

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Tese

**Balanceamento de Carga Baseado em Divisão de Tarefas Multimídia para RSSF
Reativas**

Weslen Schiavon de Souza

Pelotas, 2024

Weslen Schiavon de Souza

**Balanceamento de Carga Baseado em Divisão de Tarefas Multimídia para RSSF
Reativas**

Tese apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Lisane Brisolara de Brisolara
Coorientador: Prof. Dr. Paulo Roberto Ferreira Jr.

Pelotas, 2024

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação da Publicação

S719b Souza, Weslen Schiavon de

Balanceamento de carga baseado em divisão de tarefas multimídia para RSSF reativas [recurso eletrônico] / Weslen Schiavon de Souza ; Lisane Brisolara de Brisolara, orientadora ; Paulo Roberto Ferreira Junior, coorientador. — Pelotas, 2024.

91 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2024.

1. Redes de sensores multimídia sem fio. 2. Processamento de eventos. 3. Balanceamento de carga. 4. RSMSF. 5. RSSF. I. Brisolara, Lisane Brisolara de, orient. II. Ferreira Junior, Paulo Roberto, coorient. III. Título.

CDD 005

Elaborada por Maria Inez Figueiredo Figas Machado CRB: 10/1612

Weslen Schiavon de Souza

**Balanceamento de Carga Baseado em Divisão de Tarefas Multimídia para RSSF
Reativas**

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 11 de outubro de 2024

Banca Examinadora:

Prof. Dr. Lisane Brisolara de Brisolara (orientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Leonardo Bidese de Pinho

Doutor em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro.

Prof. Dr. Felipe de Souza Marques

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Leandro Buss Becker

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Dedico aos meus pais, os quais proporcionaram que tudo fosse possível...

AGRADECIMENTOS

Agradeço primeiramente aqueles sem os quais eu jamais estaria presente, meus pais, Rosângela e Valdirnei, eles que sempre se esforçaram para me proporcionar as melhores oportunidades de ensino e sempre me incentivaram a seguir em frente.

Aos meus amigos e professores que de alguma forma contribuíram para que este trabalho fosse possível, agradeço a todos...

O homem erudito é um descobridor de fatos que já existem, mas o homem sábio é um criador de valores que não existem e que ele faz existir.

— ALBERT EINSTEIN

RESUMO

SOUZA, Weslen Schiavon de. **Balanceamento de Carga Baseado em Divisão de Tarefas Multimídia para RSSF Reativas.** Orientador: Lisane Brisolara de Brisolara. 2024. 91 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2024.

Este trabalho apresenta o TDMEN, uma abordagem inovadora para Redes de Sensores Sem Fio Multimídia (RSSFM) que visa otimizar o consumo energético e prolongar a vida útil da rede. O TDMEN aplica uma divisão estratégica de tarefas, permitindo o processamento local dos dados e o balanceamento eficiente da energia entre os dispositivos. Duas variações do TDMEN foram desenvolvidas: uma para redes com redundância temporal e outra para redes com redundância espacial. Foram realizados testes comparativos com abordagens bioinspiradas e estratégias de processamento local, em cenários com e sem perdas de pacotes. Os resultados indicaram que o TDMEN pode aumentar a vida útil da rede em até 75% em comparação com abordagens sem balanceamento e até 44% em relação a abordagens bioinspiradas. A solução mostrou-se eficaz em diferentes cenários, tanto em redes com sobreposição de sensoreamento quanto em redes sem essa característica, proporcionando um balanceamento energético superior e aumentando a eficiência geral da rede.

Palavras-chave: Redes de Sensores Multimídia Sem Fio Processamento de Eventos Balanceamento de Carga RSMSF RSSF

ABSTRACT

SOUZA, Weslen Schiavon de. **Task Division-Based Load Balancing for Reactive WMSNs**. Advisor: Lisane Brisolara de Brisolara. 2024. 91 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2024.

This paper presents TDMEN, an innovative approach for Wireless Multimedia Sensor Networks (WMSNs) aimed at optimizing energy consumption and extending network lifetime. TDMEN applies a strategic task division, enabling local data processing and efficient energy balancing among devices. Two variations of TDMEN were developed: one for networks with temporal redundancy and another for networks with spatial redundancy. Comparative tests were conducted with bio-inspired approaches and local processing strategies in scenarios with and without packet loss. The results indicated that TDMEN can increase network lifetime by up to 75% compared to non-balanced approaches and by up to 44% compared to bio-inspired approaches. The solution proved effective in different scenarios, both in networks with sensing overlap and in networks without this characteristic, providing superior energy balancing and enhancing overall network efficiency.

Keywords: Wireless Multimedia Sensor Network Event Processing Load Balancing WMSN WSN

LISTA DE FIGURAS

Figura 1	Rede de sensores sem fio.	21
Figura 2	Rede de sensores sem fio organizada em cluster.	24
Figura 3	Rede de sensores sem fio organizada em MultiHop.	25
Figura 4	Simulação com interface gráfica no Eboracum.	27
Figura 5	Exemplo de RSSF Sem sobreposição de sensoriamento: Interação entre nodos logo após sensoriamento de um evento.	37
Figura 6	Exemplo de RSSF Sem sobreposição de sensoriamento: Fluxo de comunicação entre nó mestre e vizinho.	38
Figura 7	Processo de Segmentação de imagem.	39
Figura 8	Diagrama de classes da implementação da abordagem TDMEN.	40
Figura 9	Segmentação e variação entre eventos com redundância temporal.	43
Figura 10	Cache de imagem.	44
Figura 11	Diagrama de classes da implementação da TDMEN-CACHE.	45
Figura 12	Exemplo de RSSFM com redundância espacial e as comunicações empregadas pela TDMEN-INTERSEC.	48
Figura 13	Fluxo de comunicação TDMEN-INTERSEC.	53
Figura 14	Diagrama de classes TDMEN-INTERSEC.	54
Figura 15	Armadilhas Eletrônicas compondo uma RSSF para manejo de pragas.	57
Figura 16	Diagrama de blocos da Armadilha inteligente	57
Figura 17	Cenário de simulação para manejo de pragas, pontos vermelho representam eventos e círculos azul, nodos sensores.	59
Figura 18	Distribuição de probabilidade espacial Normal Invertida.	59
Figura 19	Estudo de Caso 1: Média de Eventos por dia - rede com TDMEN versus rede Sem balanceamento	63
Figura 20	Estudo de Caso 1: Média de eventos por dia obtidos com as três abordagens	65
Figura 21	Estudo de Caso 1: Resultados de eventos detectados por dia pela rede com TDMEN-CACHE com e sem perdas de pacotes e pela rede sem balanceamento.	67
Figura 22	Estudo de Caso 1: Número de eventos detectado na rede com TDMEN-CACHE (com sem perdas) em comparativo com a rede sem balanceamento (sem perdas de pacote).	68
Figura 23	Estudo de Caso 1: Eventos sensoreados por dia pela TDMEN-CACHE em comparativo com a RSSF sem balanceamento, ambas considerando perdas de 1%.	70

Figura 24	Estudo de caso 1: Número de eventos detectado na rede com TDMEN-CACHE em comparativo com a RSSF sem balanceamento, ambas considerando perdas de 1%.	71
Figura 25	Estudo de Caso 1: Impacto da perda de pacotes na detecção de eventos na rede usando TDMEN-CACHE.	72
Figura 26	Diagrama de blocos da Armadilha inteligente para passaros.	74
Figura 27	Cenário de simulação para monitoramento do canto de pássaros.	77
Figura 28	Estudo de Caso 2: Comparativo de detecção de eventos por dia das estratégias TDMEN-INTERSEC, TDMEN Original, PS, Ant e Sem Balanceamento.	78
Figura 29	Estudo de Caso 2: Média de Eventos detectados pelas estratégias TDMEN-INTERSEC, TDMEN Original, PS, Ant e Sem Balanceamento.	79
Figura 30	Eventos detectados em um meio com perdas pelas estratégias TDMEN-INTERSEC, PS, Ant e Sem Balanceamento.	81
Figura 31	Detecção de eventos por dia em um meio com perdas pelas estratégias TDMEN, PS, Ant e Sem Balanceamento.	83

LISTA DE TABELAS

Tabela 1	Trabalhos relacionados	34
Tabela 2	Corrente gasta pelos módulos da Armadilhas	58
Tabela 3	Custo de energia da tarefa sem balanceamento de carga.	61
Tabela 4	Custo de energia de cada tarefa com TDMEN.	62
Tabela 5	Custo de energia de cada tarefa com TDMEN-CACHE.	62
Tabela 6	Estudo de Caso 1: Resultados de Tempo de duração em dias da rede com TDMEN e Sem Balanceamento	64
Tabela 7	Estudo de Caso 1: Resultados do tempo de vida da rede em dias para as três abordagens	66
Tabela 8	Estudo de Caso 1: Resultados do tempo de vida da rede em dias para TDMEN-CACHE com e sem perdas e para a rede sem balanceamento de carga	67
Tabela 9	Estudo de Caso 1: Resultados de eventos detectados pelo TDMEN-CACHE (com perdas e sem perdas de pacote) e pela rede sem balanceamento (sem perdas de pacote).	68
Tabela 10	Estudo de Caso 1: Porcentagem de eventos perdidos pela rede com TDMEN-CACHE, variando a porcentagem de perdas.	69
Tabela 11	Estudo de Caso 1: porcentagem de eventos detectados a mais pela rede com TDMEN-CACHE, com perdas de pacote variadas, em comparativo com a rede sem balanceamento (e sem perdas de pacotes).	69
Tabela 12	Estudo de caso 1: Número de eventos detectados pela rede usando TDMEN-CACHE e pela rede sem balanceamento, ambas considerado perdas de 1%.	70
Tabela 13	Estudo de Caso 2: Custo de energia de tarefas para nova abordagem.	75
Tabela 14	Estudo de Caso 2: Custo de energia de tarefas para PS, ANT e Sem Balanceamento.	75
Tabela 15	Total de eventos detectados TDMEN-INTERSEC, PS, Ant e sem balanceamento.	79
Tabela 16	Tempo de vida da rede TDMEN-INTERSEC, PS, Ant e Sem Balanceamento.	80
Tabela 17	Comparativo total de eventos detectados com perdas pelas estratégias: PS, Ant e sem Balanceamento em comparativo a abordagem proposta TDMEN-INTERSEC.	82

LISTA DE ABREVIATURAS E SIGLAS

CPU	Central Processing Unit
IA	Inteligência Artificial
NROI	número de regiões de interesse
PS	Pheromone-Signaling
RF	Radio-frequênciia
ROI	Regiões de interesse
RSSF	redes de sensores sem fio
WMSN	Wireless multimedia sensor network
WSN	Wireless Sensor Networks

SUMÁRIO

1	INTRODUÇÃO	15
2	EMBASAMENTO TEÓRICO	20
2.1	Redes de Sensores Sem Fio	20
2.1.1	Tipo de Evento sensoreado	22
2.1.2	Tipo de sensoriamento	23
2.1.3	Arquitetura de rede	24
2.2	Infraestrutura de Simulação	26
3	TRABALHOS RELACIONADOS	29
4	BALANCEAMENTO DE CARGA BASEADO EM DIVISÃO DE TAREFAS	35
4.1	TDMEN: Divisão de tarefas para eventos Multimídia em Redes de Sensores Sem Fio	36
4.2	TDMEN-CACHE: TDMEN integrado com esquema de cache	41
4.3	TDMEN-INTERSEC: Balanceamento de Carga em Redes com Sobreposição de Sensoreamento	47
4.4	Modelagem da Perda de pacotes	51
5	ESTUDO DE CASO 1: DIVISÃO DE TAREFAS EM RSSFM SEM REDUNDÂNCIA DE SENSOREAMENTO	55
5.1	RSSF para o manejo de pragas em pomares	55
5.2	Modelagem do Estudo de Caso	58
5.3	Resultados Comparativos TDMEN contra Rede sem Balanceamento de Carga	63
5.4	Resultados Comparativos TDMEN-CACHE x TDMEN x Sem Balanceamento de Carga	64
5.5	Resultados Comparativos TDMEN-CACHE com simulação de perda de pacotes	66
6	ESTUDO DE CASO 2: DIVISÃO DE TAREFAS EM RSSFM COM REDUNDÂNCIA DE SENSOREAMENTO	73
6.1	Modelagem do Estudo de Caso TDMEN-INTERSEC	73
6.2	Comparativo TDMEN-INTERSEC com demais estratégias	78
7	CONCLUSÃO	84
	REFERÊNCIAS	87

1 INTRODUÇÃO

Com os consideráveis avanços recentes na tecnologia de microprocessadores e comunicação sem fio, tornou-se possível a integração de dispositivos tecnológicos para o monitoramento e gerenciamento de uma variedade de tarefas. Entre essas tarefas, podemos citar o monitoramento da fauna e flora em áreas de preservação ambiental, o gerenciamento da aplicação de defensivos contra pragas em lavouras, na agricultura de precisão e, o monitoramento dos sinais vitais de pacientes em ambientes hospitalares.

Nesse contexto de gerenciamento e monitoramento de atividades por meio de dispositivos tecnológicos, surgem as redes de sensores sem fio (RSSF) ou (WSN, do inglês, *Wireless Sensor Networks*), definidas como redes que conectam dispositivos autônomos e distribuídos espacialmente utilizando tecnologia de conexão sem fio. Esses dispositivos possuem capacidade de sensoriamento e são normalmente empregados para o monitoramento de fenômenos físicos ou ambientais (Khalaf; Sabbar, 2019). Esses sensores são comumente chamados de nós ou nodos da RSSF, e através de comunicação eles cooperam para cumprir os objetivos da aplicação alvo. A arquitetura típica de cada nó de uma RSSF consiste em memória, sensor, bateria, transceptor sem fio e um microcontrolador ou microprocessador (Alkhatib; Baicher, 2012).

As redes de sensores sem fio são aplicadas em uma ampla gama de setores, incluindo agricultura, saúde, segurança, meio ambiente, entre outros. Na agricultura, por exemplo, essas redes são usadas para o monitoramento e controle das condições climáticas, do solo e das plantas, possibilitando uma tomada de decisão mais eficiente em relação às práticas de cultivo (Thakur; Kumar; Kumar; Singh, 2019). Na área da saúde, as redes de sensores sem fio são aplicadas no monitoramento de sinais vitais e marcadores clínicos de pacientes, notificando profissionais médicos quando necessário (Sisodia; Swati, 2020).

Um aspecto importante das redes de sensores sem fio é sua capacidade de autoconfiguração e reorganização, uma vez que esses dispositivos frequentemente são implantados em ambientes de difícil acesso. Muitas vezes, eles são simplesmente

colocados no local sem qualquer organização ou cuidado, o que pode resultar em dispositivos sobrepostos ou mesmo incomunicáveis. Além disso, em locais remotos, é comum a ausência de uma fonte de energia permanente, o que requer que cada dispositivo seja alimentado por uma bateria individual. Esse tipo de alimentação demanda um gerenciamento eficiente de energia para maximizar a vida útil de cada nó e manter a rede operacional pelo maior tempo possível (Amutha; Sharma; Nagar, 2020).

