raspberry Pi*. Para a finalidade específica deste trabalho, foi utilizado um Drone como um dispositivo portador de dados. A Figura 3.9 apresenta os elementos que compõem o agente mensageiro.

Figura 3.9: Drone Phantom e Dispositivo Raspberry acoplado com bateria.

- Drone Phantom 4 Pro plus;
- *Placa Raspberry Pi* 3, modelo B+, 1GB de RAM;
- *Power Bank Samsung.*

Drone Phantom 4 Pro Plus

O Drone é responsável por sobrevoar as áreas de interesse, onde os dispositivos coletores estão instalados. Devido à utilização do Drone em conjunto com o protocolo RadNet a comunicação entre os agentes coletores e servidores pode ser efetuada sem a necessidade de instalação de infraestrutura de rede, como por exemplo cabos ou pontos de acesso.

A Tabela 3.1 lista as especificações do Drone.

Peso (incluindo bateria e hélices)	1388g
Velocidade máxima	45mph (72 km/h) (modo S) — 36mph (58 km/h) (modo A) — 31mph (50 km/h) (modo P)
Teto Máximo de Serviço Acima do Nível do Mar	19685 pés (6000 m)
Tempo de Voo Máximo	Aprox. 30 minutos
Sistemas de Posicionamento por Satélite	GPS / GLONASS

Tabela 3.1: Especificações do Drone Phantom 4.

O Drone não possui um sistema operacional (SO), ou mesmo plataforma de software, disponível para instalação do Sistema Hermes e do protocolo RadNet. Consequentemente, foi necessário anexar um dispositivo Raspberry ao Drone. Porém, por ser uma carga a mais, este fator influenciou negativamente em sua autonomia, reduzindo seu tempo de voo em aproximadamente 20 minutos e dificultando a estabilização no ar. O Raspberry Pi juntamente com a bateria (*Power Bank*) contêm 350g. Deste modo, é necessário utilizar uma fonte de energia com peso mais adequado para conservar o tempo de autonomia do Drone, considerando trabalhos futuros.

Pseudocódigo

A seguir é apresentado o pseudocódigo implantado no agente Mensageiro.

Algoritmo 2: Agente Mensageiro

```
Entrada: ID, Interesse
início
    registra (Interesse);
    Envia Mensagem de requisição para coletor ;
    recebe Mensagem da rede;
    if buffer != null then
        Envia Mensagem de descoberta de servidor
    if Mensagem == coletor then
        Recebe dados do coletor;
        armazena dados temporariamente;
    if Mensagem == servidor then
        envia dados armazenados para servidor;
        limpar o cache;
```

O agente Mensageiro registra um interesse que o classifica como um dispositivo encarregado por transportar dados. Este agente envia dois tipos de mensagem: Requisição de dados para o coletor e Descoberta de servidor. A primeira mensagem é necessária para identificar se existe algum dispositivo coletor no raio de alcance da antena da rádio do *Raspberry Pi* que está acoplado ao drone. Caso exista, a troca de mensagens entre os agentes Coletor e Mensageiro é iniciada. O segundo tipo de mensagem é utilizado para identificar se existe algum agente registrado como servidor no alcance da antena. Este tipo de mensagem é enviado apenas quando existem dados armazenados no *buffer* do dispositivo, caso contrário, o dispositivo mensageiro não busca por um servidor disponível.

Caso exista algum dispositivo registrado como servidor, os dados armazenados em *buffer* são transferidos diretamente para o servidor através de seu prefixo (Seção 2.4.1).

3.2.5 Agente Servidor

O agente Servidor é responsável pelo armazenamento e análise dos dados recebidos do Mensageiro, de modo a operar como um repositório de dados. Todos os dados ficam disponíveis para outros serviços, aplicações e banco de dados. Toda as informação recebidas são armazenadas utilizando o formato JSON apresentado em 3.2.3.

Pseudocódigo

A seguir é apresentado o pseudocódigo implantado no agente Servidor.

Algoritmo 3: Agente Servidor

```
Entrada: ID, Interesse
início
    registra (Interesse);
    recebe Mensagem da rede;
    if Mensagem == 'DISCOVER' then
        envia mensagem de identificação do servidor
    else
        persiste dados no servidor;
```

O agente Servidor registra o interesse que o identifica como um servidor. Em seguida, o dispositivo fica aguardando uma mensagem da rede. De acordo com o conteúdo da mensagem duas ações podem ser tomadas: enviar mensagem que informa a disponibilidade do servidor ou persistir dados. Antes de iniciar a transmissão dos dados, o Mensageiro precisa identificar se existe algum dispositivo servidor capaz de armazenar as informações. Assim, o Mensageiro envia mensagens para descoberta de servidor, no qual contém 'DISCOVER' como conteúdo. Quando o servidor recebe uma mensagem com este conteúdo, ele envia uma mensagem de resposta que é utilizada pelo Mensageiro para iniciar a transferência dos dados.

3.3 Exemplo de comunicação no Projeto Hermes

A Figura 3.10 ilustra o procedimento de comunicação entre os agentes Coletor, Mensageiro e Servidor. Na figura, o Coletor possui o prefixo *Pc* e o interesse *Ic* = *app-collector://request*. O Mensageiro contém o prefixo *Pm* e interesse *Im* = *app-messenger://data*. Finalmente, o Servidor tem o prefixo *Ps* e interesse *Is* = *app-server://store*. Portanto, o interesse de cada agente indica a sua função no projeto.

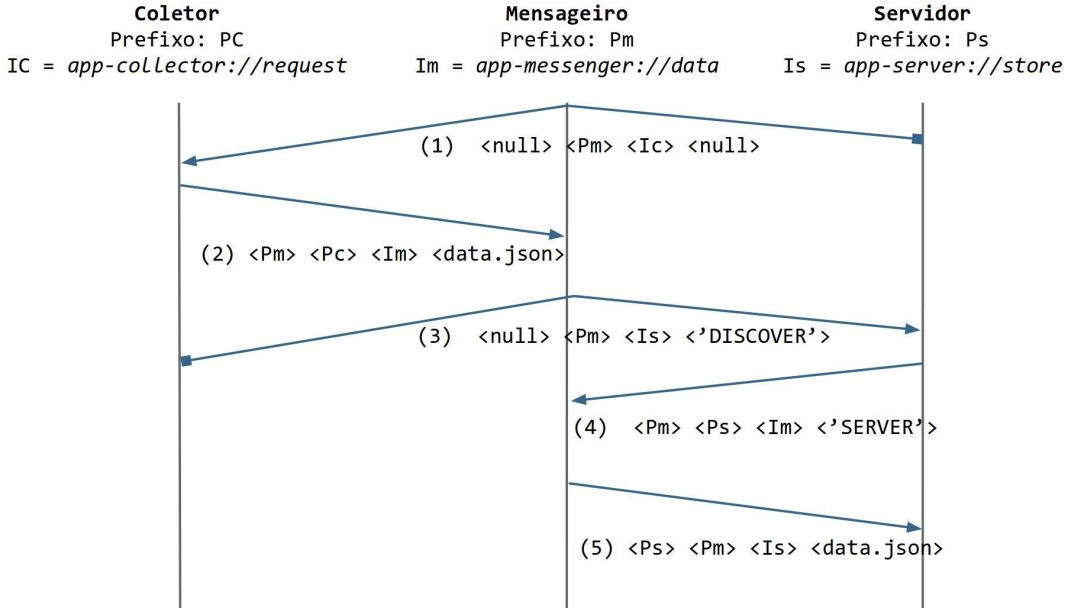

Figura 3.10: Exemplo de comunicação entre agentes.

A seguir é descrito o procedimento de comunicação.

1. O mensageiro elabora o cabeçalho da mensagem com o prefixo de destino *null*, indicando uma mensagem *broadcast* (Seção 2.4.1), prefixo de origem *Pm*, interesse *Ic = app-collector://request* e *payload null*. Em seguida, a mensagem é enviada pelo Mensageiro através do protocolo RadNet. A mensagem é descartada pelo Servidor por não haver correspondência de interesse;

2. O Coletor, em contrapartida, recebe a mensagem do Mensageiro, pois existe correspondência de interesse. Logo, o Coletor monta a mensagem com prefixo de destino *Pm*, indicando que o destino final é o Mensageiro, prefixo de origem *Pc* e interesse *Im*. Neste caso, o *payload* da mensagem é composto pelas leituras do sensoriamento de ambiente. Por fim, o Mensageiro recebe e armazena temporariamente os dados de leitura do Coletor.

3. Dado que o Mensageiro possui dados armazenados, é montado um pacote para identificar a presença de algum servidor disponível. Neste sentido, é montada e enviada a mensagem com interesse *Is* e *payload* 'DISCOVER' que indica ao servidor a intenção do Mensageiro iniciar o envio dos dados armazenados. Esta mensagem é descartada pelo coletor por não haver correspondência de interesse.

4. O servidor recebe e processa a mensagem, dado a correspondência de interesse. Logo, o servidor envia uma mensagem com *payload* 'SERVER' diretamente ao

Mensageiro por meio do prefixo de destino *Pm*. O *payload* indica ao Mensageiro que existe um servidor disponível para receber os dados.

5. Finalmente, o Mensageiro envia os dados armazenados diretamente para o servidor, através do prefixo *Ps* que é responsável por persistir estas informações numa base de dados.

É interessante observar nesta abordagem que o agente Servidor se encontra em estado de *stand by*, sem emitir qualquer mensagem, permitindo que não seja descoberto até que outro dispositivo envie uma mensagem com o interesse registrado no agente Servidor. Este modelo foi escolhido para manter o servidor mais "oculto"com a finalidade de torná-lo um repositório mais seguro.