Dentre as arquiteturas de rede adotadas em redes de sensores sem fio, destacam-se as arquiteturas MultiHop e Cluster. Em redes MultiHop, os dados sensoreados são transmitidos de um nó para outro, saltando de nó em nó pela rede até atingirem seu destino final. Geralmente, o destino final é o Sink (também chamado de Estação-base), um nó que normalmente não possui limitações energéticas e, portanto, pode processar os dados sensoreados pela rede sem dificuldades. Por outro lado, em redes baseadas em uma arquitetura de Cluster, a rede é subdividida em grupos de nós. Cada grupo tem um nó designado como Gateway, que recebe os dados dos demais nós do grupo e os retransmite. Essas retransmissões dos nós Gateways, também conhecidos como *Cluster Heads*, normalmente têm como destino outro Cluster Head ou a estação base, que pode executar o processamento dos dados enviados (Raj; Bhattacharyya; Midhunchakkaravarthy; Kim, 2021).

Os dados monitorados e/ou transmitidos em Redes de Sensores Sem Fio podem adotar diversos formatos, os quais estão diretamente relacionados ao ambiente em que essas redes estão implantadas e aos eventos que estas monitoram. No passado, devido às limitações de protocolos e à capacidade de memória limitada dos dispositivos, os dados em formato de texto ou binário eram os mais comuns. Por exemplo, uma RSSF utilizada para monitorar o solo capturaria e transmitiria dados sobre umidade, temperatura e outras informações relevantes (Thakur; Kumar; Kumar; Singh, 2019).

No entanto, atualmente, existem redes que são capazes de capturar dados multimídia do ambiente, como imagens, áudio ou até mesmo registros de vídeo (Usman; Jan; He; Chen, 2019). Essas redes são conhecidas como WMSN (do inglês, *Wireless multimedia sensor network*.) Um exemplo prático seria uma rede composta por armadilhas eletrônicas usadas para o manejo de pragas em lavouras. Essas armadilhas eletrônicas capturam imagens dos insetos que ficam presos nelas. Com as imagens obtidas, é possível realizar a identificação dos insetos e determinar se representam algum tipo de risco para a cultura em questão, onde as armadilhas estão instaladas (Thakur; Kumar; Kumar; Singh, 2019).

Nesse contexto de redes de sensores envolvendo a captura de dados multimídia, surgem novos desafios significativos. A transmissão de grandes volumes de dados consome uma quantidade considerável de energia da bateria dos nodos. Enviar arquivos multimídia, como imagens, por toda a rede até o Sink representa um alto consumo energético. O custo associado ao envio desse tipo de informação é especialmente

elevado em redes MultiHop, devido ao número de saltos que os dados realizam, requerendo a retransmissão dos dados por vários nós até de atingir seu destino. Os protocolos também apresentam limitações quanto ao máximo de bytes que um pacote pode transmitir individualmente, o que contribui para o aumento do número de transmissões requeridas, pois ao dividir a informação em mais pacotes, devem ser adicionados bytes de cabeçalho. Por fim, todos os nós possuem uma quantidade limitada de bateria, o que torna inviável o envio contínuo de eventos multimídia pela rede. Esses desafios destacam a importância do desenvolvimento de estratégias eficientes de gerenciamento de energia e de otimização de protocolos de transmissão para redes de sensores multimídia.

Diferentes técnicas de balanceamento de carga em Redes de Sensores Sem Fio estão sendo estudadas e propostas com o objetivo de reduzir o consumo de energia dos nós e prolongar o tempo de monitoramento. No entanto, a carga de processamento muitas vezes é negligenciada, onde muitos trabalhos focam em otimizações para reduzir o consumo associado à comunicação exclusivamente, como discutido em (Saleem; Alabady, 2023), (Chiwariro; Thangadurai, 2023), (Pereira; Brisolara; Ferreira, 2020) e (Genta; Lobiyal; Abawajy, 2019). A maioria dos trabalhos que buscam reduzir o consumo relacionado ao processamento exploram a redundância no sensoriamento, de modo a evitar que o processamento do mesmo evento resulte na descarga de energia em mais de um nó, como abordado em (Pereira; Brisolara; Ferreira, 2020), (Ferreira; Brisolara; Indrusiak, 2015) e (Caliskanelli; Harbin; Indrusiak; Mitchell; Polack; Chesmore, 2013). Ainda, alguns trabalhos encontrados na literatura que executavam o processamento local, dependiam de hardwares especializados (Kethireddy; Rallapalli; Chilakala; Devulapalli, 2023), ou ainda que os nodos dispusessem de conexão ativa com a internet (Varshovi; Kavian; Ansari-asl, 2019).

Poucos estudos identificados abordam o processamento local em redes de sensores sem fio multimídia, sendo que, frequentemente, esses estudos não empregam estratégias de balanceamento de carga ou apresentam requisitos muito específicos, como hardware especializado e a necessidade de conexão ativa com a internet, para operarem efetivamente. Ainda, algumas dessas abordagens demonstram desempenho satisfatório apenas em situações onde há uma sobreposição significativa entre os raios de cobertura dos nodos, o que pode não corresponder a alguns cenários de aplicação. Além disso, estas abordagens limitam-se a decidir qual nodo processará a tarefa por completo. No entanto, aplicações multimídia costumam repetir processamento em grupos de dados, o que não é explorado por estas abordagens (Matheen; Sundar, 2023).

Este é um campo de pesquisa importante e em evolução, uma vez que a captura e transmissão de dados multimídia em RSSFM apresenta desafios adicionais em termos de consumo de energia e processamento. Futuras pesquisas podem ser direcionadas

para desenvolver estratégias de balanceamento de carga específicas para redes de sensores multimídia, levando em consideração as demandas únicas que surgem com a detecção e transmissão de eventos multimídia.

Este trabalho propõe uma abordagem inovadora, fundamentada nos estudos identificados na literatura, para enfrentar os desafios associados ao processamento de dados multimídia em redes de sensores sem fio, também conhecidas como RSSFM. A principal proposta dessa abordagem é a divisão das tarefas de processamento de dados multimídia, visando alcançar um balanceamento mais eficiente da carga entre os nós da rede. Nesse contexto, nenhum nó será encarregado do processamento completo de um evento multimídia; em vez disso, a carga de processamento de cada evento será dividida e suas partes serão distribuídas entre os nós vizinhos. Essa estratégia visa evitar sobrecarga de nós e atingir uma descarga mais uniforme das baterias dos nós da rede, prolongando, assim, a vida útil da rede como um todo.

Assim, esta tese propõe a abordagem TDMEN (*Task Division for Multimedia Events in Wireless Sensor Networks*), que trata da divisão de tarefas para eventos multimídia em redes de sensores sem fio, como uma estratégia para permitir o processamento local, evitando transmissão de grandes volumes. A abordagem foi implementada e avaliada utilizando o simulador Eboracum em um estudo de caso com nodos sem sobreposição nas áreas de sensoriamento. Uma otimização dessa abordagem foi denominada TDMEN-CACHE, a qual explora a possível redundância de informações entre eventos multimídia consecutivos, evitando a execução de processamentos de dados redundantes.

Além disso, visando aproximar as simulações empregadas de um ambiente real, foi conduzido um estudo sobre a influência das interferências externas nas transmissões de dados. Por meio deste estudo, foi possível introduzir, nos nossos experimentos, a simulação de perda de pacotes durante as transferências de dados.

Complementando a contribuição principal desta tese, foi estudada e proposta uma adaptação da abordagem TDMEN para permitir seu emprego em redes com nodos com sobreposição nas áreas de sensoreamento. Essa estratégia foi denominada TDMEN-INTERSEC, a qual aproveita a detecção redundante dos nodos para平衡ar a carga entre eles, sem a necessidade de gastar energia enviando os dados para processamento. Através de um estudo de caso, experimentos compararam a abordagem TDMEN-INTERSEC com outras abordagens de balanceamento de carga.

O restante deste documento está organizado da seguinte forma: ****Capítulo 2 - Embasamento Teórico****: este capítulo apresentará o embasamento teórico fundamental para compreender o contexto deste estudo. Ele incluirá conceitos relacionados a redes de sensores sem fio e detalhará a infraestrutura de modelagem e simulação utilizada neste trabalho; ****Capítulo 3 - Trabalhos Relacionados****: neste capítulo, serão discutidos os trabalhos relacionados relevantes para este estudo. Será feita uma

análise da literatura existente para contextualizar a pesquisa; ****Capítulo 4 - Metodologia e Estratégia Proposta****: este capítulo apresentará a metodologia e a estratégia proposta para resolver o problema abordado neste documento. Serão detalhadas as abordagens e os métodos utilizados; ****Capítulo 5 e 6 - Resultados dos Estudos de Caso Abordados****: aqui, serão discutidos os resultados obtidos a partir da execução das estratégias propostas e desenvolvidas nesta tese. Os resultados serão analisados e discutidos em detalhes; ****Capítulo 7 - Conclusão****: O último capítulo realizará uma discussão abrangente sobre os resultados já obtidos, além de abordar possíveis direções para trabalhos futuros.

2 EMBASAMENTO TEÓRICO

Neste capítulo, é apresentada uma revisão sobre redes de sensores sem fio na Seção 2.1, e a revisão da infraestrutura empregada para a concepção e avaliação das estratégias desenvolvidas por este presente estudo é detalhada na Seção 2.2.

2.1 Redes de Sensores Sem Fio

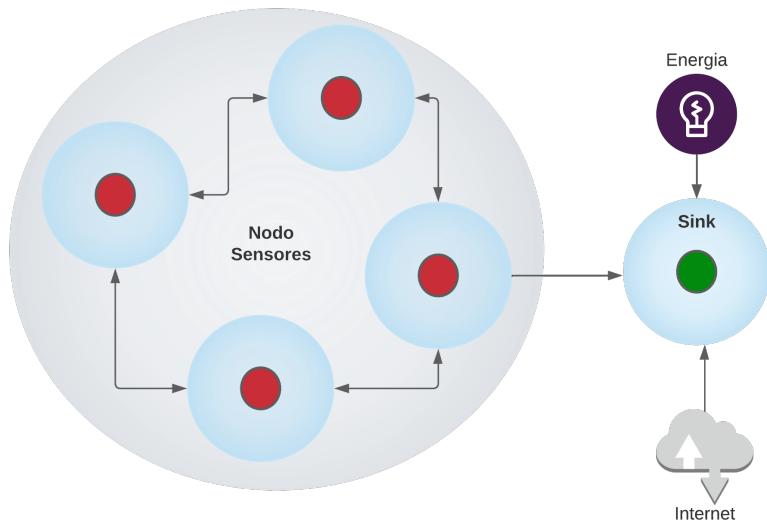
No contexto do gerenciamento e monitoramento de atividades por meio de tecnologia, surgem as redes de sensores sem fio (RSSF). Essas redes são definidas como conjuntos de dispositivos autônomos e distribuídos espacialmente, com capacidades de sensoriamento, empregados para monitorar fenômenos físicos ou ambientais (Kandris; Nakas; Vomvas; Koulouras, 2020). Cada dispositivo é conhecido como um nó ou nodo, e normalmente se comunica com outros nós por meio de uma rede sem fio, permitindo a cooperação mútua entre os elementos da rede (Khalaf; Sabbar, 2019). A arquitetura típica de cada nó inclui memória, sensor, bateria, transceptor sem fio e um microcontrolador (Alkhatib; Baicher, 2012).

A Figura 1 ilustra uma rede de sensores sem fio com quatro nós sensores e um nó Sink. Neste exemplo, os nós sensores se comunicam entre si para enviar as informações sensoriadas do ambiente até o nó Sink. O nó Sink é geralmente posicionado a uma distância maior dos nós sensores, facilitando o acesso e permitindo que ele possua conectividade com a Internet e fonte de energia elétrica disponível, conforme explicado por (Demetillo; Japitana; Taboada, 2019).

As principais características de uma WSN incluem a capacidade de autoconfiguração e reorganização em caso de falha de algum nó, a presença de uma considerável quantidade de sensores distribuídos, a necessidade de gerenciamento contínuo de energia devido à limitação das baterias disponíveis nos nós e a distribuição dos nós em um meio físico muitas vezes desafiador e de difícil acesso, onde os sensores são implantados sem seguir um padrão ou uniformidade, como mencionado por (Kandris; Nakas; Vomvas; Koulouras, 2020).

Dadas as características mencionadas das redes de sensores sem fio e sua ampla

Figura 1 – Rede de sensores sem fio.



Fonte: Elaborada pelo autor.

aplicabilidade para auxiliar e solucionar diversos problemas em diferentes contextos, esse paradigma computacional tem um considerável potencial para facilitar e melhorar várias tarefas do dia a dia das pessoas. No entanto, para que isso seja possível, ainda existem alguns desafios que precisam ser superados para que as redes de sensores sem fio se tornem cada vez mais presentes e integradas ao nosso mundo, como apontado em (Kandris; Nakas; Vomvas; Koulouras, 2020).

Entre os principais desafios comuns em redes de sensores sem fio, podemos citar:

Limitação de energia — Como os nós sensores são geralmente posicionados em áreas remotas de difícil acesso, eles são alimentados por baterias com vida útil limitada. Isso resulta na necessidade de um gerenciamento eficiente de energia pelos nós, a fim de prolongar ao máximo a vida útil das baterias; **Adaptabilidade a falhas** — Uma vez que uma rede de sensores é composta por diversos nós que trabalham em conjunto, quando um desses nós falha devido à falta de bateria ou a uma falha crítica de comunicação, a rede não pode parar. Os outros nós que anteriormente se comunicavam com o nó falho devem ser capazes de se conectar a outros nós para manter a comunicação e o fluxo de envio de dados de forma transparente (Kandris; Nakas; Vomvas; Koulouras, 2020).

O consumo de energia é um dos conceitos mais discutidos em redes de sensores sem fio, devido à ampla variedade de problemas para os quais elas podem ser aplicadas. Esse contexto faz com que o desafio do consumo de energia varie consideravelmente e exija soluções específicas para cada situação. Nas subseções a seguir, serão descritas as principais características e classificações das redes de sensores sem fio em relação ao tipo de evento, tipo de sensoriamento e tipo de arquitetura de

rede utilizada (Kandris; Nakas; Vomvas; Koulouras, 2020). Essas descrições ajudarão na compreensão do trabalho proposto neste estudo.