Capítulo 4

Experimentos e Resultados

Este capítulo apresenta a avaliação experimental realizada sobre o sistema de monitoramento proposto neste trabalho. A Seção 4.1 introduz o objetivo da avaliação experimental. A Seção 4.2 mostra as configurações base para os cenários de teste. A Seção 4.3 discursa sobre os experimentos realizados sobre o sistema de monitoramento.

4.1 Objetivo

O objetivo da avaliação experimental foi testar, analisar e avaliar o desempenho da aplicação de monitoramento proposta neste trabalho com a finalidade de validar sua aplicabilidade num ambiente sem infraestrutura de redes previamente estabelecida. Como apresentado em [Cavilla et al. 2004], os resultados obtidos por meio de uma avaliação experimental simulada não iria considerar diversos aspectos físicos importantes como localização, obstáculos, ruídos eletromagnéticos e reflexão do sinal. Portanto, para este trabalho foi realizada uma implementação prática da aplicação de monitoramento juntamente com uma avaliação num cenário real.

4.2 Raspberry Pi e Sistema Operacional

Para execução de todos os cenários de testes foi utilizado o Raspberyy Pi 3 juntamento com o sistema operacional Raspbian, baseado no *kernel* do Linux. O Raspberry foi configurado para operar no modo de comunicação *ad hoc* utilizando uma rede sem fio IEEE 802.11 [Committee et al. 2007], assim foi possível prover comunicação entre os dispositivos independente da rede local. Todos os experimentos foram executados no campus do Instituto Nacional de Metrologia, Qualidade e Tec-

nologia (Inmetro). Os procedimentos para instalação e configuração do Raspberry estão descritos no Apêndice A.

4.3 Experimentos

Os experimentos tiveram como objetivo validar a aplicação considerando parâmetros específicos de cada cenário de teste. Para cada cenário de testes são apresentadas as configurações, resultados e discussões.

4.3.1 Experimento 1 - Testes de transmissão de dados mantendo a distância fixa

Neste cenário, os testes foram executados em um ambiente controlado, dentro do laboratório de informática (Lainf) como exibido na Figura 4.1, a comunicação foi realizada entre dois dispositivos Raspberry considerando uma distância fixa.

Foram analisados os seguintes parâmetros:

- **Quantidade de mensagens:** Este parâmetro diz respeito à quantidade de mensagens enviadas e recebidas entre os dispositivos;
- **Tamanho das mensagens:** Tamanho em *bytes* dos dados transferidos;
- **Integridade dos dados transferidos:** Quantidade de dados íntegros recebidos entre os dispositivos.

Estes parâmetros foram obtidos enviando 10, 100,1000 mensagens variando seu tamanho em 300, 500, 1000 bytes. Salientando que, como exposto na Seção 3.2.3 do Capítulo 3, o protocolo RadNet apresenta uma limitação 1500 bytes em relação ao tamanho da mensagem, pois este protocolo opera sobre o protocolo Ethernet. Para validar os experimentos, este teste foi executado 20 vezes.

Configuração e Ambiente de teste

A Figura 4.1 apresenta o ambiente onde este experimento foi executado. Os dispositivos Raspberry foram posicionados de modo a ficarem a uma distância de aproximadamente 9m. Esta foi a distância padrão considerada entre os dispositivos para este caso de teste. O único obstáculo entre os dispositivos foi uma parede modular de madeira que não se apresentou como um obstáculo significativo no tocante à transmissão de mensagens. Todas as medições apresentadas na Figura 4.1 estão na escala de metros.

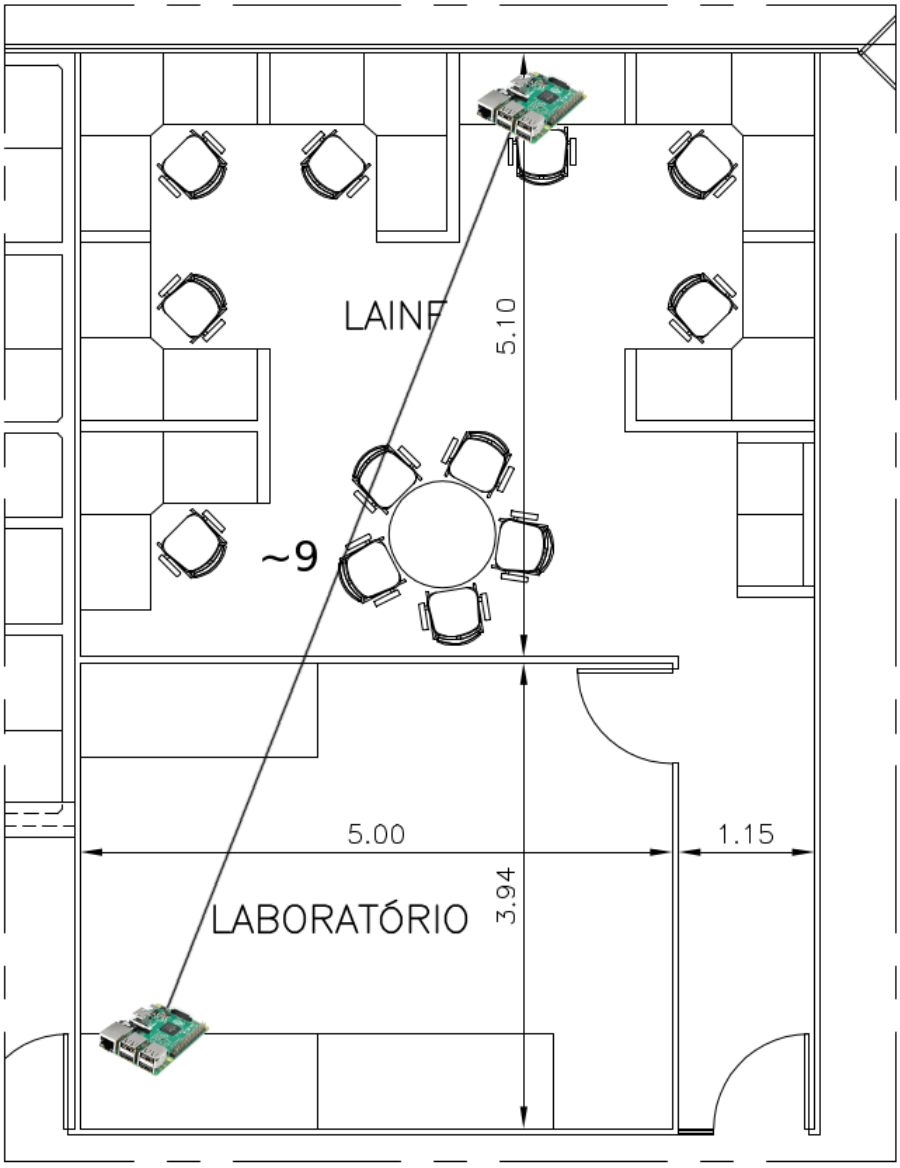

Figura 4.1: Ambiente de testes referente ao primeiro cenário de teste (medições em metros).

Resultados e discussões

A Tabela 4.1 exibe os resultados obtidos a partir da execução dos testes de transmissão considerando uma distância fixa. Devido ao fato dos testes terem sido realizados dentro de um ambiente controlado (cenário ideal), com pouco obstáculos e ruídos, o número de recepção de mensagens foi de 100%, considerando os parâmetros apresentados anteriormente. Portanto, é possível concluir que no dado cenário o Sistema Hermes é capaz de transmitir uma quantidade significativa de informações de sensoriamento sem apresentar perdas.

Tamanho do mensagem	Quantidade de mensagens	Média	Desvio padrão	Percentual
300 Bytes	10	10	0,0	100%
	100	100	0,0	100%
	1000	1000	0,0	100%
500 Bytes	10	10	0,0	100%
	100	100	0,0	100%
	1000	1000	0,0	100%
1 kilobytes	10	10	0,0	100%
	100	100	0,0	100%
	1000	1000	0,0	100%

Tabela 4.1: Resultados do primeiro cenário de teste.

4.3.2 Experimento 2 - Testes de transmissão de dados variando a distância entre os dispositivos

Neste experimento foi verificada a capacidade de comunicação entre os dispositivos variando a distância entre estes. O objetivo foi identificar a distância ideal para realizar a comunicação entre o agente mensageiro e coletor de modo a reduzir a perda de mensagens no momento da transmissão de dados. Este experimento foi necessário, visto que é preciso considerar que o dispositivo mensageiro possui capacidade de mobilidade e deve funcionar dentro do alcance do sinal de rádio.

Configuração e Ambiente de teste

Como indicado na Figura 4.2, foram consideradas as distâncias horizontais de 5, 10, 15, 20 e 25 metros sem obstáculos entre os dispositivos. Estas distâncias foram selecionadas considerando as especificações técnicas da antena de rádio do dispositivo Raspberry Pi 3. Neste cenário, a aplicação foi testada num ambiente *outdoor*, ou seja, fora do Laboratório de informática (Lainf). Para cada distância indicada foi transferido 100 arquivos de 300 *bytes*. Cada teste foi executado 20 vezes a fim validar os experimentos.

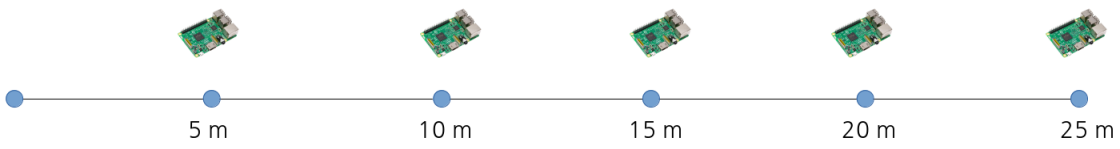

Figura 4.2: Ambiente de testes referente ao segundo cenário de teste.

Resultados e discussões

A Figura 4.3 e Tabela 4.2 apresentam os resultados do segundo cenário de teste proposto para a avaliação da aplicação. Como indicado no gráfico, na distância de 5 metros houve uma alta taxa de recepção das mensagens, de modo semelhante ao primeiro cenário de teste como apresentado na Seção 4.3.1.Tanto a proximidade quanto o baixo nível de ruído contribuíram para a alta taxa de recepção das mensagens transferidas. Os resultados entre 10 e 20 metros apresentaram poucas perdas das mensagens enviadas. É possível observar que em 20 metros houve menos perdas do que em 15 metros, isso ocorreu devido ao fator não determinístico do meio físico de transmissão das mensagens [Maróti et al. 2004]. Em 25 metros houve uma queda significativa na quantidade de mensagens íntegras recebidas. Esta queda é justificada pela distância entre o emissor/fonte e receptor das mensagens. Como o protocolo RadNet utiliza o protocolo UDP [Postel et al. 1980] não houve retransmissões das mensagens [Dutra et al. 2012].

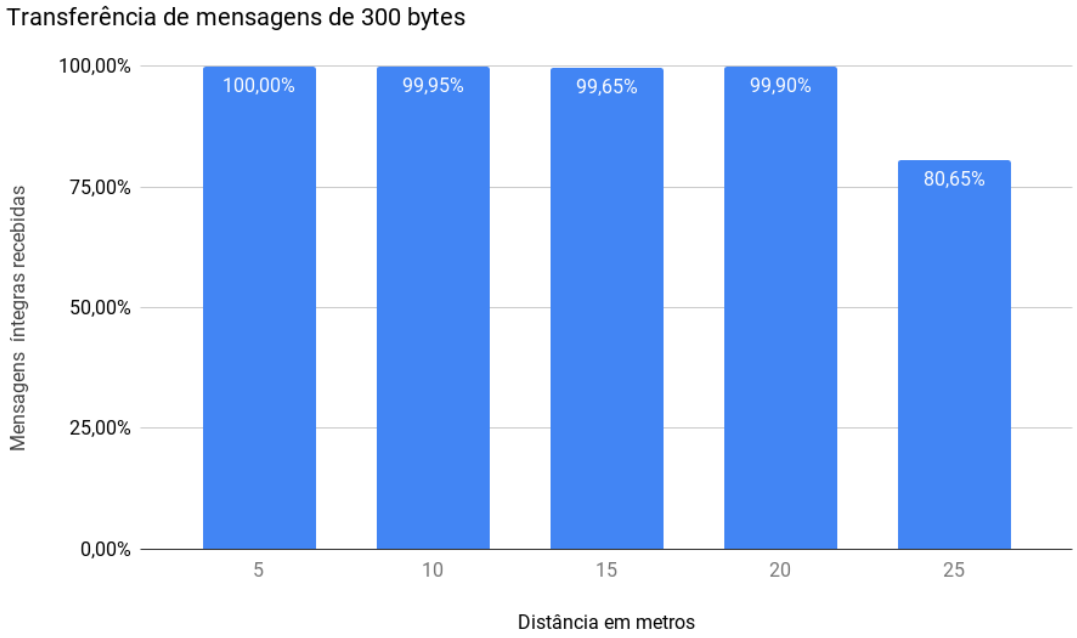

Figura 4.3: Resultados do segundo cenário de teste.

Quantidade de mensagens	Distância (metros)	Média	Desvio padrão	percentual
100	5	100	0,00	100%
	10	99,95	0,22	99,95%
	15	99,65	1,57	99,65%
	20	99,90	0,45	99,90%
	25	80,65	12,10	80,65%

Tabela 4.2: Resultados do segundo cenário de teste

4.3.3 Experimento 3 - Testes de transmissão de dados com dispositivo mensageiro em movimento

Este experimento teve como objetivo principal demonstrar que o sistema de sensoriamento remoto desenvolvido neste trabalho está apto a operar em um cenário onde o dispositivo móvel está em constante movimento sobre um ambiente onde a infraestrutura de redes não está disponível. Como especificado na Seção 3.1, a aplicação proposta neste trabalho pode ser transportada por um humano através de um dispositivo *mobile* ou, como foi proposto neste trabalho, carregado por Drone. Portanto, para este cenário de teste foi considerado os seguintes modos de transporte do Raspberry:

- **Transportado por um humano:** Este parâmetro de teste é referente ao dispositivo sendo carregado por uma pessoa;
- **Transportado por um Drone:** Neste modo, o dispositivo Raspberry é acoplado ao Drone como foi apresentado na Seção 3.2.4.

Vale ressaltar que o Raspberry não contém fonte de energia própria o que torna necessário a utilização de uma bateria externa para prover alimentação durante o deslocamento do dispositivo.

Configuração do ambiente de teste

Para execução deste caso de teste, um computador Raspberry Pi, configurado como um agente Coletor, foi posicionado e instalado no prédio de número 4. Em contrapartida, um segundo Raspberry foi configurado e aplicado como agente Mensageiro no prédio 2. A Figura 4.4 indica o posicionamento dos equipamentos. Os dispositivos foram posicionados de maneira a impedir o alcance do sinal sem fio da antena entre os dispositivos. Assim, a única forma de comunicação entre estes se deu através do deslocamento do agente Mensageiro em direção ao coletor.

Diante desta configuração, o dispositivo Raspberry, acoplado a um portador, se locomoveu do prédio 2 em direção ao prédio 4, considerando uma velocidade média constante de aproximadamente 2 m/s e uma altura de aproximadamente 2 metros. Ao chegar ao alcance do sinal de rádio da antena do agente coletor, o dispositivo Mensageiro ficou estacionado em aproximadamente 15 metros de distância do agente Coletor durante 2 minutos. Este tempo foi considerado para transmissão completa dos dados armazenados no equipamento. Como este cenário de teste valida a aplicação num cenário mais próximo ao real, os testes consideraram uma transmissão de 100 mensagens de 200 bytes. Este tamanho de mensagem foi definido pois é a quantidade de *bytes* ocupada pela estrutura de dados apresentada na Seção 3.2.3.

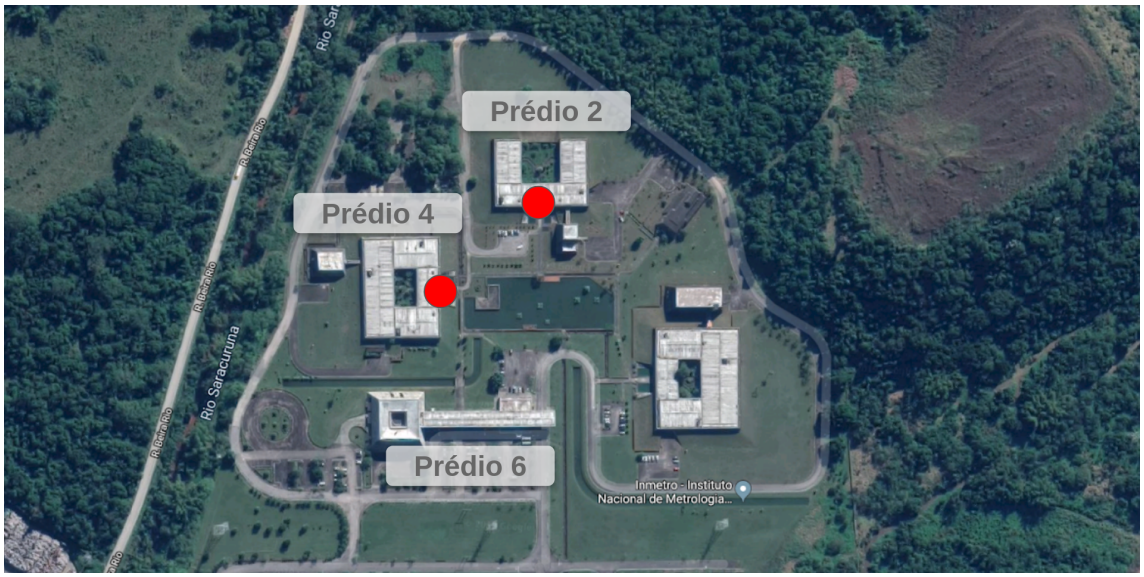

Figura 4.4: Cenário de teste.

Para calcular a velocidade de movimento do Raspberry transportado por humano foi utilizado o aplicativo *SpeedMeter* disponível para *download* na *Play Store*. A Figura 4.5 apresenta a interface do aplicativo juntamente com a medição em m/s, no caso deste cenário de teste, a velocidade foi 2m/s. A Figura 4.6 apresenta a interface do controle do Drone, demonstrando a velocidade horizontal (H.S) e altitude (H), assim, é possível observar que o Drone foi deslocado com a velocidade horizontal de aproximadamente 2m/s e altura 2 metros.

Figura 4.5: Interface do aplicativo *speedMeter*.

Figura 4.6: Interface do controle do Drone.