2.1.1 Tipo de Evento sensoreado

É importante destacar que diversos tipos de eventos podem ser monitorados por redes de sensores sem fio. No entanto, cada tipo de evento apresenta seus próprios desafios e requer estratégias específicas para solucionar ou contornar esses problemas, como discutido por Genta et al. (Genta; Lobiyal; Abawajy, 2019). Essa diferenciação é fundamental para compreender como as redes de sensores sem fio são aplicadas em cenários diversos e como suas características podem ser adaptadas para atender a diferentes necessidades de monitoramento.

2.1.1.1 *Eventos simples*

Eventos simples são considerados neste contexto, eventos que não envolvem grandes volumes de dados, como medições de temperatura e umidade. Em uma WSN dedicada à agricultura de precisão, com foco no monitoramento do solo, a transmissão das informações sensoreadas pelos nós através da rede geralmente não afeta significativamente a vida útil de um nó ou da rede como um todo, uma vez que cada evento contém apenas alguns bytes de informação, conforme explicado por (Nabi; Jamwal; Padmanbh, 2022).

Portanto, uma estratégia comum adotada nesse contexto de redes com sensoriamento de eventos simples é o envio dos dados sensoreados pelos nós até o Sink (nó central) para ele executar todo o processamento das informações. Essa estratégia é preferida devido ao fato de o Sink geralmente não ter restrições significativas de energia. Dessa forma, o custo energético pago por cada um dos nós para transmitir as informações de eventos com poucos bytes até o Sink, mesmo por meio de múltiplos saltos, torna-se mais vantajoso do que realizar o processamento desses dados localmente (Nabi; Jamwal; Padmanbh, 2022)

2.1.1.2 *Eventos Multimídia*

Eventos multimídia estão relacionados ao tipo dos dados coletados pelos nodos sensores de uma rede, isto é, um evento multimídia é aquele que possui um dado multimídia em sua essência. Deste modo, o dado multimídia em questão pode ser, por exemplo, uma foto ou um áudio qualquer capturado por um nodo sensor da rede (Genta; Lobiyal; Abawajy, 2019).

Na área das redes de sensores multimídia sem fio, o tratamento dos eventos sensoreados assume uma considerável complexidade devido ao grande volume de dados gerados. Um único evento pode conter vários kilobytes de informações. Em redes de sensores sem fio alimentadas por bateria, o gerenciamento de energia deve ser

cuidadoso e complexo. Portanto, a transmissão de uma grande quantidade de dados relacionados a um único evento pode não ser suportada pelo protocolo de rede adotado ou pode ser extremamente dispendiosa em termos de consumo de energia. Nesse contexto, é necessário considerar estratégias mais sofisticadas para processar e transmitir informações até o ponto central (Sink) (Genta; Lobiyal; Abawajy, 2019).

2.1.2 Tipo de sensoriamento

No que diz respeito ao tipo de sensoriamento adotado, as redes de sensores sem fio se subdividem principalmente em redes proativas e reativas. Essa denominação é baseada na forma como a rede, ou mais especificamente, os nós sensores, reagem aos eventos ao seu redor. Portanto, cada um desses tipos de sensoriamento pode se adaptar melhor a problemas específicos, mas ambas as estratégias podem ser utilizadas com sucesso sem restrições significativas (Kandris; Nakas; Vomvas; Koulouras, 2020).

2.1.2.1 *Pro ativo*

Em redes de sensores proativas existe normalmente a configuração da rede para que haja uma periodicidade onde os eventos, isto é, em que a coleta de dados será executada, uma vez por minuto, hora ou dia, por exemplo. Esta estratégia de periodicidade é benéfica pelo fato de os nodos poderem ser quase que totalmente desligados, consumindo um mínimo de energia possível, aumentando assim a vida útil de cada nodo sensor e consequentemente da rede como um todo. Em meios que se pode mapear com precisão a periodicidade dos fenômenos a serem monitorados, esta pode ser uma estratégia bastante eficaz na resolução do problema em questão (Deebak; Al-turjman, 2020). Porém, nem todos os eventos podem ser mapeados temporalmente de modo que não se sabe quando ele irá ocorrer, gerando o problema da possível perda de informação caso este ocorra quando os nodos da rede estiverem desligados (Deebak; Al-turjman, 2020).

2.1.2.2 *Reativo*

Uma rede de sensores sem fio reativa está sempre ativa, ao menos uma parte dela, a qual é responsável por desempenhar o papel de sensoriar os eventos ao seu redor e notificar o nodo sensor e a rede como um todo. Esta parte sempre ativa da rede ou nodo pode ser, por exemplo, um sensor de passagem onde quando algo transpassa, é imediatamente detectado acionando o restante da rede ou somente o nodo sensor em questão. Este tipo de estratégia tem como principal vantagem sempre ou quase sempre, detectar os eventos ao seu redor, já que de certa forma está sempre ativa sensoriando os eventos em seu meio (Deebak; Al-turjman, 2020).

As redes reativas são um tipo de Rede de Sensores Sem Fio (RSSF) que se con-

centram na detecção e resposta a eventos específicos no ambiente em tempo real. Nesse tipo de rede, os nós sensores são configurados para coletar e/ou transmitir dados somente quando ocorre um evento específico. Uma rede reativa pode ser aplicada a diversos problemas e ambientes, no entanto, devido ao fato de normalmente envolver a operação contínua de sensores ativos, que consomem energia para o monitoramento de eventos, essa estratégia geralmente resulta em um custo energético mais elevado em comparação com uma rede proativa. No entanto, atualmente, existem diversas estratégias e algoritmos que visam reduzir o consumo de energia dessas redes e aumentar sua vida útil (Deebak; Al-turjman, 2020).

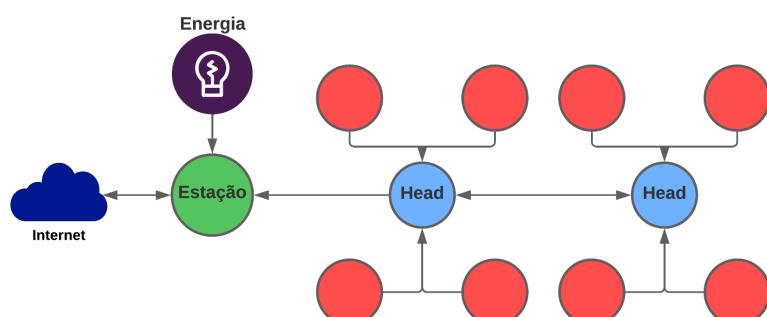
2.1.3 Arquitetura de rede

As redes de sensores sem fio geralmente apresentam dois tipos principais de arquitetura de rede: a organização dos nós em clusters e a arquitetura multihop. Cada uma dessas arquiteturas pode oferecer vantagens diferentes em relação à outra, e a escolha entre elas pode depender principalmente do ambiente em que a rede será implantada ou do problema que ela pretende resolver (Kandris; Nakas; Vomvas; Koulouras, 2020).

2.1.3.1 Cluster

Em uma rede de sensores sem fio baseada em clusters, os nós sensores da rede são organizados em grupos menores chamados de clusters. Em cada um desses grupos de nós, um dos nodos assume a função de transmitir os dados sensoriados pelos demais nodos do grupo. Esse nodo é comumente chamado de "cluster head." A principal função do cluster head é receber as informações sensoriadas por cada um dos nodos do seu cluster e, em seguida, encaminhar esses dados para outros nós da rede ou para a estação base (Mohan; Subramani; Alotaibi; Alghamdi; Khalaf; Ulaganathan, 2022).

Figura 2 – Rede de sensores sem fio organizada em cluster.



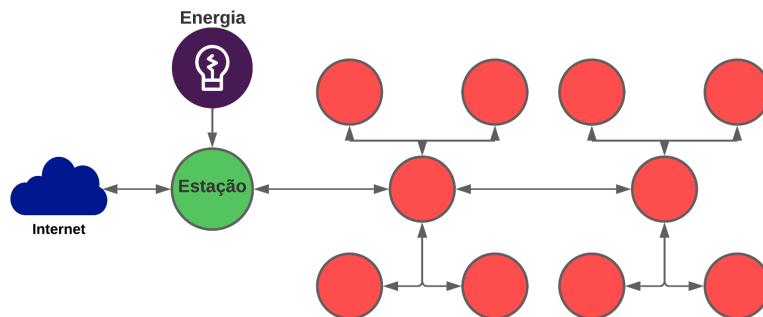
Fonte: Elaborada pelo autor.

A Figura 2 ilustra uma rede de sensores sem fio simplificada organizada em clusters. Nesta representação, é possível identificar dois clusters, cada um deles com um cluster head responsável por receber as informações dos nós do seu cluster e transmiti-las para o cluster head vizinho ou para a estação base. Alguns estudos têm indicado melhorias significativas no consumo de energia e na escalabilidade da rede quando a organização em clusters é utilizada, em comparação com uma arquitetura multihop padrão. No entanto, é importante destacar que esses resultados podem variar dependendo do problema específico para o qual a rede de sensores está sendo aplicada (Mohan; Subramani; Alotaibi; Alghamdi; Khalaf; Ulaganathan, 2022).

2.1.3.2 Mult Hop

A organização de redes de sensores sem fio em *MultiHop* permite que todos os nodos transmitam ou recebam informações, de modo a facilitar a comunicação entre os nodos da rede. Cada um dos nodos pode enviar as informações de cada sensoramento ou evento detectado a um destino final, passando os dados por nodos intermediários até atingir o nodo destino desejado. Usualmente, o nodo destino é a estação base ou Sink, a qual poderá executar o processamento dos dados coletados e transmitidos pelos nodos (Mohan; Subramani; Alotaibi; Alghamdi; Khalaf; Ulaganathan, 2022).

Figura 3 – Rede de sensores sem fio organizada em MultiHop.



Fonte: Elaborada pelo autor.

A Figura 3 representa uma configuração simplificada de uma arquitetura *MultiHop* em redes de sensores sem fio. Pode-se identificar nesta figura que todos os nodos têm um canal de comunicação, seja direta ou indiretamente por meio do repasse da comunicação por algum outro nodo intermediário, podendo existir mais de uma rota de comunicação possível. Este tipo de arquitetura de rede possibilita diversas rotas para um mesmo ponto, dependendo dos raios de comunicação dos nodos. Assim, caso algum nodo morra, outros podem assumir o seu lugar, atuando como um novo caminho ou rota para o envio das informações de sensoriamento vindas de uma dada região.

Não somente em situações de falha dos nodos, mas a estratégia de *MultiHop* se torna muito eficaz em meios que possuam algum tipo de obstáculo para o envio direto dos dados por algum sensor até a estação base, possibilitando que este repasse esta informação para qualquer um de seus vizinhos que ficará encarregado de encaminhar a mesma para o destino final desejado (Mohan; Subramani; Alotaibi; Alghamdi; Khalaf; Ulaganathan, 2022).

2.2 Infraestrutura de Simulação

O Eboracum é um framework usado para projetar e avaliar Redes de Sensores Sem Fio (RSSF), permitindo a modelagem e simulação dessas redes em um alto nível de abstração. O objetivo é a avaliação da eficiência energética de soluções voltadas para uma aplicação específica. Por isso, o simulador propõe a modelagem não só dos nodos e das suas conexões, mas também da carga de trabalho a qual a rede será submetida.

Essa infraestrutura foi desenvolvida com o uso das classes do Ptolemy¹ visando assim prover um ambiente para modelar RSSF reativas. O framework foi implementado em Java, onde a representação dos componentes do cenário a ser simulado é feita por meio de classes. Para criar um componente, basta instanciar ou estender essas classes.

Em uma RSSF, dentre os principais componentes podem-se citar os nós sensores e a estação base (ou Sink). No Eboracum, esses componentes são representados pelas classes *SimpleWSNNode* e *NetworkMainGateway*, respectivamente. As instâncias dessas classes possuem capacidade de comunicação, onde um parâmetro define o alcance da comunicação. No entanto, as instâncias de *SimpleWSNNode* também possuem uma carga de bateria, que é inicializada com a capacidade total da bateria, enquanto *NetworkMainGateway* não precisa gerenciar o consumo de energia, pois não possui bateria. Os nós sensores também têm a capacidade de detectar eventos externos, onde um parâmetro define o alcance do sensor. Além disso, as instâncias de *SimpleWSNNode* têm a capacidade de processar os dados detectados e enviar uma mensagem resultante pela rede para o Sink, usando outros nós da rede para retransmissão, se necessário.

A classe *SimpleWSNNode* estende a classe *BasicWirelessSensorNode*, que representa um nó sensor e possui uma CPU interna para processar tarefas associadas aos eventos capturados. A CPU é definida por meio de uma interface que especifica alguns métodos para representar o comportamento geral de uma CPU. Para o *SimpleWSNNode*, existe uma implementação dessa interface chamada *SimpleFIFO-BasedCPU*, que descreve o comportamento de uma CPU baseada em FIFO.

¹<https://ptolemy.berkeley.edu>

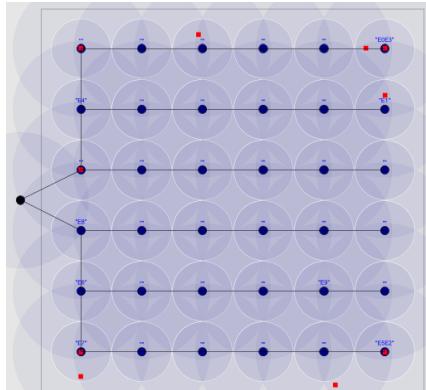
O simulador com base nos raios de comunicação monta uma rede ad hoc, determinando para cada nodo o caminho mais curto até a estação base (ou sink). A medida em que nodos ficam inativos, devido a descarga de suas baterias ao longo da simulação, esta rede pode ser remontada e novas conexões serão definidas.

O simulador oferece uma visualização gráfica do cenário da rede a ser simulada, representando nodos, e as conexões entre os nós, como pode ser observado na Figura 4. O círculo pintado representa o nodo e tem duas circunferências sombreadas no seu entorno. Neste exemplo, a circunferência menor representa o raio de sensoriamento e a maior o raio de comunicação. Para que dois nós se comuniquem é necessário que seus raios de comunicação tenham sobreposição.

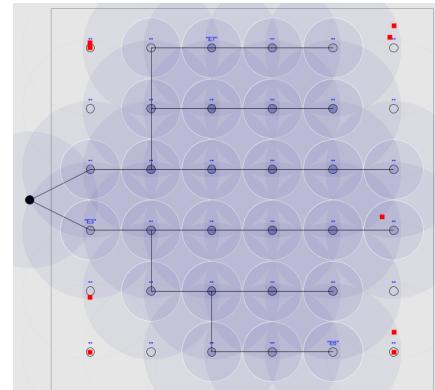
No estágio inicial da simulação, se os nodos estiverem posicionados em grade e houver cada nodo até três vizinhos comunicáveis em um salto, teremos conexões como a ilustrada na Figura 4a. Neste esquema de roteamento multi-hop todos os nodos estão conectados ao Sink direta ou indiretamente, dependendo do alcance do seu canal de comunicação.