Testes preliminares indicaram uma possível interferência gerada pelos motores de hélice do Drone. Portanto, para melhor compreensão do problema foi elaborada uma blindagem sobre o dispositivo Raspberry e sobre a bateria externa. A blindagem foi elaborada seguindo os conceitos da Gaiola de Faraday[1]. A gaiola foi implementada utilizando um material condutor e através de um aterramento por meio do pino GND do dispositivo Raspberry. Como o dispositivo ficou totalmente isolado pela blindagem foi necessário a utilização de uma antena Wi-Fi externa para permitir a comunicação entre os dispositivos. Vale destacar que apenas o Raspberry anexado ao Drone foi blindado. A Figura 4.7 apresenta o dispositivo Raspberry coberto com a blindagem.

Figura 4.7: Raspberry com blindagem e antena externa.

[1]A gaiola de Faraday [Kraus 1992] é uma blindagem elétrica desenvolvida para impedir o ruído eletromagnético. Uma onda eletromagnética é composta por campos elétricos e magnéticos oscilantes, gerado por motores, ondas de rádio e televisão, ou qualquer equipamento que utilize corrente alternada como fonte de energia.

Resultados e discussões

A Figura 4.8 e Tabela 4.3 apresentam os resultados obtidos a partir da execução dos experimentos. É possível observar que a aplicação apresenta uma maior taxa de recepção quando o dispositivo Raspberry é carregado por uma pessoa. Neste cenário é possível notar que houve uma perda de 15,8% das mensagens transmitidas. Esta perda ocorreu em virtude de dois fatores principais: Influência do meio de transmissão e variação na distância no momento de transmissão das mensagens. O meio de transmissão Wi-Fi pode ser influenciado por conta de fatores físicos como refração e reflexão do sinal ou, inclusive, por interferências externas como outras antenas de rádio. Além disso, a comunicação entre os dispositivos, por definição da aplicação, é iniciada no momento em que o dispositivo Mensageiro entra na zona de alcance do sinal de rádio do dispositivo Coletor. Consequentemente, a probabilidade de perdas se torna maior visto que a comunicação não é iniciada considerando uma distância ideal. Esta questão pode ser solucionada através da utilização de serviços de geolocalização que permitem identificar a distância entre os equipamentos de modo a identificar a distância ideal para, só assim, inicializar a transmissão de mensagens.

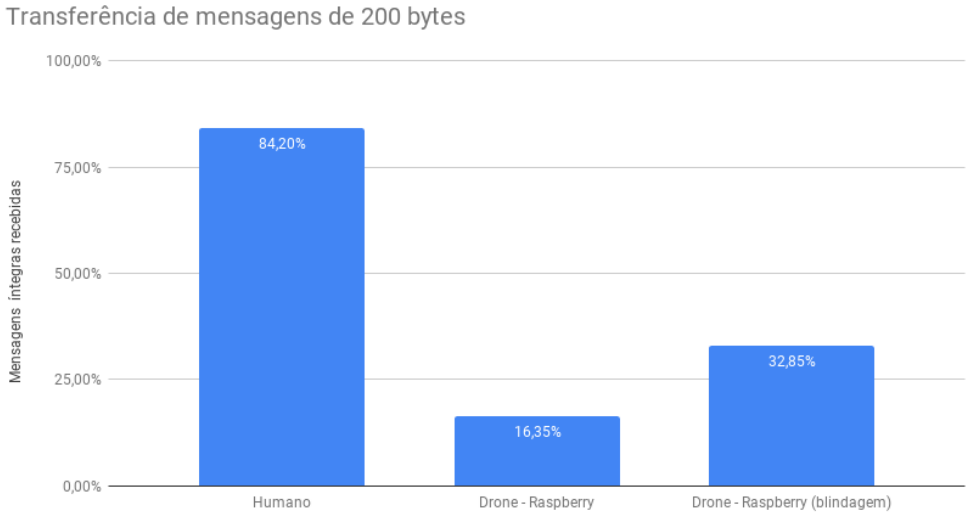

Figura 4.8: Resultados do terceiro cenário de teste.

Quantidade de mensagens	Portador	Média	Desvio Padrão	Percentual
100	Humano	84,20	10,43	84,20%
	Drone - Raspberry	16,35	11,39	16,35%
	Drone - Raspberry (blindagem)	32,85	10,78	32,85%

Tabela 4.3: Resultados do terceiro cenário de teste

É observado uma queda considerável na taxa de transmissão de mensagens no momento em que é utilizado o Drone, especificado na Seção 3.2.4, como portador do dispositivo Raspberry. A baixa taxa de recepção é justificada pela interferência eletromagnética causada pela rotação dos motores de hélice do Drone. No cenário em que o Raspberry está sendo carregado pelo Drone a perda na transmissão é de 83,35%. E depois, nota-se uma pouca melhora, perda de 67,35% de mensagens, quando a camada de blindagem foi adicionada ao Raspberry. A seguir é exposto a análise sobre o comportamento do espectro Wi-Fi diante da utilização do Drone.

4.3.4 Experimento 4 - Análise da influência dos elementos de comunicação do Drone

Este experimento foi especificado com a finalidade de analisar e identificar se as operações comunicação entre o Drone e seu respectivo controle remoto apresentam alguma influência sobre a transmissão de mensagens do Sistema Hermes.

Configuração do ambiente de teste

Para composição deste cenário de teste, foi reproduzido o roteiro de comunicação e movimentação do veículo autônomo especificado na Seção 4.3.3. Assim, este experimento reprisou as definições de testes já apresentadas, porém, foi considerado a variação dos canais e frequências de comunicação utilizadas pelos dispositivos (Drone e Raspberry Pi).

O Drone utilizado neste trabalho dispõe de oito canais de comunicação (entre os canais 13 e 20). O Drone, por padrão, é configurado para trocar os canais de forma dinâmica de acordo com a variação na utilização dos mesmos. Porém, ainda, é possível selecionar um canal específico que será utilizado para realizar a comunicação. A Figura 4.9 apresenta as configurações de seleção de canal, onde é exposto o canal que está sendo utilizado (*Current Channel*). São exibidos, também, o modo de comunicação (*Channel Mode*) e frequência utilizada para a comunicação.

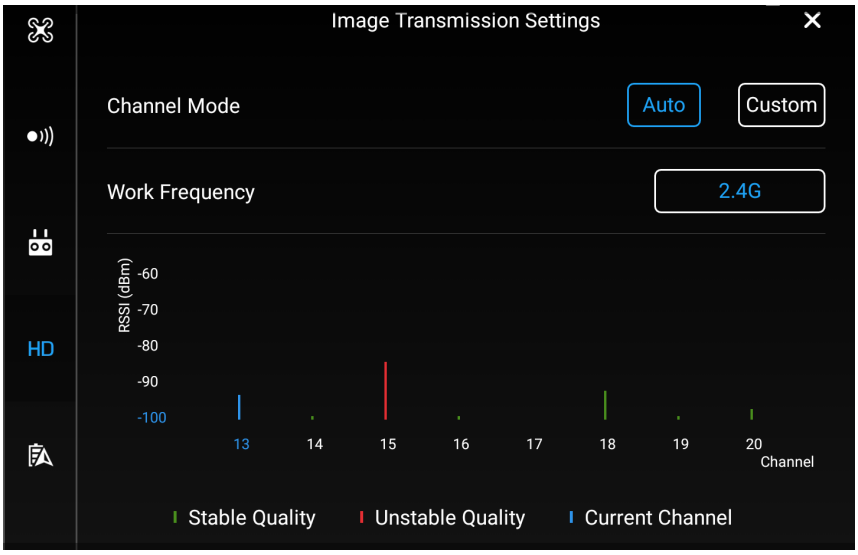

Figura 4.9: Canais e frequências disponíveis no Drone.

Além disso, O dispositivo Raspberry também fornece mecanismos para visualizar e configurar os canais e as faixas de frequência disponíveis em sua interface de rede. A Figura 4.10 exibe os canais e frequências disponíveis para a comunicação do dispositivo Raspberry Pi utilizado para execução dos testes. Neste cenário, o Raspberry possui 32 canais disponíveis.

```
wlan0     32 channels in total; available frequencies :
          Channel 01 : 2.412 GHz
          Channel 02 : 2.417 GHz
          Channel 03 : 2.422 GHz
          Channel 04 : 2.427 GHz
          Channel 05 : 2.432 GHz
          Channel 06 : 2.437 GHz
          Channel 07 : 2.442 GHz
          Channel 08 : 2.447 GHz
          Channel 09 : 2.452 GHz
          Channel 10 : 2.457 GHz
          Channel 11 : 2.462 GHz
          Channel 12 : 2.467 GHz
          Channel 13 : 2.472 GHz
          Channel 36 : 5.18 GHz
          Channel 40 : 5.2 GHz
          Channel 44 : 5.22 GHz
          Channel 48 : 5.24 GHz
          Channel 52 : 5.26 GHz
          Channel 56 : 5.28 GHz
          Channel 60 : 5.3 GHz
          Channel 64 : 5.32 GHz
          Channel 100 : 5.5 GHz
          Channel 104 : 5.52 GHz
          Channel 108 : 5.54 GHz
          Channel 112 : 5.56 GHz
          Channel 116 : 5.58 GHz
          Channel 120 : 5.6 GHz
          Channel 124 : 5.62 GHz
          Channel 128 : 5.64 GHz
          Channel 132 : 5.66 GHz
          Channel 136 : 5.68 GHz
          Channel 140 : 5.7 GHz
          Current Frequency:5.765 GHz
```

Figura 4.10: Canais e frequências disponíveis no Raspberry Pi 3 Model B+.

Resultados e discussões

Após a reprodução dos testes, variando os canais e faixas de frequências (2.4Ghz e 2.5GHz) disponíveis nos dispositivo Raspberry e do Drone, não foi identificado melhora no tocante à transmissão das mensagens entre os dispositivos Mensageiro e Coletor. Os resultados da transmissão das mensagens são semelhantes aos resultados apontados na Seção 4.3.3, onde há uma notável queda na quantidade de mensagens recebidas quando o Drone é utilizado como transportador. Portanto, pode-se compreender que a variação nos canais e faixas de frequências utilizadas na comunicação entre o Drone e controle remoto não influenciou na transmissão das mensagens no Sistema Hermes.

4.3.5 Experimento 5 - Análise da influência do Drone sobre espectro Wi-Fi

Este experimento foi proposto com a finalidade de compreender a influência do Drone sobre a taxa de transmissão de mensagens do sistema de sensoriamento remoto apresentado neste trabalho de dissertação.

Configuração do ambiente de teste

Para execução deste caso de teste foi necessário a utilização de um analisador de espectro. O analisador de espectro é um instrumento eletrônico utilizado para observar as componentes harmônicas de sinais elétricos. Estes componentes podem ser de frequências e amplitudes diferentes espalhadas pelo espectro de frequências. Existem analisadores para a faixa de áudio e para sinais de rádio frequência [Deery 2007]. Neste trabalho foi utilizado o analisador de espectro para rádio de frequência, modelo *Spectrum Master* MS2724B, utilizando uma antena omnidirecional de 2,5DBi de ganho. Este modelo de analisador permite análise do espectro entre 9KHz até 20GHz o que o tornou adequado para realização do experimento em questão. As informações técnicas do analisador podem ser encontradas através do *link*: https://www.anritsu.com/en-us/test-measurement/products/ms2724b. A Figura 4.11 mostra a bancada de testes utilizada para execução dos experimentos.

A análise do espectro foi realizada com os parâmetros definidos a seguir:

- **Drone desligado:** Foi colhida uma amostra do cenário de espectro no momento em que o Drone estava desligado. Neste caso, o Drone estava completamente sem energia e fora de operação, ou seja, tanto o sistema do controle remoto quanto o Drone estavam desligados;

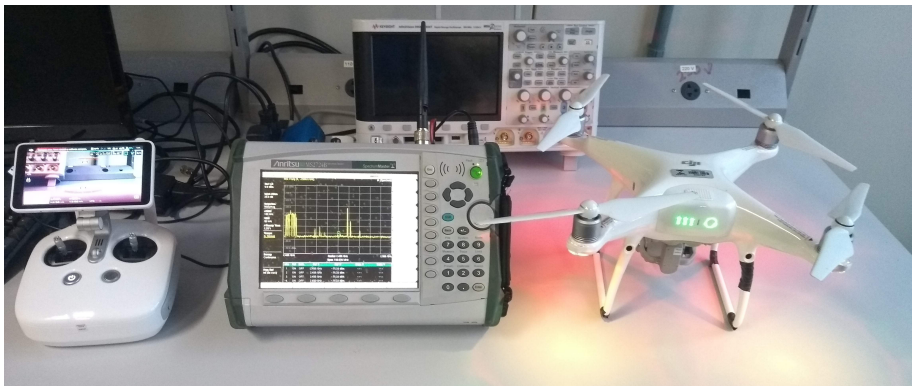

Figura 4.11: Bancada de teste.

- **Drone ligado (*Stand by*):** Neste parâmetro, o Drone e o controle remoto estavam ligados, porém sem operação de voo. As únicas informações de comunicação entre o controle e o Drone foram transmissão de imagem (em tempo real) e dados do estado atual do Drone, por exemplo, nível de bateria;
- **Drone com motores de hélices ligados:** Neste caso foi colhida uma amostra do espectro no momento em que o Drone estava com os motores de hélices ligados e em operação de voo.

Todas as amostras foram colhidas entre as frequências de 2.4GHz e 2.5GHz, pois esta é a faixa de frequência utilizada pelo Wi-Fi.

Resultados e discussões

A Figura 4.12 apresenta a varredura do espectro com Drone e controle desligados. As amostras foram colhidas entre a faixa de frequência não licenciada do WLAN 802.11 (*Wireless Local Area Network*) que utilizam 2.4Ghz a 2.5Ghz; em que há grande quantidade de fontes de interferência devido a popularização dos dispositivos Wi-Fi existente em nosso cotidiano, tais como: Roteadores, *Access Point* e antenas. No gráfico gerado pelo analisador de espectro é possível visualizar alguns sinais na faixa de frequência 2.400 GHz a 2.500 GHz. Os sinais relevantes plotado no gráfico mostra 5 sinais com amplitude destacados no retângulo. Estes sinais são gerados pelos roteadores do Lainf e laboratórios adjacentes.

Constatou-se que o ruído gerado pelos 4 motores do Drone, enquanto desligado, não geram nenhum tipo de espúrios que interferem no sistema de comunicação entre os dispositivos utilizados durante os testes.

A Figura 4.13 expõe o espectro no momento em que tanto o Drone quanto o controle estavam ligados. Neste estado o Drone fica em *Stand By* aguardando instruções para início das operações de voo. Observa-se que um novo sinal foi gerado, este sinal é referente a transmissão de vídeo em tempo real da câmera do Drone para o

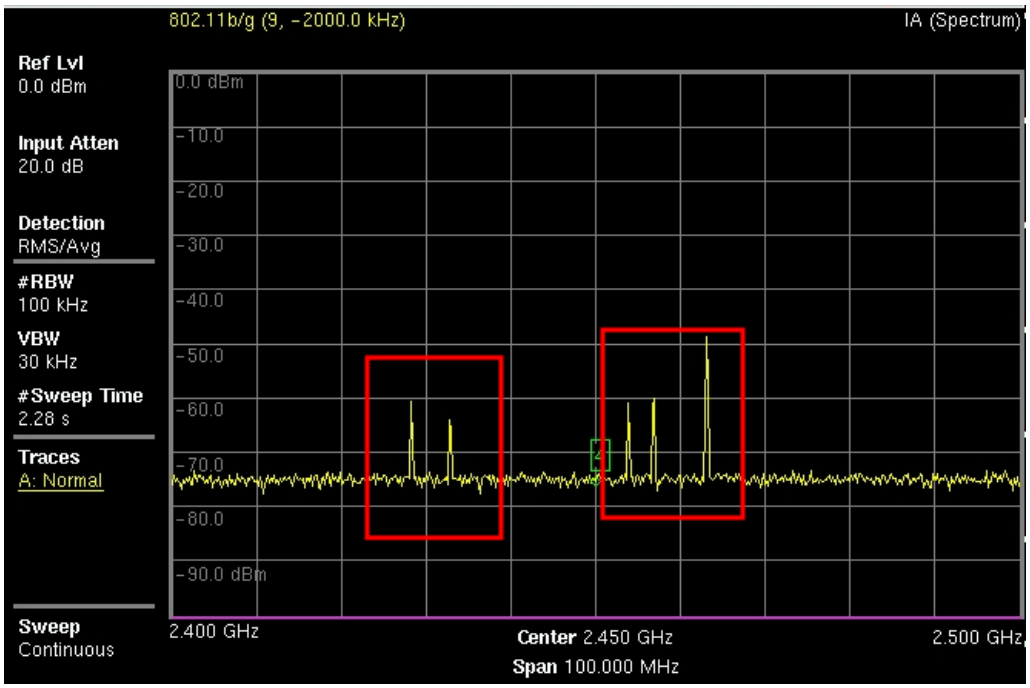

Figura 4.12: Amostra do espectro com o Drone desligado.

controle. Ainda assim, este sinal não representou uma interferência na comunicação sem fio entre as aplicações dos agentes Mensageiro e Coletor.

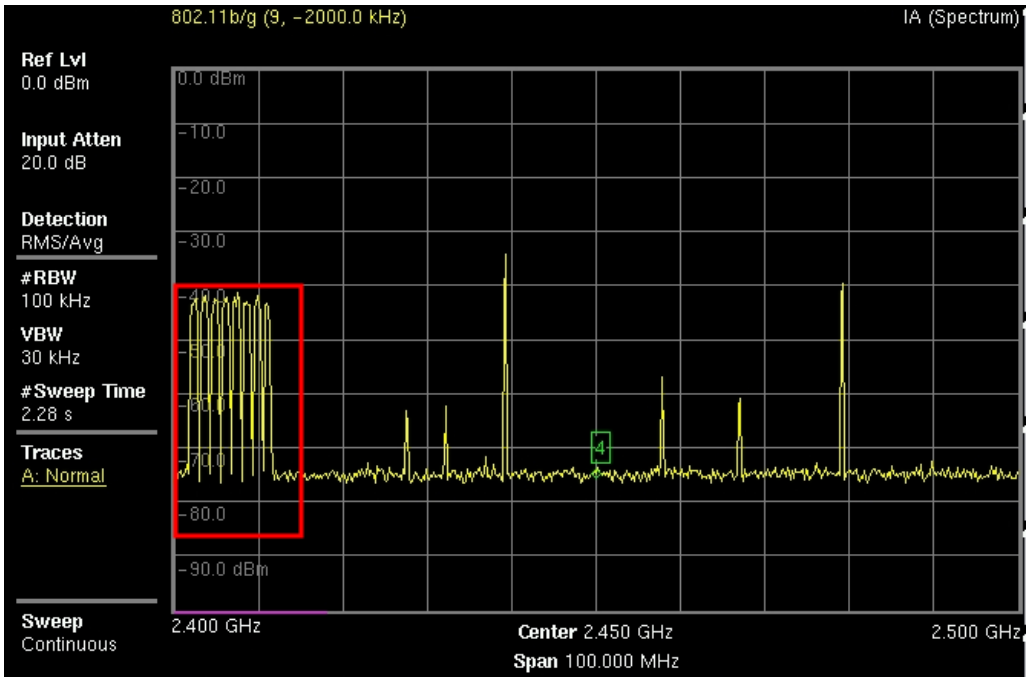

Figura 4.13: Amostra do espectro com o Drone ligado e em *Standy By*.