Figura 4 – Simulação com interface gráfica no Eboracum.

(a) Inicio de Simulação



(b) Simulação parcial



Fonte: Elaborada pelo autor.

Já a Figura 4b representa um estágio posterior de simulação, quando alguns nodos já ficaram inativos. Visualmente, uma coloração mais fraca indica que os nodos estão com menos bateria. Também é possível visualizar nodos totalmente inativos, que são representados sem o rádio de sensoriamento e de comunicação no seu entorno. Após a inatividade destes nodos, as conexões foram remontadas pelo Eboracum para manter assim a rede ativa por mais tempo (Marques; Brisolara; Ferreira; Indrusiak, 2016). Todos os experimentos executados neste trabalho usaram a funcionalidade da remontagem da rede após a morte de nodos, embora este seja um parâmetro que possa ser desabilitado na representação do modelo de simulação.

A avaliação do consumo de energia da rede no Eboracum é calculada com base na

descarga da bateria dos nós, que varia de acordo com a carga de trabalho associada aos eventos acionados pelo simulador. Essa descarga de energia é modelada em três diferentes modos de operação para os nós sensores: ocioso, processando e transmitindo. Os custos de energia relacionados a esses modos de operação caracterizam a simulação de uma RSSF dedicada a uma aplicação específica.

As simulações realizadas no Eboracum são baseadas em eventos discretos, o que permite acelerar significativamente o tempo de execução de cada simulação, uma vez que as atualizações ocorrem somente quando ocorre um evento desencadeador. Nesse modelo, a carga de trabalho da aplicação é representada por eventos que ativam os atores, que são os sensores da rede.

O Eboracum oferece diversas classes para a simulação de eventos, permitindo representar fenômenos naturais de várias maneiras, levando em consideração aspectos temporais e espaciais dos eventos. Por exemplo: A classe *PeriodicEvent* modela a ativação dos nós de forma periódica, com base em intervalos de tempo definidos; A classe *StochasticJumperEvent* representa eventos distribuídos no espaço de acordo com modelos estocásticos; *StochasticDailyJumperEvent* representa eventos distribuídos no espaço e no tempo de acordo com modelos estocásticos, sendo que as distribuições de tempo consideram um período de 24h.

As classes que recebem no nome *Stochastic* permitem uma modelagem mais precisa e realista de fenômenos naturais, levando em consideração modelos probabilísticos. Muitos fenômenos naturais, como infestações de pragas em lavouras, seguem padrões estocásticos que podem ser representados por modelos baseados em distribuições de probabilidade. No primeiro estudo de caso apresentado neste trabalho, foi empregada a classe *StochasticDailyJumperEvent*, que simula o comportamento das moscas-das-frutas em pomares. Essa classe utiliza modelos estocásticos para representar o comportamento das moscas, gerando eventos concentrados no final do dia. Isso reflete o comportamento desses insetos, que tendem a sair em busca de alimentos no final da tarde.

3 TRABALHOS RELACIONADOS

Neste presente estudo, trabalhos relacionados que abordam o balanceamento de carga em redes reativas foram identificados. Destes trabalhos identificados, dos que faziam o processamento dos eventos localmente, ou recorriam a algoritmos bio inspiradas, ou aplicavam técnicas de otimização de restrições de modo a reduzir o número de comunicações, ou selecionar o melhor nodo para executar o processamento de um evento detectado. Abaixo será apresentado um resumo dos principais trabalhos tomados como base para a elaboração deste estudo.

Uma estratégia baseada no comportamento de insetos sociáveis, denominado PS (do inglês, Pheromone-Signaling) foi proposta em (Caliskanelli; Harbin; Indrusiak; Mitchell; Polack; Chesmore, 2013). Na técnica PS, os nodos utilizam apenas as informações disponíveis localmente para tomar a decisão de realizar ou não o processamento do evento detectado pelo nó. Esta técnica foi proposta para reduzir a redundância em redes de sensores compostas por nós com raios de detecção sobrepostos, assim o algoritmo decide qual nó irá processar o evento entre os nós que detectaram o mesmo evento. O desempenho do PS depende do nível de sobreposição entre os sensores. Pois, o esquema empregado para balancear a carga nesta técnica baseia-se em habilitar o nodo a fazer processamento, ou não, baseado na troca de feromônios com nodos vizinhos e prover uma alternância entre nodos habilitados e não habilitados. Se a sobreposição dos raios de sensoriamento for limitada, aumenta-se a chance de eventos serem perdidos.

No (Ferreira; Brisolara; Indrusiak, 2015), é proposta uma técnica baseada no comportamento de formigas, onde os nodos decidem probabilisticamente, quem irá processar um evento. Nessa abordagem, os eventos produzem estímulos para os nós que, a partir de seus limiares internos, determinam diferentes probabilidades de processamento desses eventos. Quando os nós detectam um evento, cada nó decidirá se deve ou não processar esse evento considerando sua probabilidade interna computada. Em resumo, essa abordagem funciona modificando a probabilidade de um nó processar um evento conforme o número de nós que detectaram esse evento em simultâneo, e o número de vezes que esse nó processou eventos anteriores. Em com-

paração apresentada em (Ferreira; Brisolara; Indrusiak, 2015), a técnica bio inspirada no comportamento de formigas obteve resultados muito próximos aos obtidos pelo PS, com a vantagem de seu desempenho não depender de um alto nível de sobreposição e não exigir a reconfiguração de parâmetros para manutenção da eficiência da técnica quando a densidade da rede é modificada.

Recentemente, uma técnica de balanceamento de carga semidistribuída foi proposta em (Pereira; Brisolara; Ferreira, 2020). Esta técnica é baseada nos algoritmos do Problema de Otimização de Restrições - COP. Com este tipo de algoritmo é possível definir, entre os nodos que detectaram o mesmo evento, o nodo mais adequado para processá-lo, sem a necessidade de muitas trocas de mensagens. Como resultado, os autores demonstraram que esta técnica proposta não há perda na detecção de eventos enquanto a rede está viva, apresentando ainda um desempenho superior em redes que possuem baixa densidade em comparação com PS (Caliskanelli; Harbin; Indrusiak; Mitchell; Polack; Chesmore, 2013) e a técnica bio inspirada no comportamento de formigas (Ferreira; Brisolara; Indrusiak, 2015).

O artigo (Saleem; Alabady, 2023) descreve o protocolo Energy-Efficient Multipath Clustering with Load Balancing (EEMCL) para Redes de Sensores Multimídia Sem Fio. Os autores Destacam a eficiência energética da estratégia apresentada em comparação com outros protocolos de clusters, com a presença de MCHs (nós principais de cluster) e nós sensores normais. O EEMCL utiliza 12 clusters, cada um com um MCH, sete nós sensores normais e dois nós sensores normais como SCHs (nós secundários de cluster). O protocolo apresentado possui quatro fases: inicialização, Clustering de Rede, formação de cluster e transmissão de dados. Durante a transmissão, os nós sensores normais enviam dados para os MCHs via SCHs, que agregam e comprimem os dados usando TDMA. A comunicação inter-cluster utiliza CSMA com multi-canal. O objetivo é equilibrar o consumo de energia, garantindo eficiência na transmissão e prolongando a vida útil da rede, através do roteamento baseado em clusters e protocolos multipath. O EEMCL seleciona clusterheads com mais energia, estabelecendo rotas intra e inter-cluster para otimizar a vida útil da rede.

No trabalho de (Varshovi; Kavian; Ansari-asl, 2019), é proposta a implementação de uma Rede de Sensores Multimídia sem fio (RSSFM) para detecção de movimento, utilizando coprocessamento local FPGA. A rede de sensores proposta realiza o processamento das imagens sensoreadas localmente, utilizando pequenos processadores FPGA para identificar possíveis movimentos e alterações nas imagens sensoreadas, transmitindo posteriormente os dados já processados para a estação base. No artigo, dois algoritmos de detecção de movimento foram implementados nas FPGAs Spartan 6, e os resultados locais da detecção, assim como o resultado global da rede, são apresentados. Para os testes, os autores destacam que, devido às limitações dos recursos do FPGA utilizados na implementação, foi necessário utilizar quadros

de vídeo de baixa resolução. Ainda enfrentando esse problema, o artigo sugere a adoção de FPGAs mais avançados e nós de rede mais recentes com transceptores mais poderosos para aprimorar o desempenho. Além disso, destaca a possibilidade de analisar os tipos de movimentos detectados através de um banco de dados contendo várias ações como uma melhoria potencial no futuro. Na conclusão, os autores enfatizam os desafios enfrentados e as soluções propostas na implementação de uma Rede de Sensores Multimídia sem fio para detecção de movimento, utilizando o co-processamento local FPGA.

O artigo de (Genta; Lobiyal; Abawajy, 2019) aborda desafios na redução do consumo de energia em Redes de Sensores sem Fio Multimídia (RSSFM). Propõe um algoritmo multipercorso energeticamente eficiente com clustering dinâmico baseado em Algoritmo Genético (GA). Esse algoritmo realiza o particionamento dinâmico da rede em clusters, seleciona líderes eficientes e implementa roteamento multicaminho otimizado pelo GA. A seleção dos Cluster Heads considera parâmetros como custo de energia, energia residual e distâncias. O GA representa o caminho do Cluster Head até o Sink por um cromossomo. A função de fitness maximiza a eficiência energética considerando distâncias mínimas e energia residual. Contribuições incluem um novo algoritmo de clustering e um multicaminho eficiente com base em GA. O desempenho é avaliado comparativamente com algoritmos de última geração para RSSFM. O artigo busca melhorar a eficiência energética e a vida útil da rede, apresentando uma abordagem inovadora para enfrentar os desafios específicos de RSSFM.

No trabalho de (Kethireddy; Rallapalli; Chilakala; Devulapalli, 2023), apresenta-se o CECIT (Coherent Energy Cooperative Image Transmission), uma solução desenvolvida para superar os desafios encontrados em redes de sensores sem fio multimídia. Estes desafios incluem o elevado consumo de energia associado à transmissão de grandes volumes de dados. O CECIT utiliza códigos QR para representar imagens enviadas à internet pelos nós, eliminando a necessidade de realizar processamentos adicionais, como a conversão para escala de cinza. Esta estratégia, considerada uma compressão eficiente, é alcançada por meio da Transformada Wavelet Discreta Bidirecional (DWT), que identifica redundâncias nos dados. Isso facilita a transmissão e reduz a carga de armazenamento, constituindo o processo de pré-processamento dos eventos. Além de reduzir a transmissão de dados, minimizando o consumo de energia, essa abordagem contribui para prolongar a vida útil da rede. Ao oferecer uma alternativa dinâmica e eficiente para a transmissão de imagens em Redes de Sensores Sem Fio (RSSF), o CECIT destaca-se como uma solução que atende às crescentes demandas tecnológicas e de estilo de vida, proporcionando maior satisfação aos usuários finais.

O artigo de (Abazeed; Faisal; Ali, 2019) aborda o conceito de multipath em Redes de Sensores Sem Fio para Multimídia, propondo um protocolo cross-layer que integra

informações de várias camadas. A estratégia proposta visa maximizar os caminhos para nós disjuntos, com o objetivo de reduzir a congestão e pontos de falha. Uma função heurística classifica as rotas, considerando métricas como atraso, energia restante e proporção de recepção de pacotes para a transmissão de multimídia. Além disso, prioriza pacotes com base em sua importância, sendo que cada parâmetro da função heurística possui um peso, otimizado por meio de uma busca exaustiva. A abordagem proposta, chamada Cross-Layer Multipath Routing (CLMR), restringe nós conjuntos, otimizando o valor da transmissão. O CLMR classifica os pacotes de multimídia em três tipos (I, P, B), permitindo a existência de pelo menos três caminhos do nó de origem até o destinatário. Alguns desses caminhos podem ser utilizados exclusivamente para o tráfego de pacotes com prioridade. O processo de criação de rotas envolve a coleta de informações dos vizinhos, a construção de uma tabela de vizinhos, a seleção de três próximos vizinhos para a descoberta de multipath e o envio de pacotes de controle de rota até alcançar o destino. Posteriormente, o destino calcula os custos dos caminhos e retorna essas informações ao nó emissor por meio de pacotes de controle de rota resposta.

O artigo de (Bavarva; Jani; Ghetiya, 2018) aborda o desafio do elevado consumo de energia durante os processos de transmissão, recepção e processamento. Os autores propõem a utilização de técnicas de múltiplas entradas e múltiplas saídas (MIMO) em conjunto com detecção de compressão (CS) para solucionar esses problemas. A estratégia MIMO mencionada pelos autores proporciona maior confiabilidade, permitindo o direcionamento de pacotes por várias rotas distintas. Além disso, esses pacotes podem ser destinados por rotas mais confiáveis, com maior disponibilidade de energia. Outra estratégia abordada é a técnica de compressão CS, capaz de reduzir o tamanho da imagem com baixo consumo de energia de processamento. Essa técnica ainda mantém a imagem criptografada, uma vez que apenas uma imagem em formato de ruído é enviada pela rede. Quando o Sink recebe essa informação, executa o processo inverso conhecido, removendo o ruído da imagem. É importante ressaltar que esse processo gera algumas perdas, mas, conforme mencionado pelos autores, a qualidade da imagem perdida é insignificante quando comparada ao custo energético economizado no envio da imagem. Além disso, destaca-se a segurança adicional agregada aos dados.

O artigo de (Chiwariro; Thangadurai, 2023) concentra-se em abordar o congestionamento de rede e aprimorar o desempenho em redes de sensores multimídia sem fio. Para resolver esse desafio, é introduzido um mecanismo de classificação de pacotes baseado em prioridades, utilizando inteligência artificial (IA) para extrair a semântica de cada pacote e decidir qual deles terá uma prioridade mais alta. Outra estratégia discutida pelos autores é o uso de múltiplas rotas, em que uma heurística desenvolvida no estudo utiliza as prioridades estabelecidas pela IA para determinar a rota que

cada pacote seguirá, seja ela mais rápida ou energeticamente menos custosa. Nos testes conduzidos pelos autores, os resultados obtidos são promissores. A abordagem conseguiu reduzir a perda de pacotes por meio da otimização de rotas, além de diminuir o atraso e o consumo de energia na rede.

Dentre os trabalhos identificados durante o levantamento bibliográfico, observou-se que algumas dessas pesquisas se concentraram em propor abordagens para redes de sensores sem fio com intersecção de sensoreamento. Nestas análises, as estratégias propostas tinham como objetivo tomar a melhor decisão entre os nós que detectaram o evento, determinando qual deles seria responsável por executar o processamento correspondente (Pereira; Brisolara; Ferreira, 2020), (Ferreira; Brisolara; Indrusiak, 2015), (Caliskanelli; Harbin; Indrusiak; Mitchell; Polack; Chesmore, 2013). No entanto, essas estratégias não incorporaram qualquer mecanismo de balanceamento de carga ou divisão de trabalho entre os nós. Além disso, nenhuma das abordagens mencionadas levou em consideração a perda de pacotes nas transmissões da rede, nem explorou o cenário de um meio sem intersecção de sensoreamento.