Finalmente, foi realizada uma análise sobre o espectro no momento em que os motores de hélices são acionados para iniciar as operações de voo. Como exibido na Figura 4.14. É possível notar uma pequena variação no sinal de transmissão de vídeo, porém, ainda assim, não foi identificado nenhuma anomalia ou outro sinal gerado por conta das operações de voo do Drone.

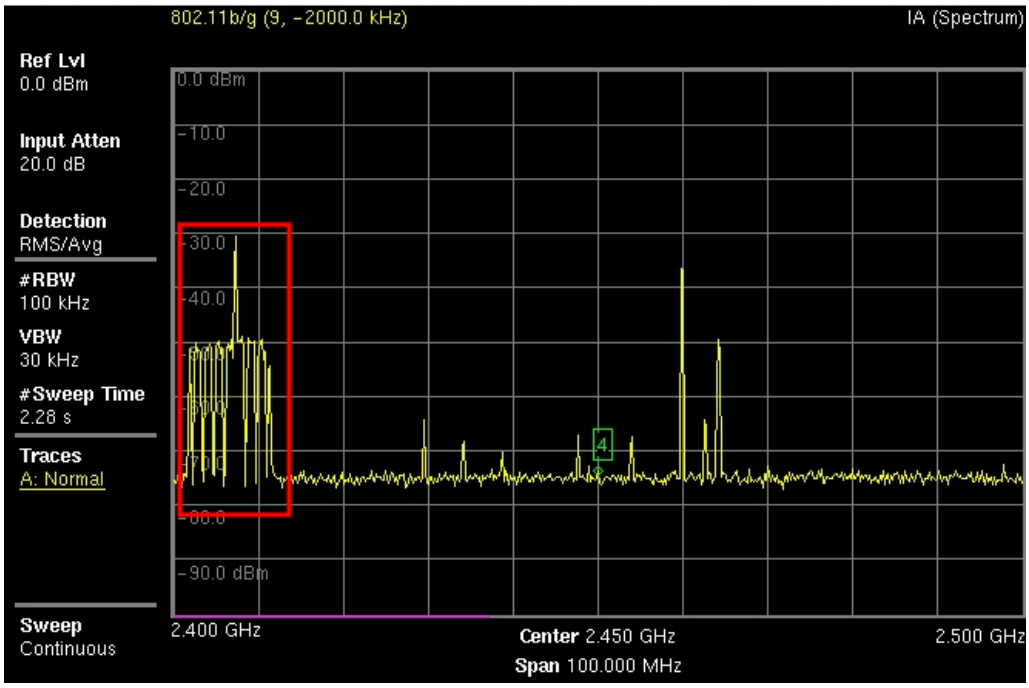

Figura 4.14: Amostra do espectro com os motores de hélices ligados.

Com base nas análises realizadas sobre o espectro, considerando diferentes parâmetros, não foi detectado nenhuma possibilidade de ruído, gerado pelos motores de hélice do Drone, na faixa de frequência do Wi-Fi. Foi condierado, com base nestes resultados e experimentos anteriores, que existe um interferência gerada pelos motores do hélice do Drone que influencia na eficiência de transmissão de mensagens no Sistema Hermes. Porém, esta é uma interferência indutiva sobre os componentes eletrônicos embarcados no placa Raspberry.

Capítulo 5

Conclusão

Este capítulo apresenta a conclusão deste trabalho. A Seção 5.1 tece as considerações finais e conclusões do trabalho. A Seção 5.2 aponta direcionamentos de trabalhos futuros.

5.1 Considerações Finais

Embora a pilha de protocolos TCP/IP sejam amplamente utilizada na Internet, existem abordagens alternativas que não se baseiam no endereçamento IP para prover comunicação. Neste trabalho, foi apresentado o Sistema Hermes, uma aplicação de sensoriamento remoto que utiliza um protocolo de redes oportunísticas centrada em interesse, denominada RadNet. Para esta finalidade, foi elaborada uma completa descrição do desenvolvimento do sistema de monitoramento, salientando o uso do protocolo de rede oportunística centrada em interesses como mecanismo de transmissão de dados em ambientes desafiadores. A aplicação proposta mostrou como essa rede pode operar em cenários onde não existem uma infraestrutura de redes pré-estabelecida e como podem ser enviadas informações de monitoramento caso a infraestrutura da rede local existente seja comprometida.

A aplicação de sensoriamento permitiu implementar o protocolo RadNet em uma nova arquitetura onde os agentes Coletor, Mensageiro e Servidor foram desenvolvidos especificamente para esse cenário de sensoriamento, utilizando um veículo aéreo não tripulado (Drone) no transporte de dados entre os pontos distantes. Contudo, a mesma arquitetura poderá ser adaptada, substituindo o Drone por outros dispositivos móveis, em cenários de monitoramento ambiental e prevenção de catástrofes, aplicações de Internet das Coisas voltadas para a área de saúde ou mobilidade de veículos, por exemplo.

Além da completa descrição do desenvolvimento da solução, foi realizada uma avaliação experimental sobre o sistema proposto com a finalidade de validar sua aplicabilidade em ambientes com pouca ou nenhuma comunicação. Os resultados

da avaliação experimental mostraram que a aplicação de sensoriamento é capaz de se comunicar em ambiente sem infraestrutura de redes previamente estabelecida. O dispositivo Raspberry sendo transportado por humano apresentou pouca perda das mensagens enviadas, tais perdas podem ser mitigadas implementando serviços de geolocalização ou utilizando antenas com um ganho maior. Por outro lado, quando transportado pelo Drone, proposto neste trabalho, foi notado uma queda no desempenho da aplicação em virtude do ruído gerado pelos motores de hélice do Drone. Análises sobre o espectro Wi-Fi indicaram que o ruído não afeta a faixa de frequência do Wi-Fi, portanto, foi considerado uma interferência por indução sobre os componentes internos do Raspberry.

5.2 Trabalhos Futuros

Apesar das contribuições apresentadas, estes trabalho possui algumas limitações no tocante ao desenvolvimento e implementação. Tais limitações representam possibilidade de extensão e melhoria deste trabalho. Portanto, o material já apresentado também oferece perspectivas de trabalhos futuros, listadas a seguir:

- Implementar o protocolo de rede oportunística centrada em Interesse (RadNet) em dispositivos de baixo consumo como arduino e ESP32;
- Desenvolver sistema para correlacionar dados e identificar eventos;
- Desenvolver rede em malha para propagação dos dados;
- Investigar e analisar as implementações de mecanismos de segurança adequados na coleta de dados, transmissão de mensagens criptografadas, técnicas de distribuição de chaves para assegurar a comunicação entre os nós da rede e a autenticação dos dispositivos utilizando o protocolo RadNet.

Referências Bibliográficas

[ISO 2016] (2016). Iso guide 30:2015.

[mqt 2019] (2019). Mqtt specification. `http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html`.

[Abidy et al. 2014] Abidy, Y., Saadallahy, B., Lahmadi, A., and Festor, O. (2014). Named data aggregation in wireless sensor networks. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–8. IEEE.

[Agrawal and Chauhan 2015] Agrawal, V. M. and Chauhan, H. (2015). An overview of security issues in mobile ad hoc networks. *International Journal of Computer Engineering and Sciences*, 1(1):9–17.

[Ahlgren et al. 2012] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36.

[Ain et al. 2009] Ain, M., Trossen, D., Nikander, P., Tarkoma, S., Visala, K., Rimey, K., Burbridge, T., Rajahalme, J., Tuononen, J., Jokela, P., et al. (2009). D2. 3–architecture definition, component descriptions, and requirements. *Deliverable, PSIRP 7th FP EU-funded project*, 11.

[Al-Fuqaha et al. 2015] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 17(4):2347–2376.

[Amadeo et al. 2016] Amadeo, M., Campolo, C., Quevedo, J., Corujo, D., Molinaro, A., Iera, A., Aguiar, R. L., and Vasilakos, A. V. (2016). Information-centric networking for the internet of things: challenges and opportunities. *IEEE Network*, 30(2):92–100.

[Amadeo and Molinaro 2011] Amadeo, M. and Molinaro, A. (2011). Chanet: A content-centric architecture for ieee 802.11 manets. In *2011 International Conference on the Network of the Future*, pages 122–127. IEEE.

[Amadeo et al. 2013] Amadeo, M., Molinaro, A., and Ruggeri, G. (2013). E-chanet: Routing, forwarding and transport in information-centric multihop wireless networks. *Computer communications*, 36(7):792–803.

[Arshad et al. 2018] Arshad, S., Azam, M. A., Rehmani, M. H., and Loo, J. (2018). Recent advances in information-centric networking-based internet of things (icn-iot). *IEEE Internet of Things Journal*, 6(2):2128–2158.

[Ashton et al. 2009] Ashton, K. et al. (2009). That 'internet of things' thing. *RFID journal*, 22(7):97–114.