Outra parcela significativa dos trabalhos identificados buscou aprimorar a tomada de decisões em relação às rotas e estratégias de rede, concentrando-se em escolher a rota mais vantajosa para transmitir um evento através da rede (Saleem; Alabady, 2023), (Chiwariro; Thangadurai, 2023), (Genta; Lobiyal; Abawajy, 2019), (Abazeed; Faisal; Ali, 2019), (Bavarva; Jani; Ghetiya, 2018). Diversos algoritmos e estratégias foram implementados na seleção dessas rotas, incluindo abordagens baseadas em algoritmos genéticos ou em inteligência artificial (Chiwariro; Thangadurai, 2023), (Genta; Lobiyal; Abawajy, 2019). Essas abordagens, muitas vezes, permitiam a classificação por prioridades dos pacotes, visando reduzir a latência no envio de determinados dados. No entanto, em sua maioria, esses trabalhos não contemplaram processamento local dos dados, tampouco adotaram mecanismos de balanceamento ou compartilhamento de tarefas entre os nós. Além disso, os estudos não consideraram a perda de pacotes na rede.

Outros trabalhos identificados propuseram abordagens com ênfase no processamento local (Kethireddy; Rallapalli; Chilakala; Devulapalli, 2023), (Varshovi; Kavian; Ansari-asl, 2019). Em (Varshovi; Kavian; Ansari-asl, 2019), o processamento local dos dados é otimizado por hardware dedicado, enquanto em (Kethireddy; Rallapalli; Chilakala; Devulapalli, 2023), a abordagem envolve o pré-processamento local de imagens. Nestas abordagens, nenhuma delas implementou qualquer forma de compartilhamento ou divisão de tarefas entre nós vizinhos. Além disso, nenhuma das abordagens considerou possíveis perdas de dados durante o envio das informações.

A Tabela 1 apresenta uma análise comparativa entre as abordagens discutidas na literatura, revisadas nesta seção, e a abordagem proposta, TDMEN. Esta análise considera diversos aspectos relevantes para aplicações em Redes de Sensores Sem Fio

Tabela 1 – Trabalhos relacionados.

	Sem Intersecção	Com Intersecção	Considera Perda de Pacotes	Processamento Local	Executa Distribuição de Tarefas
PEREIRA et al., 2020		✓		✓	
FERREIRA et al., 2015		✓		✓	
CALISKANELLI et al., 2013		✓		✓	
SALEEM et al., 2023		✓			
CHIWARIRO et al., 2023	✓				
GENTA et al., 2019	✓				
ABAZEED et al., 2019		✓			
BAVARVA et al., 2018	✓				
KETHIREDDY et al., 2023		✓		✓	
VARSHOVI et al., 2019	✓			✓	
TDMEN	✓	✓	✓	✓	✓

Fonte: Elaborada pelo autor.

Multimídia (RSSFM), incluindo a sobreposição de sensores (redundância espacial), perda de pacotes, bem como questões relacionadas ao processamento e à distribuição de tarefas entre nodos vizinhos.

O estudo revelou uma carência de abordagens que considerem simultaneamente as duas configurações possíveis de redes de sensores sem fio multimídia (com e sem sobreposição). Além disso, nenhum dos trabalhos revisados discutiu perda de pacotes e sua influência no balanceamento de carga proposto. Esta tese se destaca por ser a única a abordar esses três aspectos em seus estudos de caso. Outro diferencial da abordagem TDMEN, proposta neste trabalho, é o fato desta expandir o conceito de balanceamento de carga ao permitir a divisão de uma tarefa computacional em grãos menores a serem distribuídos entre nodos vizinhos, possibilitando a cooperação no processamento de uma mesma tarefa. Esta abordagem reduz a granularidade do balanceamento da carga de processamento, e consequentemente amplia as alternativas para obtenção de uma descarga mais homogênea.

4 BALANCEAMENTO DE CARGA BASEADO EM DIVISÃO DE TAREFAS

Redes de sensores reativas podem gerar cargas distribuídas de forma muito heterogênea nos nodos sensores, podendo levar a morte prematura de certos nodos. Esta heterogeneidade é mais presente em redes reativas, onde a ocorrência do evento, gera a necessidade da captura e consequente processamento e /ou comunicação (Hussien; Matloob, 2018). A medida em que estas redes podem empregar sensores com captura de dados multimídia, a carga de trabalho aumenta de forma representativa, podendo produzir demanda por mais comunicação ou mais processamento aos nodos.

Em um evento multimídia, comumente há possibilidade de se dividir o dado capturado em áreas de interesse, como, por exemplo, quadrantes de uma imagem, tempo de um som, etc (Long; Cao; Jiang; Zhang, 2017). Esta segmentação permite a execução do processamento dos dados de um evento multimídia de forma cooperativa, onde diferentes nodos processam parte dos dados (Long; Cao; Jiang; Zhang, 2017). O presente trabalho explora esta possibilidade de divisão da tarefa de processamento em subtarefas e propõe a distribuição destas subtarefas entre os nós vizinhos, buscando equilibrar a carga de trabalho destinada a cada nó e consequentemente alcançar um consumo de energia na rede mais uniforme. A divisão de tarefas leva em consideração o nível de bateria dos nós, o número de subtarefas processadas pelos nós e a disponibilidade dos nós. Um protocolo de comunicação, que visa facilitar a execução da divisão e compartilhamento das tarefas entre os nós, também foi proposto. A divisão de tarefas em RSSFM foi tratada em três abordagens distintas: TDMEN, TDMEN-CACHE e TDMEN-INTERSEC.

A abordagem TDMEN endereçou RSSFM sem sobreposição das áreas de sensoriamento, onde um mesmo evento só pode ser detectado por um único nodo. A Seção 4.1 detalha esta abordagem. A abordagem TDMEEn-CACHE representa uma variação da primeira que explora a questão da redundância temporal evitando processar repetidamente a mesma informação. Esta abordagem será detalhada na Seção 4.2. A abordagem TDMEN-INTERSEC aborda as RSSFM com sobreposição nas áreas dos

sensores, ou seja com redundância espacial, e será apresentada na Seção 4.3. Todas elas foram implementadas no simulador Eboracum, detalhes de implementação são apresentados nas seções correspondentes.

Por fim, na Seção 4.4 é proposto um modelo matemático de perdas de pacote estimado para redes de sensores sem fio que fazem uso do protocolo ZigBee e rádios S2C, que também foi integrar ao simulador Eboracum visando uma simulação mais próxima do mundo real.

4.1 TDMEN: Divisão de tarefas para eventos Multimídia em Redes de Sensores Sem Fio

A presente abordagem proposta, denominada TDMEN (*Task Division for Multimedia Events in Wireless Sensor Networks*) aborda o balanceamento de carga em RSSF Multimídia reativas. A estratégia é dividir a carga de processamento associada a um evento entre os nodos vizinhos, considerando o nível de bateria dos nós, o número de subtarefas processadas por estes e a disponibilidade dos mesmos. Nesta seção, é detalhado como a abordagem TDMEN distribui a carga de processamento desde a detecção de um evento até o envio da mensagem resultante para o Sink.

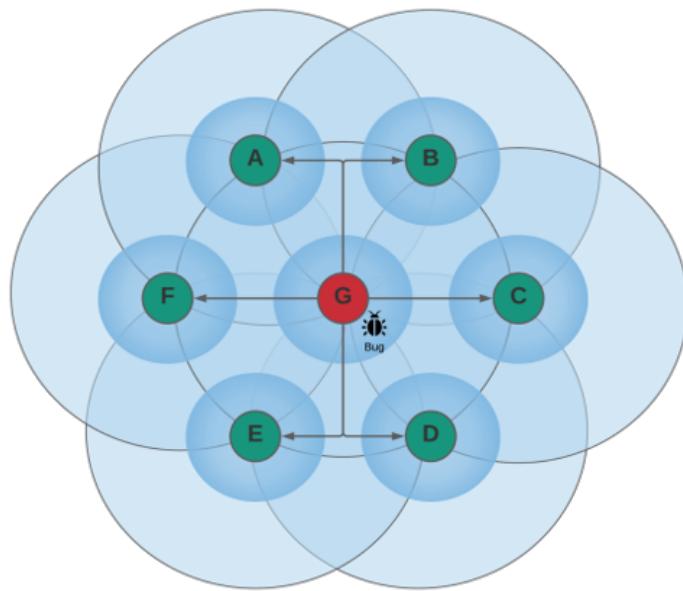
A distribuição da carga é baseada em um esquema de comunicação. Este esquema de comunicação inicia quando um evento é detectado por um nó, fazendo com que este envie uma mensagem *broadcast* para todos os seus vizinhos imediatos, realizando assim o reconhecimento de sua vizinhança. A Figura 5 ilustra um cenário composto por cinco nós denominados A-G, com raio sensoriamento de eventos (círculo menor) e raio de cobertura de comunicação (círculo maior) onde o alcance deste raio define quais serão seus vizinhos. Neste cenário de exemplo, o nó G detecta um evento e depois transmite uma mensagem de *broadcast* para reconhecer seus vizinhos. Neste contexto, o nodo que acabou de detectar um evento é denominado de nodo mestre. No exemplo ilustrado, o nodo G é o nodo mestre.

Como resposta ao nodo mestre (i.e. nodo G), os nodos vizinhos enviam uma mensagem contendo as seguintes informações: seu nível atual de bateria; seu nome identificador exclusivo; número de segmentos processados; estado atual que indica se já está processando ou não uma tarefa. Assim, com essas informações, o nodo Mestre (nó G, neste exemplo) monta uma tabela com os dados recebidos de todos os seus vizinhos imediatos.

A Figura 6 ilustra este esquema de comunicação usando como exemplo o cenário apresentado na Figura 5, onde o nó G detectou um evento e o nó A é um de seus vizinhos. A Figura 6 detalha a sequência de mensagens trocadas entre eles. Inicialmente, o nó G envia uma mensagem *Hello* em *broadcasting* para seus vizinhos imediatos (A-F). O nó A recebe esta mensagem identificada pela *tag Hello* e como resposta, o nó A

envia uma mensagem para G, que inclui sua carga de bateria, número de segmentos processados e seu status. Esta mensagem de resposta, marcada como *Hey*, foi simplificada na Figura 6 mostrando apenas a bateria. Quando o nó Mestre (nó G, neste exemplo) recebe uma mensagem *Hey*, ele atualiza as informações de A na tabela de vizinhos. Após, tendo as informações atualizadas de todos os vizinhos, G usa essas informações para decidir para qual de seus vizinhos irá distribuir tarefas. No exemplo ilustrado, G envia uma mensagem *Segment* para A, enviando os dados do segmento para serem processados pelo vizinho. O nó A ao receber um *Segment*, processa-o e depois envia uma mensagem *OK* para G incluindo informações sobre a carga atual de sua bateria e o resultado do processamento do segmento recebido.

Figura 5 – Exemplo de RSSF Sem sobreposição de sensoriamento: Interação entre nodos logo após sensoriamento de um evento.

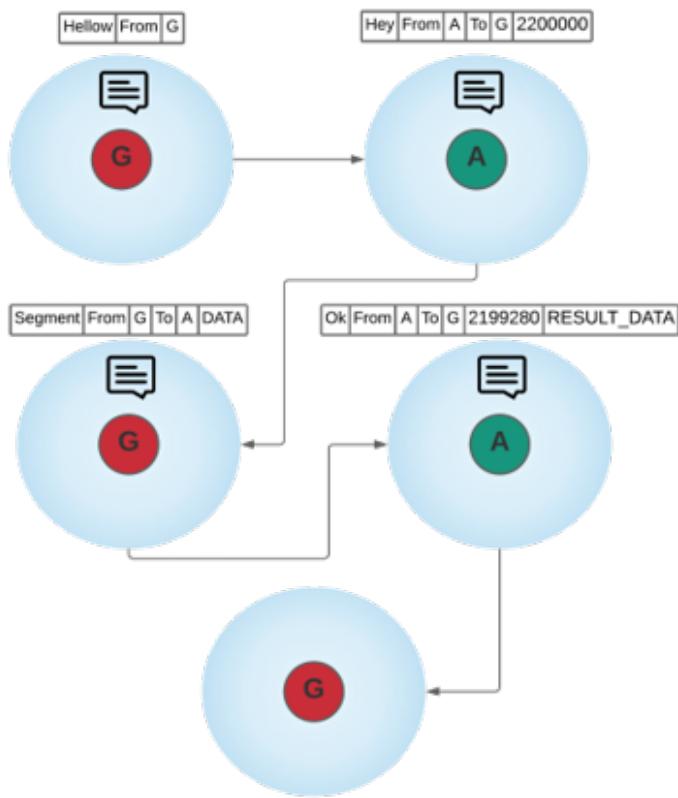


Fonte: Elaborada pelo autor.

Para efetuar a divisão do processamento, o nodo Mestre executa um pré-processamento inicial das informações capturadas, esta tarefa tem por objetivo segmentar o dado multimídia. Esta segmentação viabilizará a divisão do processamento em subtarefas e sua distribuição entre nodos vizinhos.

Uma ilustração simplificada do processo de segmentação pode ser vista na Figura 7, onde o estado A representa uma foto capturada por uma armadilha de insetos. Essa armadilha visa identificar os insetos que ficarem presos nela. Deste modo, a etapa de preprocessamento desta armadilha visa identificar os pontos de interesse na imagem capturada, segmentando a imagem em quadrantes de interesse, onde esta segmentação é ilustrada pelo estado B.

Figura 6 – Exemplo de RSSF Sem sobreposição de sensoriamento: Fluxo de comunicação entre nó mestre e vizinho.



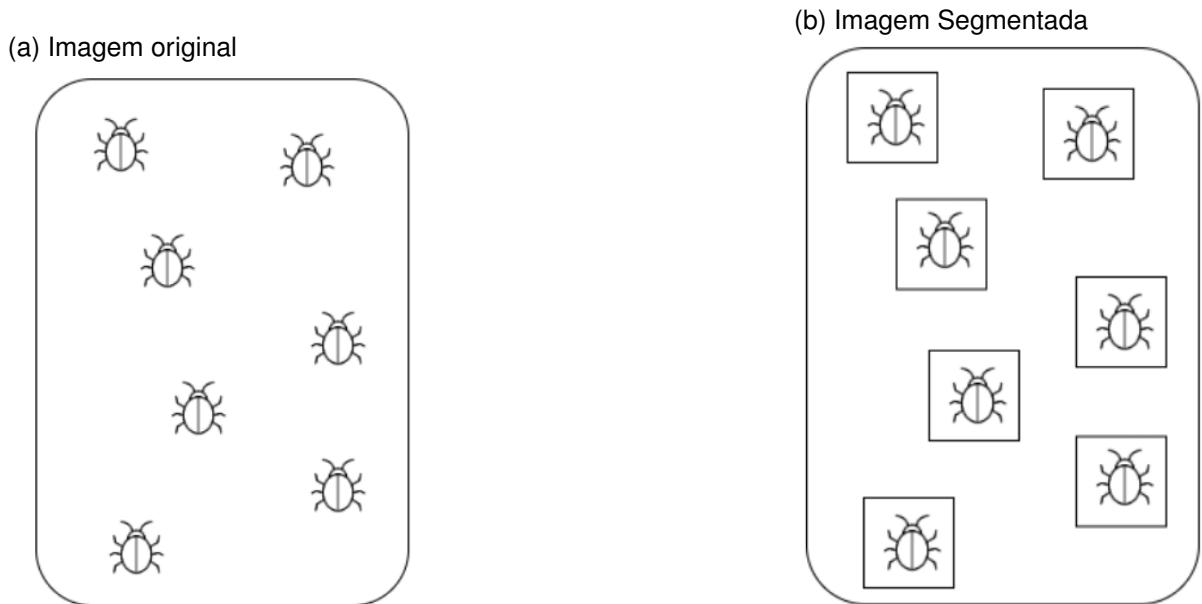
Fonte: Elaborada pelo autor.

Com a segmentação finalizada, o nodo Mestre usa a tabela de informações criada no momento do sensoriamento para tomar a decisão referente a quais nodos serão enviados os segmentos. Os vizinhos elegíveis a receberem segmentos para processar são ordenados com base principalmente na quantidade de bateria que os mesmos possuem e se estão ocupados ou não. Nodos ocupados não são considerados elegíveis, bem como nodos com níveis críticos de bateria que caso executassem o processamento de um segmento ficariam inativos pouco tempo depois.

Cada vizinho elegível do nodo Mestre recebe um segmento por vez. Quando um destes vizinhos finaliza o processamento, ele sinaliza ao Mestre enviando o resultado do processamento, assim como as informações atualizadas de bateria após executar a tarefa. Com o resultado do processamento do segmento e a tabela de informações atualizada, o nodo Mestre, caso ainda possua segmentos para ser processados, pode destinar mais segmentos para um mesmo vizinho processar, caso este ainda seja considerado elegível.

Esta abordagem é indicada quando a RSSFM não tem áreas de sobreposição de sensoriamento entre os nodos. A TDMEN não trata esta redundância espacial,

Figura 7 – Processo de Segmentação de imagem.



Fonte: Elaborada pelo autor.

podendo gerar mais de um nodo mestre e assim, apesar de dividir a carga, esta carga poderá estar sendo replicada, pois mais de um nó pode detectar o mesmo evento e disparar a distribuição.

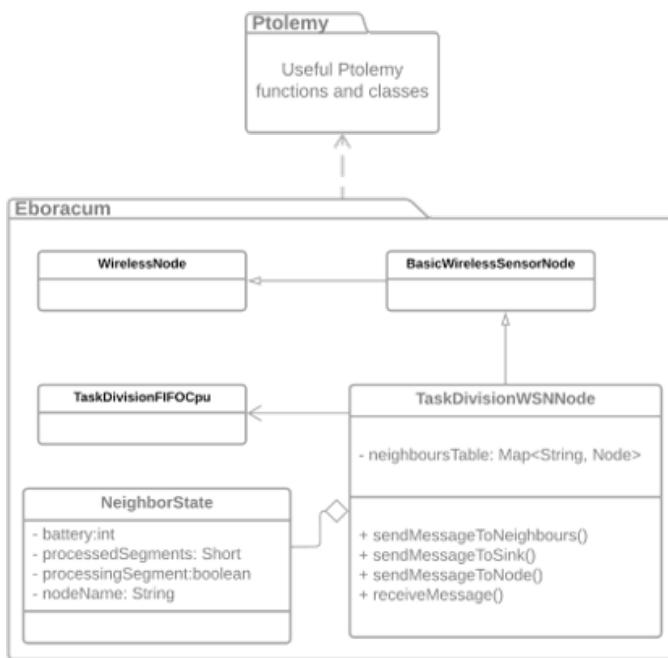
Para a implementação da abordagem TDMEN, fez-se uso do Framework Eboracum, ferramenta que tem sido usada em diversos estudos do presente grupo de pesquisa. Como o Eboracum é inteiramente desenvolvido em Java, a criação de novos comportamentos segue basicamente a criação de novas classes, as quais devem estender classes-base do Eboracum. Assim, para desenvolver a TDMEN, foram adicionadas três novas classes à infraestrutura do Eboracum: *TaskDivisionWSNNode*, *NeighborState* e *TaskDivisionFIFO*.

A *TaskDivisionWSNNode* é a principal responsável pela execução do fluxo de comunicação e controle da divisão das tarefas nesta abordagem, distribuindo os segmentos para os demais nodos. A classe *NeighborState* é usada para armazenar informações sobre os nodos. Estas informações serão usadas na tomada de decisão quanto à distribuição dos segmentos. Cada nodo possui uma lista de instâncias da classe *TaskDivisionWSNNode* a qual representa as informações de cada um dos vizinhos deste nodo. As informações desta lista são atualizadas durante as comunicações executadas entre os nodos.

Por fim, a classe *TaskDivisionFIFO* representa a fila de processamento do nodo, onde as tarefas ficam armazenadas e retiradas na ordem que chegam FIFO. Na Figura 8 é apresentado o diagrama de classes que sintetiza a implementação,

onde pode-se identificar as três classes desenvolvidas, bem como as classes-base do Eboracum que as mesmas derivam. A *TaskDivisionWSNNode* é uma subclasse da *BasicWirelessSensorNode*, classe que representa os nodos na modelagem da RSSF. Uma instância de *TaskDivisionWSNNode*, nodo na abordagem proposta, conterá a tabela de vizinhos (*neighboursTable*), além de estar associado a uma instância de *TaskDivisionFIFOcpu* e a N instâncias de *NeighborState*, para representar as subtarefas na sua fila e seu estado, correspondentemente.

Figura 8 – Diagrama de classes da implementação da abordagem TDMEN.



Fonte: Elaborada pelo autor.

O Algoritmo 1 apresenta um pseudocódigo da estratégia TDMEN proposta por este estudo. Este código simplificado mostra a sequência básica executada por cada um dos nós da rede. Na linha 1, o algoritmo verifica se o nó ainda possui bateria para executar tarefas; já na linha 2, o fluxo de execução aguarda a detecção de um evento.

Nas linhas 3 e 4 o algoritmo verifica se um evento foi detectado. Caso afirmativo, na linha 4 é feita a notificação dos vizinhos do nó mestre por meio de uma tag, indicando o sensor que fez a detecção e incluindo informações básicas sobre a identificação do evento na mensagem.

Na linha 5 o algoritmo verifica se recebeu alguma mensagem de seus vizinhos. Se tiver recebido, ele identifica, na mensagem usando a tag, se é uma mensagem para processar um segmento, uma mensagem de notificação(hey, hellow) ou ainda se é uma mensagem de resultado(ok).

Caso seja uma mensagem de hellow, na linha 7 o nodo responde ao nodo que

Algorithm 1: Pseudo código protocolo TDMEN

```

1 while hasBattery() do
2   event  $\leftarrow$  waitSenseEvent();
3   if event then
4     sendBroadCastMessage(tag, msgData);           // send Hellow tag
5   while msg  $\leftarrow$  waitMessage() do
6     if msg.HasHellow() then
7       sendMessage(msg.nodID, tag, msgData);           // send hey tag
8     else if msg.HasHey() then
9       updateNeighborsTable(msg.nodID, msgData);
10      nodId  $\leftarrow$  checkNeighborsTable();
11      sendMessage(nodId, tag, msgData);           // send segment tag
12    else if msg.HasSegment() then
13      resultData  $\leftarrow$  procSegment(msg.data()); // get segment and do
14      SendResult(msg.nodID, tag, resultData);
15    else
16      results  $\leftarrow$  gedAndCheckResults(msg.NodID, msg.resultData);
17      sendToSink(results);           // summarize results and send to Sink
  
```

enviou a mensagem com uma tag de hey e informa os dados da sua bateria. Ainda se a mensagem recebida for de hey, na linha 9 o nodo atualiza a tabela de vizinhos com as informações recebidas, na linha 10 decide qual nodos iram processar os dados e na 11 envia os segmentos aos nodos que iram processar os dados. já se a mensagem recebida conter uma tag de processamento(segment), ele executa esta tarefa na linha 13, usando os dados enviados do segmento na mensagem, e na linha 14, retorna o resultado ao nó mestre usando o identificador do nó incluso na mensagem.

Por fim, nas linhas 16 e 17, caso seja uma mensagem de resultados(ok), o nó mestre, ao receber, sumariza esses dados com os demais já processados e, caso tenha terminado, envia o resultado ao Sink.

4.2 TDMEN-CACHE: TDMEN integrado com esquema de cache

A abordagem TDMEN, apresentada na Seção 4.1, propõe a descentralização do processamento através da divisão e distribuição da carga de um evento entre os vizinhos. Analisando ainda a carga de processamento em RSSFM reativas, observa-se redundância em processamento, que poderia ser evitado com um esquema de cache. Visto que uma fração de dados, por exemplo, com quadrantes similares ou até idênticos em uma imagem, pode ser capturada por dois eventos distintos, o que acarretaria processamento redundante (Matheen; Sundar, 2023). O TDMEN-CACHE é proposto como uma adaptação da TDMEN que permite evitar o processamento redundante,

causado pela redundância temporal entre eventos multimídia.

Considerando um evento que captura uma imagem, onde esta imagem será usada para identificar algum objeto, como, por exemplo, uma armadilha de identificação de Borboletas (Lima; Almeida leandro; Valero; Coronel; Bazzo, 2020). Nesta aplicação, a borboleta será atraída para uma armadilha com alimento, onde uma imagem será capturada por uma câmera ativada pelo movimento do inseto. Quando esta borboleta entra na armadilha, ela dispara o evento e uma foto é capturada, porém, esta mesma borboleta poderá disparar outros eventos consecutivos pelo seu movimento. Deste modo, as imagens capturadas em sequência muito provavelmente serão muito parecidas, possuindo um considerável volume de dados idênticos.

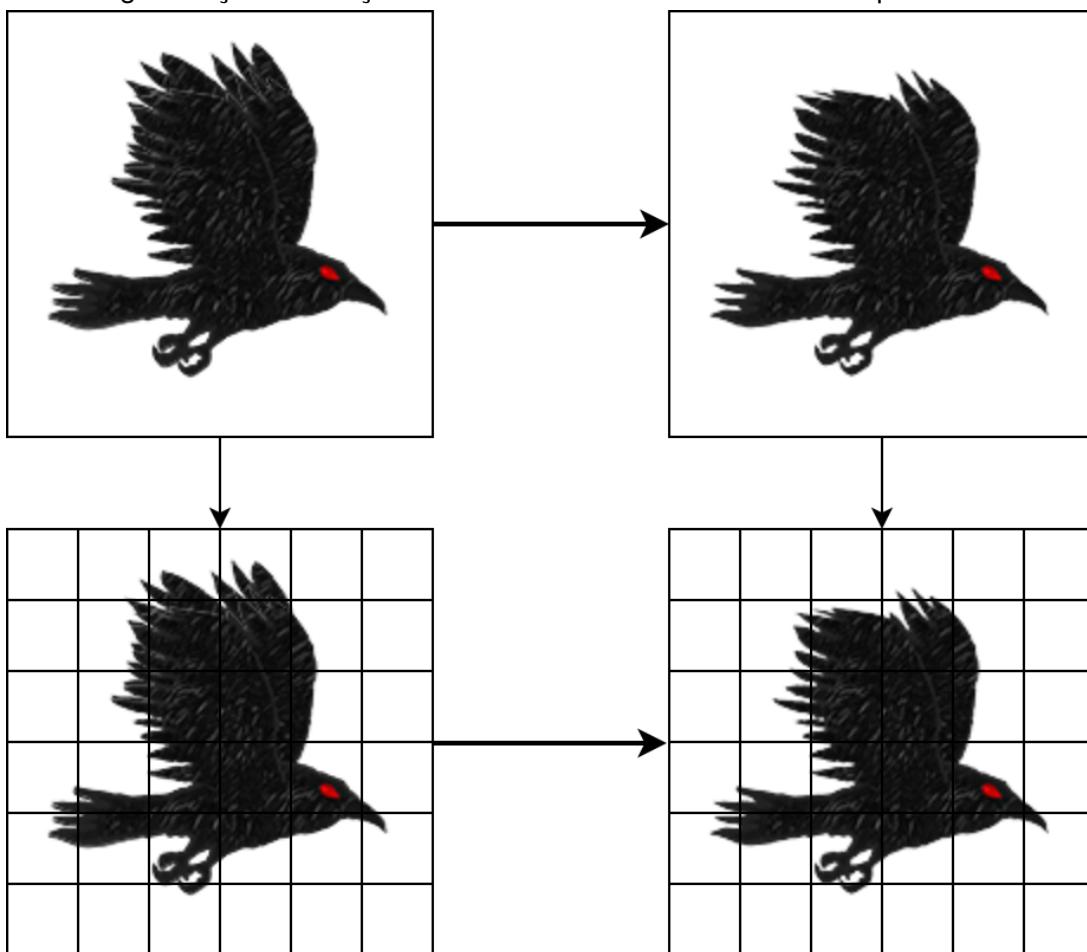
Essa redundância de dados pode ocorrer em diversas outras aplicações de redes de sensores multimídia, como, por exemplo, na identificação de pássaros por seus cantos. Nesta aplicação, o nó de uma RSSF detecta um som em uma floresta e, por meio de um algoritmo de IA, a identificação do pássaro pode ser executada. Comumente, um mesmo pássaro irá produzir o mesmo som, canto, diversas vezes em sequência (Williams, 2004), disparando assim vários eventos multimídia contendo uma mesma informação.

Este conteúdo repetido a cada evento, gera um retrabalho considerável a cada nova detecção de evento, drenando a bateria dos nodos gradativamente e por fim causando a morte prematura deles (Matheen; Sundar, 2023). Assim, de modo a contornar esse problema, a estratégia apresentada nesta seção faz uso de uma pequena memória cache em cada nodo, a qual é usada para armazenar os dados do evento anterior, usando estas informações para comparar com a ocorrência do evento atual e deste modo extrair apenas a fração dos dados que são notoriamente novos no evento e assim executar o processamento final apenas destes.