[Atzori et al. 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

[Bassi and Horn 2008] Bassi, A. and Horn, G. (2008). Internet of things in 2020: A roadmap for the future. *European Commission: Information Society and Media*, 22:97–114.

[Bellavista et al. 2013] Bellavista, P., Cardone, G., Corradi, A., and Foschini, L. (2013). Convergence of manet and wsn in iot urban scenarios. *IEEE Sensors Journal*, 13(10):3558–3567.

[Bogdan 2016] Bogdan, M. (2016). How to use the dht22 sensor for measuring temperature and humidity with the arduino board. *ACTA Universitatis Cibiniensis*, 68(1):22–25.

[Carzaniga et al. 2011] Carzaniga, A., Papalini, M., and Wolf, A. L. (2011). Content-based publish/subscribe networking and information-centric networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 56–61. ACM.

[Castillo-Effer et al. 2004] Castillo-Effer, M., Quintela, D. H., Moreno, W., Jordan, R., and Westhoff, W. (2004). Wireless sensor networks for flash-flood alerting. In *Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems, 2004.*, volume 1, pages 142–146. IEEE.

[Cavilla et al. 2004] Cavilla, A. L., Baron, G., Hart, T. E., Litty, L., and De Lara, E. (2004). Simplified simulation models for indoor manet evaluation are

not robust. In *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, pages 610–620. IEEE.

[Committee et al. 2007] Committee, I. C. S. L. S. et al. (2007). Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11ˆ*.

[Conti et al. 2015] Conti, M., Boldrini, C., Kanhere, S. S., Mingozzi, E., Pagani, E., Ruiz, P. M., and Younis, M. (2015). From manet to people-centric networking: Milestones and open research challenges. *Computer Communications*, 71:1–21.

[de Brito et al. 2012] de Brito, G. M., Velloso, P. B., and Moraes, I. M. (2012). Redes orientadas a conteúdo: Um novo paradigma para a internet. *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC*, 2012:211–264.

[de Castro Dutra 2012] de Castro Dutra, R. (2012). *REDES AD HOC CENTRADAS EM INTERESSES PARA AMBIENTES MOVEIS*. PhD thesis, Universidade Federal do Rio de Janeiro.

[Deery 2007] Deery, J. (2007). The real history of real-time spectrum analyzers. *Sound & vibration*, page 6.

[Del Campo et al. 2016] Del Campo, A., Gambi, E., Montanini, L., Perla, D., Raffaeli, L., and Spinsante, S. (2016). Mqtt in aal systems for home monitoring of people with dementia. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE.

[Dhurandher et al. 2013] Dhurandher, S. K., Sharma, D. K., Woungang, I., and Bhati, S. (2013). Routing protocols in infrastructure-less opportunistic networks. In *Routing in Opportunistic Networks*, pages 353–382. Springer.

[Dutra et al. 2012] Dutra, R. C., Moraes, H. F., and Amorim, C. L. (2012). Interest-centric mobile ad hoc networks. In *2012 IEEE 11th International Symposium on Network Computing and Applications*, pages 130–138. IEEE.

[Etefia and Zhang 2012] Etefia, B. and Zhang, L. (2012). Named data networking for military communication systems. In *2012 IEEE Aerospace Conference*, pages 1–7. IEEE.

[Eugster et al. 2003] Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, 35(2):114–131.

[Fall 2003] Fall, K. (2003). A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM.

[Fidler et al. 2005] Fidler, E., Jacobsen, H., Li, G., and Mankovski, S. (2005). Publish/subscribe system. *Feature Interactions in Telecommunications and Software Systems VIII*, page 12.

[Gia et al. 2015] Gia, T. N., Jiang, M., Rahmani, A.-M., Westerlund, T., Liljeberg, P., and Tenhunen, H. (2015). Fog computing in healthcare internet of things: A case study on ecg feature extraction. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 356–363. IEEE.

[Gonçalves et al. 2016] Gonçalves, F. B., França, F. M., and de Amorim, C. L. (2016). Interest-centric vehicular ad hoc network. In *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–10. IEEE.

[Hande et al. 2006] Hande, A., Polk, T., Walker, W., and Bhatia, D. (2006). Self-powered wireless sensor networks for remote patient monitoring in hospitals. *Sensors*, 6(9):1102–1117.

[Hudhajanto et al. 2018] Hudhajanto, R. P., Fahmi, N., Prayitno, E., et al. (2018). Real-time monitoring for environmental through wireless sensor network technology. In *2018 International Conference on Applied Engineering (ICAE)*, pages 1–5. IEEE.

[Jacobson et al. 2009] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. (2009). Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM.

[Juang et al. 2002] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. S., and Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking:

Design tradeoffs and early experiences with zebranet. *ACM SIGARCH Computer Architecture News*, 30(5):96–107.

[Khan et al. 2012] Khan, R., Khan, S. U., Zaheer, R., and Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *2012 10th international conference on frontiers of information technology*, pages 257–260. IEEE.

[Koponen et al. 2007] Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I. (2007). A data-oriented (and beyond) network architecture. *ACM SIGCOMM Computer Communication Review*, 37(4):181–192.

[Kraus 1992] Kraus, J. (1992). *Electromagnetics*. McGraw-Hill.

[Kurose and Ross] Kurose, J. F. and Ross, K. W. *COMPUTER NETWORKING A Top-Down Approach*.

[Li et al. 2006] Li, M., Wu, M.-Y., Li, Y., Cao, J., Huang, L., Deng, Q., Lin, X., Jiang, C., Tong, W., Gui, Y., et al. (2006). Shanghaigrid: an information service grid. *Concurrency and Computation: Practice and Experience*, 18(1):111–135.

[Li et al. 2014] Li, S., Zhang, Y., Raychaudhuri, D., and Ravindran, R. (2014). A comparative study of mobilityfirst and ndn based icn-iot architectures. In *10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 158–163. IEEE.

[Lindgren et al. 2003] Lindgren, A., Doria, A., and Schelén, O. (2003). Probabilistic routing in intermittently connected networks. In *ACM International Symposium on Mobilde Ad Hoc Networking and Computing, MobiHoc 2003: 01/06/2003-03/06/2003*.

[Loo et al. 2016] Loo, J., Mauri, J. L., and Ortiz, J. H. (2016). *Mobile ad hoc networks: current status and future trends*. CRC Press.

[Mandula et al. 2015] Mandula, K., Parupalli, R., Murty, C. A., Magesh, E., and Lunagariya, R. (2015). Mobile based home automation using internet of things (iot). In *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pages 340–343. IEEE.

[Maróti et al. 2004] Maróti, M., Kusy, B., Simon, G., and Lédeczi, Á. (2004). The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49. ACM.

[Mohapatra and Rekha 2012] Mohapatra, S. and Rekha, K. S. (2012). Sensor-cloud: a hybrid framework for remote patient monitoring. *International Journal of Computer Applications*, 55(2).

[Montanini et al. 2016] Montanini, L., Raffaeli, L., De Santis, A., Del Campo, A., Chiatti, C., Rascioni, G., Gambi, E., and Spinsante, S. (2016). Overnight supervision of alzheimer's disease patients in nursing homes: System development and field trial. In *2nd International Conference on Information and Communication Technologies for Ageing Well and e-Health, ICT4AWE 2016*, pages 15–25. SciTePress.

[Oh et al. 2010] Oh, S.-Y., Lau, D., and Gerla, M. (2010). Content centric networking in tactical and emergency manets. *Wireless days*, 10.

[Osman et al. 2009] Osman, E., El-Gazar, M., Shaat, M., El-Kafas, A., Zidan, W., and Wadoud, A. (2009). An estimation of a passive infra-red sensor probability of detection.

[Othman and Shazali 2012] Othman, M. F. and Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. *Procedia Engineering*, 41:1204–1210.

[Pelusi et al. 2006] Pelusi, L., Passarella, A., and Conti, M. (2006). Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE communications Magazine*, 44(11):134–141.

[Perera et al. 2014] Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414–454.

[Piyare 2013] Piyare, R. (2013). Internet of things: ubiquitous home control and monitoring system using android based smart phone. *International journal of Internet of Things*, 2(1):5–11.

[Plagemann et al. 2006] Plagemann, T., Goebel, V., Mauthe, A., Mathy, L., Turletti, T., and Urvoy-Keller, G. (2006). From content distribution networks to content networks—issues and challenges. *Computer Communications*, 29(5):551–562.

[Postel et al. 1980] Postel, J. et al. (1980). User datagram protocol.

[Ray 2018] Ray, P. P. (2018). A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3):291–319.

[Rodriguez et al. 2017] Rodriguez, L. G. A., de Jeus, J. A., do Rosário, V. M., da Silva, A. F., Peres, L. P., de Moraes, H. F., and de Amorim, C. L. (2017). mybee: An information system for precision beekeeping. In *ICEIS (2)*, pages 577–587.

[Royer and Toh 1999] Royer, E. M. and Toh, C.-K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE personal communications*, 6(2):46–55.

[Saadallah et al. 2012] Saadallah, B., Lahmadi, A., and Festor, O. (2012). Ccnx for contiki: implementation details.

[Salles 2014] Salles, R. C. (2014). Avaliação de capacidade e consumo de energia de rede móvel ad hoc centrada em interesse. Master's thesis, Universidade Federal do Rio de Janeiro. https://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/2497.

[Santos et al. 2016] Santos, B. P., Silva, L., Celes, C., Borges, J. B., Neto, B. S. P., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. (2016). Internet das coisas: da teoria à prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos*.