Na Figura 9, é ilustrado o processo de segmentação de uma imagem, que consiste na subdivisão da foto em quadrantes com pontos de interesse. Esses processos são frequentemente adotados para facilitar o processamento de uma imagem. A estratégia proposta aqui aproveita esse processo para identificar os novos dados a serem processados, como também é evidenciado na Figura 9. As imagens exibidas nos quadrantes superiores desta figura representam os dados de dois eventos multimídia capturados em sequência. A segmentação destas imagens é ilustrada nos quadrantes inferiores da referida figura.

Comparando a imagem da esquerda com a da direita na Figura 9, pode-se observar que uma grande parte da informação capturada pelos dois eventos é redundante, o que é comum no sensoreamento de sequências de eventos multimídia temporalmente próximos. Assim, essa redundância temporal pode ser muitas vezes identificada com o uso de uma memória cache que armazene os dados do evento multimídia anterior. Esses dados podem ser então comparados com as informações atuais, identificando

Figura 9 – Segmentação e variação entre eventos com redundância temporal.



Fonte: Elaborada pelo autor.

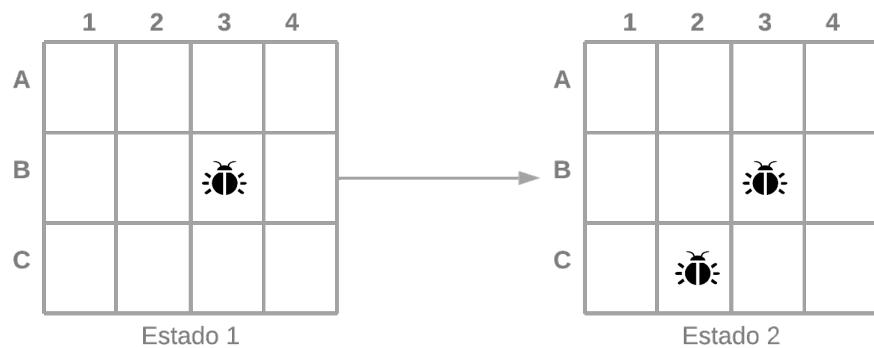
a redundância e evitando o processamento completo das informações novamente.

Quando a TDMEN-CACHE é empregada, os dados do evento anterior são salvos no nodo detector e representam o estado anterior. Este dado representa um evento já processado pela rede. A abordagem realiza a intersecção entre a imagem do evento capturado com a do evento anterior, para isolar os dados que ainda requerem processamento, evitando assim retrabalho.

Após ser realizada a identificação dos novos segmentos deste evento, os quais deverão ser processados, o fluxo de execução da divisão e distribuição de tarefas entre os vizinhos segue o mesmo do TDMEN apresentado na Seção 4.1. Na sequência é feita a identificação dos vizinhos com a montagem da tabela pelo nodo que detectou o evento e posteriormente, caso necessário, o nodo irá distribuir o trabalho entre seus vizinhos para que eles executem o processamento destes segmentos. A divisão realizada pelo TDMEN, integrado ao esquema de cache, se restringe aos dados não redundantes entre eventos consecutivos, permitindo uma maior economia de bateria dos nodos.

Para explicar a integração da cache no funcionamento da abordagem TDMEN-CACHE, será considerada uma aplicação de identificação de insetos capturados em armadilhas com pisos adesivos e câmeras disparadas por sensores de passagem apresentada em (Martins; Freitas; Aguiar; Brisolara; Ferreira, 2019) e detalhada por (Freitas; Martins; Aguiar; Brisolara; Ferreira jr., 2022). Cada inseto capturado no piso adesivo gera novos eventos, e consequentemente novas imagens do mesmo piso adesivo. A Figura 10 ilustra dois estados do piso adesivo, representando a imagem segmentada no evento anterior (estado 1) e a imagem capturada pelo evento atual (estado 2). Comparando as duas segmentações, apenas o segmento C2 teve uma mudança de estado, em relação ao cache armazenado, assim este será o único segmento para o qual será necessário executar o processamento. Neste exemplo, ao invés do processamento de 11 segmentos, seria necessário apenas o processamento do C2. Assim, o esquema de cache permite uma considerável redução do número de segmentos a serem processados, e consequentemente a redução do consumo energético. Embora seja introduzida a necessidade da execução de uma nova operação, a de intersecção entre os segmentos da imagem, esta ainda é consideravelmente menos custosa que a de executar novamente o processamento de todos os segmentos.

Figura 10 – Cache de imagem.



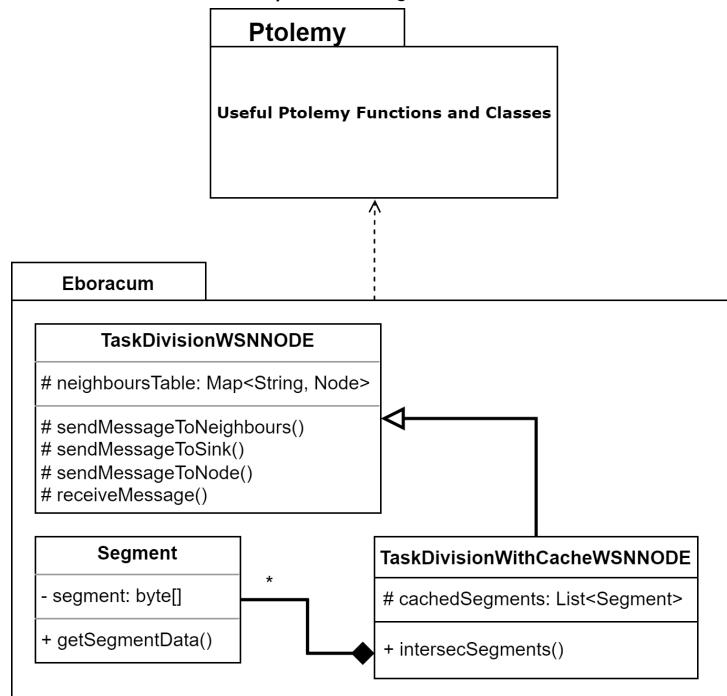
Fonte: Elaborada pelo autor.

Para implementação e avaliação desta abordagem também tomou-se como base o *Framework Eboracum*, o mesmo já empregado para o desenvolvimento da abordagem TDMEN original. Os nodos habilitados a dividir tarefas seguindo a abordagem TDMEN eram representados como instâncias de *TaskDivisionWSNNode*. Esta classe foi então estendida para permitir a integração do esquema de cache, gerando a classe *TaskDivisionWithCacheWSNNode*. Os nodos representados por esta classe possuem uma área para armazenamento que representa os segmentos da imagem, representados no framework pela classe *Segment*. Estas classes e suas relações estão ilustradas no

diagrama da Figura 11.

A classe *TaskDivisionWithCacheWSNNode* foi desenvolvida, estendendo a classe *TaskDivisionWSNNode*, com a inclusão de toda a lógica do esquema de cache proposto. Assim, a *TaskDivisionWithCacheWSNNode* permite o armazenamento dos segmentos do evento e a intersecção do evento anterior com o novo evento, de modo a extrair apenas os segmentos novos a serem processados. Enquanto a classe *Segment* representa o armazenamento dos bytes de um segmento, a qual representa os dados de um evento multimídia.

Figura 11 – Diagrama de classes da implementação da TDMEN-CACHE.



Fonte: Elaborada pelo autor.

No diagrama ilustrado na Figura 11 os principais métodos e atributos que compõem as classes empregadas na implementação da abordagem TDEMN-CACHE podem ser observados. O método *IntersecSegments* é o principal responsável por executar a comparação entre os dados do evento multimídia atual com o evento disponível em cache, buscando assim identificar a recorrência do mesmo.

O Algoritmo 2 apresenta um pseudocódigo da estratégia TDMEN-CACHE proposta por este estudo. Este código simplificado mostra a sequência básica executada por cada um dos nós da rede. Na linha 1, o algoritmo verifica se o nó ainda possui bateria para executar tarefas; já na linha 2, o fluxo de execução aguarda a detecção de um evento.

Algorithm 2: Pseudo código protocolo TDMEN-Cache

```

1 while hasBattery() do
2   event  $\leftarrow$  waitSenseEvent();
3   if event then
4     segmentsToProc  $\leftarrow$  checkCachedSegments(event.currentSegments());
5     if segmentsToProc  $>$  0 then
6        $\downarrow$  sendBroadCastMessage(tag, msgData);           // send Hellow tag
7     else
8        $\downarrow$  sendToSink(getCached(event));           // send results to Sink
9
10   while msg  $\leftarrow$  waitMessage() do
11     if msg.HasHellow() then
12        $\downarrow$  sendMessage(msg.nodID, tag, msgData);           // send hey tag
13     else if msg.HasHey() then
14        $\downarrow$  updateNeighborsTable(msg.nodID, msgData);
15       nodId  $\leftarrow$  checkNeighborsTable();
16        $\downarrow$  sendMessage(nodId, tag, msgData);           // send segment tag
17     else if msg.HasSegment() then
18        $\downarrow$  resultData  $\leftarrow$  procSegment(msg.data());     // get segment and do
19        $\downarrow$  SendResult(msg.nodID, tag, resultData);
20     else
21        $\downarrow$  results  $\leftarrow$  gedAndCheckResults(msg.NodID, msg.resultData);
            $\downarrow$  sendToSink(results);           // summarize results and send to Sink
  
```

Na linha 3, o algoritmo verifica se um evento foi detectado. Caso afirmativo, o nó que detectou o evento, na linha 4, verifica se os dados desse evento já estão inclusos em seu cache, identificando se o evento atual é uma recorrência e assim, evitando o processamento duplicado das mesmas informações. Ainda na linha 4, com o parâmetro e retorno do método que verifica o cache, é feita a atualização do cache e dos segmentos a serem processados respectivamente. Na linha 5 o nodo verifica se existe algum segmento a ser processado, caso afirmativo este envia uma mensagem de broad cast com a tag de hellow. Ainda, na Linha 8 caso não haja nenhum, segmento a ser processado, indicando que os dados já estão em cache, o nodo envia os resultados conhecidos ao Sink.

Na linha 9 o algoritmo verifica se recebeu alguma mensagem de seus vizinhos. Se tiver recebido, ele identifica, na mensagem usando a tag, se é uma mensagem para processar um segmento, uma mensagem de notificação(hey, hellow) ou ainda se é uma mensagem de resultado(ok).

Caso seja uma mensagem de hellow, na linha 11 o nodo responde ao nodo que enviou a mensagem com uma tag de hey e informa os dados da sua bateria. Ainda se a mensagem recebida for de hey, na linha 13 o nodo atualiza a tabela de vizinhos com as informações recebidas, na linha 14 decide qual nodos iram processar os dados e na 15 envia os segmentos aos nodos que iram processar os dados. já se a mensagem recebida conter uma tag de processamento(segment), ele executa esta tarefa na linha 17, usando os dados enviados do segmento na mensagem, e na linha 18, retorna o resultado ao nó mestre usando o identificador do nó incluso na mensagem.

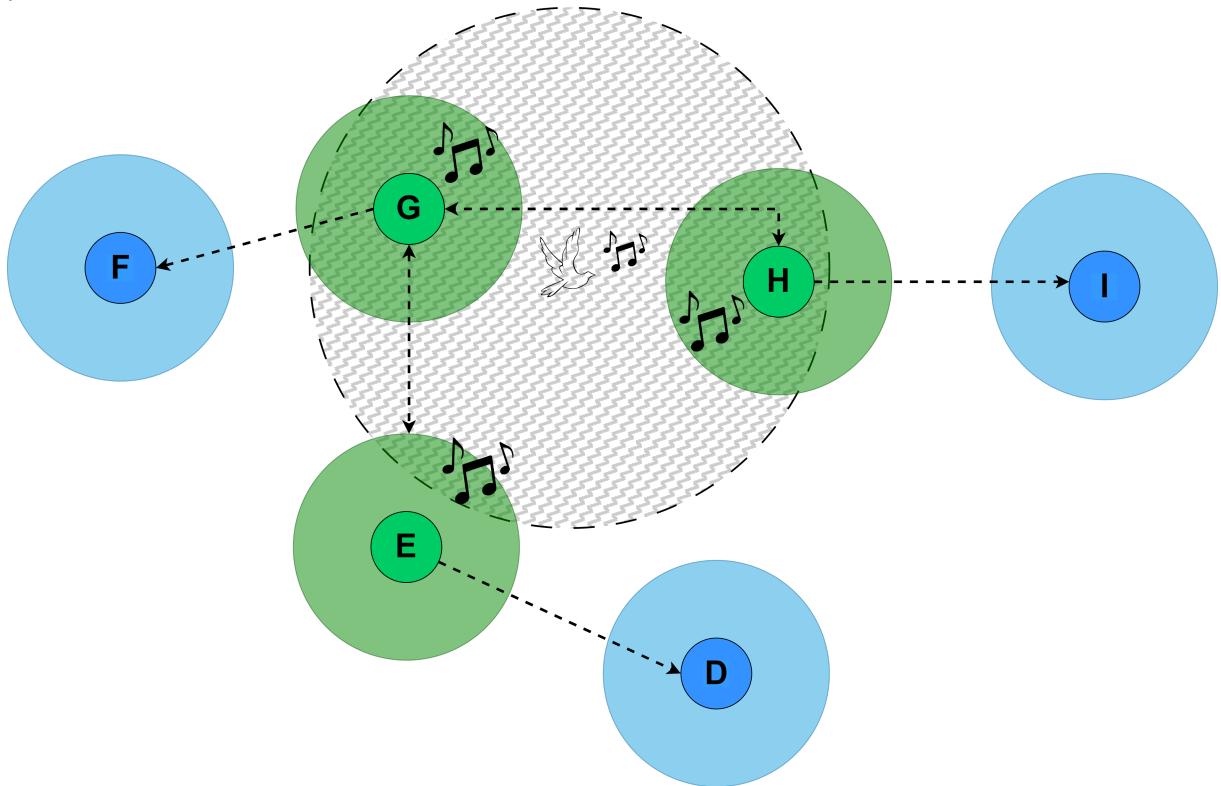
Por fim, nas linhas 20 e 21, caso seja uma mensagem de resultados(ok), o nó mestre, ao receber, sumariza esses dados com os demais já processados e, caso tenha terminado, envia o resultado ao Sink.