[Shkurti et al. 2017] Shkurti, L., Bajrami, X., Canhasi, E., Limani, B., Krrabaj, S., and Hulaj, A. (2017). Development of ambient environmental monitoring system through wireless sensor network (wsn) using nodemcu and "wsn monitoring". In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–5. IEEE.

[Silva et al. 2013] Silva, M. D., Moraes, H. F., and Amorim, C. L. (2013). Persistência de dados em redes móveis ad-hoc centradas em interesses. In *Anais Estendidos do XIX Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 77–80. SBC.

[Sobin et al. 2016] Sobin, C., Raychoudhury, V., Marfia, G., and Singla, A. (2016). A survey of routing and data dissemination in delay tolerant networks. *Journal of Network and Computer Applications*, 67:128–146.

[Statista 2019] Statista (2019). Iot: number of connected devices worldwide 2012-2025.

[Swan 2012] Swan, M. (2012). Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. *Journal of Sensor and Actuator networks*, 1(3):217–253.

[Trifunovic et al. 2017] Trifunovic, S., Kouyoumdjieva, S. T., Distl, B., Pajevic, L., Karlsson, G., and Plattner, B. (2017). A decade of research in opportunistic networks: challenges, relevance, and future directions. *IEEE Communications Magazine*, 55(1):168–173.

[Upton and Halfacree 2014] Upton, E. and Halfacree, G. (2014). *Raspberry Pi user guide*. John Wiley & Sons.

[Vahdat et al. 2000] Vahdat, A., Becker, D., et al. (2000). Epidemic routing for partially connected ad hoc networks.

[Vermesan et al. 2011] Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I. S., Mazura, M., Harrison, M., Eisenhauer, M., et al. (2011). Internet of things strategic research roadmap. *Internet of things-global technological and societal trends*, 1(2011):9–52.

[Xylomenos et al. 2013] Xylomenos, G., Ververidis, C. N., Siris, V. A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K. V., and Polyzos, G. C. (2013). A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, 16(2):1024–1049.

[Zanella et al. 2014] Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32.

[Zhou et al. 2012] Zhou, H., Liu, B., and Wang, D. (2012). Design and research of urban intelligent transportation system based on the internet of things. In *Internet of Things*, pages 572–580. Springer.

Apêndice A

Preparação do Ambiente de Testes

O objetivo deste apêndice é abordar o procedimento de instalação e configuração do ambiente de testes direcionados a este trabalho.

A.1 Procedimento de instalação do sistema operacional no cartão SD

Para instalação do sistema operacional na placa Raspberry Pi 3 é necessário a utilização dos seguintes equipamentos:

- Teclado e Mouse (USB)
- Fonte de alimentação de 5v / 3A com conexão micro USB;
- Um monitor de vídeo com entrada HDMI;
- Um cartão microSD (8GB recomendado); e
- Um computador com entrada para cartão SD;

A partir deste ponto, é necessário instalar o Sistema Operacional "Raspbian"no cartão microSD. A seguir são descritos os passos para instalação do Sistema Operacional:

1. Baixar a imagem do Sistema Operacional https://downloads.raspberrypi.org/raspbian_latest.
2. Descompactar o arquivo baixado "2019-09-26-raspbian-buster.zip"e extrair o arquivo imagem que contém no formato ".img".

3. Para instalar o arquivo ".img"no cartão microSD é necessário a utilização de um software específico para esta finalidade. Um software recomendado é o *Balena Etcher* que pode ser baixado através do endereço https://www.balena.io/etcher/;

4. Por fim, com o cartão microSD inserido no computador e o *software Balena Etcher* instalado no sistema operacional, selecione o arquivo imagem (2019-09-06-raspbian-buster.zip), escolha o cartão microSD através da opção *drive* e clique em *flash*. A Figura A.1 exibe a interface gráfica do *Balena Etcher*.

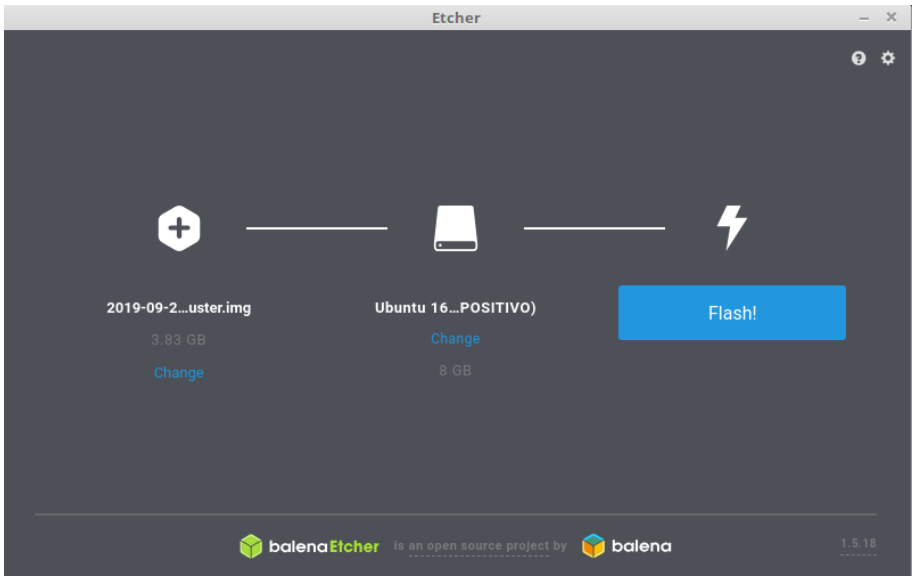

Figura A.1: Interface do *software Balena Etcher*

Após a instalação do sistema operacional, no cartão microSD possuirá duas partições: na primeira, formatada em FAT32, estão os arquivos bootcode.bin, start.elf, fixup.dat, config.txt, kernel.img e cmdline.txt e na segunda está o sistema de arquivos do root.

A.2 Configuração do computador Raspberry para o modo *ad hoc*

1. Fazer uma cópia do arquivo "interfaces"localizado em /etc/network/.

```
sudo cp /etc/network/interfaces /etc/network/interfaces.copy
```

2. Abrir o arquivo "interfaces"

```
sudo nano /etc/network/interfaces
```

3. Inserir o seguinte conteúdo no arquivo:

```
auto wlan0
iface wlan0 inet static
address 192.168.1.1
netmask 255.255.255.0
wireless-channel 1
wireless-essid MyNetwork
wireless-mode ad-hoc
```

4. Reiniciar o Raspberry

Vale ressaltar que o endereço IP será diferente em cada dispositivo configurado. Além disso, a interface de rede, neste caso identificado como wlan0, poderá ser diferente em outros dispositivos. Portanto, é necessário veriricar qual é o seu identificador. A interface de rede pode ser verificada através do comando "ifconfig".

Apêndice B

Instalação do Sistema Hermes na plataforma Raspberry

O objetivo deste apêndice é informar o procedimento de instalação do Sistema Hermes juntamente com o protocolo RadNet. Este procedimento foi inteiramente executado sobre a plataforma Raspberry Pi 3 utilizando o sistema operacional "Raspbian".

O Sistema Hermes e o protocolo de comunicação RadNet podem ser obtidos entrando em contato com o autor deste projeto através do e-mail: lseverino[at]cos.ufrj.br.

B.1 Dependências do projeto

Para implementação funcional do Sistema Hermes é necessário a instalação das seguintes dependências:

- **Python versão 2.7**: Para execução do projeto. É sugerido a utilização do Python 2.7 uma vez que toda a implementação e testes do Sistema Hermes foram executados com base nesta versão;
- **Adafruit's DHT library**: Bilioteca para Arduino e Raspberry com a finalidade de efetuar as leituras dos sensores de temperatura e umidade (DHT22);
- **Protocolo RadNet**: Protocolo de comunicação que implementa o conceito de redes oportunísticas centradas em interesse;

B.2 Instalação das dependências do projeto

Python Versão 2.7

```
sudo apt-get install python2.7
```

Adafruit's DHT library

```
sudo pip2 install Adafruit_DHT
```

B.2.1 Protocolo RadNet

O procedimento de instalação do protocolo RadNet foi testado nas distribuições Ubuntu, Mint e Raspbian. Todos os comando apresentados a seguir devem ser executados dentro do diretório do projeto da RadNet. Para tal finalidade, abra o terminal de comando dentro do diretório do projeto. Execute o comando abaixo para instalação dos recursos de compilação do projeto:

```
sudo apt-get install make gcc autoconf
```

É necessário compilar e instalar a biblioteca estática de utilidades antes de realizar a compilação do *daemon* da RadNet. Portanto, ainda no diretório do projeto da RadNet, vá ao diretório **repa-c-utils** e execute os comandos informados a seguir:

```
sudo chmod +x autogen.sh
./autogen.sh
.configure
make
sudo make install
```

Agora é preciso compilar o *daemon*. Entre no diretório **repa-daemon** e:

```
make clean
make
sudo make install
```

Neste ponto, o protocolo de comunicação da RadNet já está instalado e pronto para ser utilizado no dispositivo em questão. Assim sendo, o *daemon* da radNet pode ser executado através do "repad -t"no terminal, a *flag* "-t"exibe um *log* no terminal.

Após a instalação do *daemon*, é necessário gerar o módulo Python. Para isso é necessário executar, ainda dentro do diretório **repa-daemon**, os seguinte comandos:

```
python setup.py build
sudo python setup.py install
sudo chmod 755 /usr/local/lib/python2.7/dist-packages/*.so
```

Obs. Para que essa compilação funcione, o pacote python-dev deve estar instalado no sistema.