4.3 TDMEN-INTERSEC: Balanceamento de Carga em Redes com Sobreposição de Sensores

A abordagem TDMEN original propõe a divisão do processamento referente a um evento e a distribuição destas subtarefas aos nodos vizinhos. No entanto, em RSSF onde um mesmo evento puder ser detectado por mais de um nodo, ou seja, com sobreposição nas áreas de sensoriamento dos nodos, a TDMEN original não apresentará vantagens significativas por que cada nodo que detectar o mesmo evento, fará a distribuição entre os vizinhos, podendo inclusive gerar uma sobrecarga maior do que a RSSF sem balanceamento, devido à sobrecarga no envio de informações redundantes do mesmo evento. Para RSSF com esta redundância espacial, a abordagem TDMEN-INTERSEC é proposta, a qual explora esta redundância para que os nós que capturam o mesmo evento, cooperem no processamento dos dados multimídia. Além

de evitar o reprocessamento de dados, a abordagem se favorece da redundância espacial, permitindo reduzir as comunicações necessárias para esta divisão de tarefas entre alguns dos nodos e ainda ampliar as possibilidades de distribuição com vizinhos a um hop de distância.

Figura 12 – Exemplo de RSSFM com redundância espacial e as comunicações empregadas pela TDMEN-INTERSEC.



Fonte: Elaborada pelo autor.

Para detalhar esta abordagem será usada uma RSSFM para detecção e identificação de sons de pássaros. A Figura 12 ilustra um cenário desta rede, onde três nodos (H, G e E) detectam simultaneamente o mesmo evento. Em RSSF onde há uma redundância temporal, isto é, onde dois ou mais nodos podem detectar o mesmo evento, a propagação da detecção deste evento pode ser visualizada como uma forma de onda, onde na ilustração os nodos H, G e E detectam o evento, o canto do pássaro, sequencialmente.

Como todos os nodos H, G e E possuem as informações do evento em sua memória, podem trabalhar de forma colaborativa para processar o evento, dispensando a necessidade do envio de segmentos entre eles. A comunicação entre eles fica resumida a verificação se o evento detectado é o mesmo. Além disso, dado que os três nodos que sensorearam o evento (H, G e E) possuem vizinhos distintos, a abordagem amplia as possibilidades de distribuição de subtarefas, pois permite atribuir subtarefas a cada vizinho imediato destes nodos. Na Figura 12, os nodos I, F e D, os quais

não detectaram o evento, também poderão receber parte da carga de processamento referente ao evento corrente.

Para que esta abordagem pudesse ser implementada, um protocolo de comunicação entre os nodos foi concebido, o qual define as trocas de mensagens realizadas pelos nodos da rede quando a TDMEN-INTERSEC é adotada. A Figura 13 ilustra este protocolo, no exemplo apresentado os nós que detectam um evento (G e H) enviam uma mensagem de broadcast 'Hello' para todos os seus vizinhos. Esta mensagem contém um identificador único que representa o evento e a informação atual da bateria do nó remetente.

Quando os nodos recebem uma mensagem 'Hello', eles verificam, por meio do identificador, se o mesmo evento foi recentemente sensoreado por eles. Caso a resposta seja negativa (F e I), eles enviam uma mensagem 'Hey' de volta ao nó remetente, incluindo informações sobre sua própria bateria, sinalizando assim sua disponibilidade para colaborar no processamento do evento.

Se o nó já tiver detectado o evento (no exemplo, G e H), o nó com a maior quantidade de bateria assume a posição de Mestre (G). O nó mestre é responsável por tomar decisões sobre quais nós processarão quais segmentos do evento. O outro nó, chamado de Escravo (H), envia uma tabela de seus vizinhos imediatos ao nodo Mestre, esta tabela contém nomes e quantidades de bateria dos nós vizinhos.

O nó mestre (G no esquema ilustrado), ao receber as respostas de seus vizinhos, toma decisões com base em seus níveis de bateria para determinar quais nós participarão do processamento do evento. No cenário ilustrado, o nó Mestre (G) decide enviar uma parte dos dados multimídia do evento (segmento) ao nodo F, seu vizinho imediato, para processamento. Ainda, o nodo mestre (G) comunica ao nodo escravo (H) por meio de uma mensagem 'Send' que este deve encaminhar uma parte do evento ao seu vizinho, o nó I. Como o nó H também detectou o mesmo evento, não é necessário enviar o segmento em si, sendo necessário apenas a identificação do segmento a ser processado, uma vez que o nodo H já possui os dados referentes ao evento corrente. Esta estratégia reduz o custo energético desta comunicação, sendo este apenas o custo de envio de uma mensagem normal, isto é, o custo de envio de uma mensagem sem o payload de dados multimídia.

O nó F recebe a mensagem do nodo G, processa e envia de volta o resultado, juntamente com sua nova quantidade de bateria. Esta quantidade de bateria servirá para que o nodo mestre atualize sua tabela de vizinhos e possa usar como referência caso haja necessidade da distribuição de novos segmentos do evento entre seus vizinhos.

Enquanto isso, o nó escravo (H), ao receber a mensagem do nodo mestre (G), encaminha uma mensagem do tipo 'Segment' ao nodo I, como ilustrado na Figura 13. Esta mensagem contém os dados do segmento 1, habilitando o nodo I a processá-lo. O nó I, ao receber o segmento de H, o processa e envia o resultado de volta para o

nó H, incluindo sua nova quantidade de bateria. Por fim, o nó escravo (H) retorna o resultado obtido do processamento realizado pelo nó I para o nó mestre (G).

No exemplo, o nó I não é um vizinho imediato do nó G e, portanto, se G fosse repassar os dados do segmento para o nó I, teríamos custos de retransmissão associados (distância de 2 *hops*). No entanto, por meio desta estratégia, os dados dos segmentos multimídia, que incluem custos maiores de transmissão, sempre trafegam apenas um único *hop* na rede, e mesmo assim conseguimos atingir nós a uma distância consideravelmente maior. Como essa mensagem de envio de segmento pode envolver um volume grande de dados, essa estratégia tem a capacidade de promover uma redução significativa no consumo relativo a essa comunicação.

O esquema de mestre e escravo proposto na abordagem TDMEN-INTERSEC amplia os colaboradores no processamento de um evento, sem aumentar os custos com mensagens volumosas e explorando a redundância espacial. Assim, quanto mais nós detectarem o mesmo evento, maior será a lista de nós elegíveis para colaborar no processamento dos dados multimídia (ou segmentos) desse evento, ampliando os efeitos do balanceamento de carga na rede.

Esta implementação baseou-se na criação de duas novas classes: a *TaskDivisionIntersecProtocol*, *EventDataControl* e *Protocol* as quais executam a seguinte função:

- **TaskDivisionIntersecProtocol** - classe desenvolvida com a extensão da classe *TaskDivisionWithCacheWSNNode*, discutida na Seção 4.2. Esta classe representa cada um dos nodos da rede, executando o envio, recebimento e o processamento das mensagens empregadas pela TDMEN-INTERSEC. Como o fluxo de comunicação desta abordagem é ligeiramente distinto da usada pela TDMEN-CACHE, o método *receiveMessage* foi sobreescrito para se adequar ao novo protocolo;
- **EventDataControl** - classe que armazena os segmentos de um evento e os demais dados agregados a este;
- **Protocol** - representado aqui como uma interface a qual deve designar o que cada uma das tags de mensagens do fluxo de comunicação deve executar, seja esta processar um segmento ou encaminhar o mesmo a outro nodo;

A Figura 14 apresenta o diagrama de classes da abordagem TDMEN-INTERSEC, implementada com o uso das classes fornecida pelo framework eboracum. Na figura é ilustrado a relação das 3 novas principais classes desta abordagem: *TaskDivisionIntersecProtocol*, *EventDataControl* e *Protocol*.

O Algoritmo 3 apresenta um pseudocódigo da estratégia TDMEN-INTERSEC proposta por este estudo. Este código simplificado da estratégia mostra a sequência básica executada por cada um dos nós da rede. Na linha 1, o algoritmo verifica se

Algorithm 3: Pseudo código protocolo TDMEN-INTERSEC

```

1 while hasBattery() do
2   event  $\leftarrow$  waitSenseEvent();
3   if event then
4     sendBroadCastMessage(tag, msgData);           // send Hellow tag
5   while msg  $\leftarrow$  waitMessage() do
6     for all proTag proc in protocolResolver do
7       proc.resolve(msg.getTag(), msg.data());           // get tag and do
  
```

o nó ainda possui bateria para executar tarefas; já na linha 2, o fluxo de execução aguarda a detecção de um evento. Nas linhas 3 e 4 o algoritmo verifica se um evento foi detectado. Caso afirmativo, o nó que detectou o evento notifica seus vizinhos por meio de uma tag, indicando o sensor que fez a detecção, incluindo informações básicas sobre a identificação do evento na mensagem. Finalmente, nas linhas 5, 6 e 7, o nó verifica se recebeu alguma mensagem de seus vizinhos. Se tiver recebido, ele identifica, por meio da tag na mensagem, qual tarefa deve executar, seja processar um evento ou sinalizar sua disponibilidade.

4.4 Modelagem da Perda de pacotes

Visando aproximar os resultados de nossas simulações aos resultados de experimentos realizados em ambiente real, nesta seção será discutida a perda de pacotes relacionadas sobretudo a interferências comuns em ambientes reais. Com base nesta discussão, é proposta uma forma de incluir um modelo de perda nas simulações realizadas nesta tese.

A modelagem de perdas concebida aqui utilizou como referência os estudos de (Yin; Huang; Chen, 2016) e (Fakhri; Gharghan; Mohammed, 2018), que conduziram testes com módulo S2C e o protocolo ZigBee. Esses estudos permitiram identificar e modelar equações que descrevem a perda de pacotes e a intensidade do sinal com base na distância entre os nodos, tanto em ambientes externos quanto internos.

$$p = 1.279 * 10^{-11} * e^{-0.2641x} \quad (1)$$

A Equação 1 modela o percentual de perda de pacotes em uma rede wireless com o emprego do protocolo ZigBee. O parâmetro x corresponde ao RSSI do nó receptor dos dados da rede, e o valor de p expressa o percentual de perdas, conforme apresentado em (Yin; Huang; Chen, 2016). Com base na equação e nos demais dados e resultados apresentados em (Yin; Huang; Chen, 2016), pode-se identificar que quando o valor do RSSI é igual ou inferior a 75, o percentual de perda de pacotes

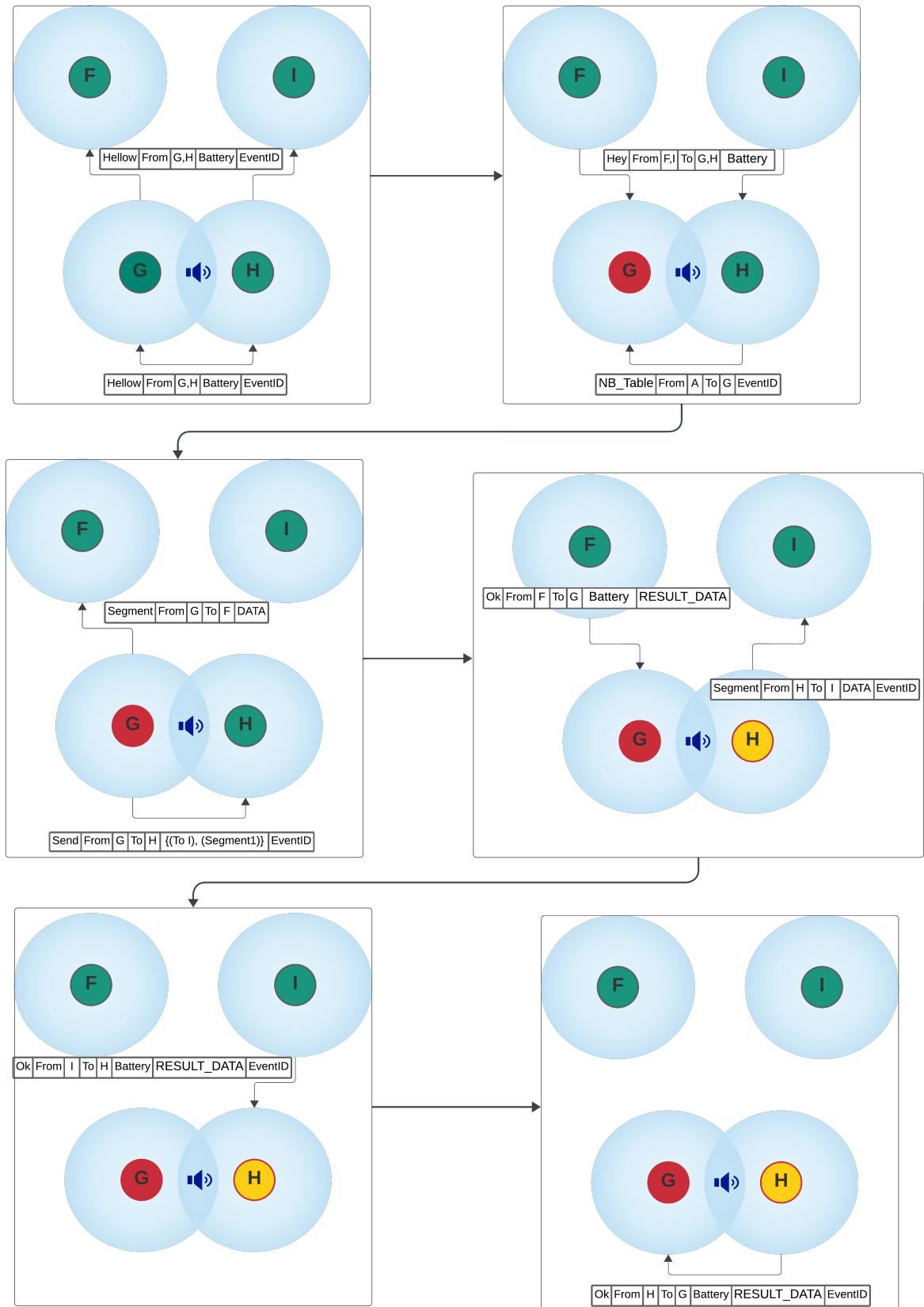
não ultrapassa 0.5%. Entretanto, os autores ainda apontam que com valores de RSSI acima de 75, o percentual de perda de pacotes aumenta exponencialmente. As perdas permanecem abaixo de 1% até um RSSI de 77, porém, esta porcentagem sobe para aproximadamente 7% quando o RSSI atinge 85.

$$RSSI = -17.435\log(d) - 35.149 \quad (2)$$

A Equação 2, apresentada no estudo (Fakhri; Gharghan; Mohammed, 2018), modela uma estimativa de sinal RSSI no S2C Zigbee em ambientes externos. Nesta equação, o valor da variável d corresponde à distância em metros entre um nodo e outro. Com base nesta equação, considerando a distância entre os nodos e fazendo relação com o estudo de (Yin; Huang; Chen, 2016) que compara o valor do RSSI com a perda de pacotes, pode-se modelar a perda de pacotes aproximada em nossas simulações. Por exemplo, em um cenário de RSSFM para manejo de pragas onde os nodos estão distantes em 150m, pode ser considerado um RSSI de 75, resultando em uma perda média de pacotes de aproximadamente 0.5%.

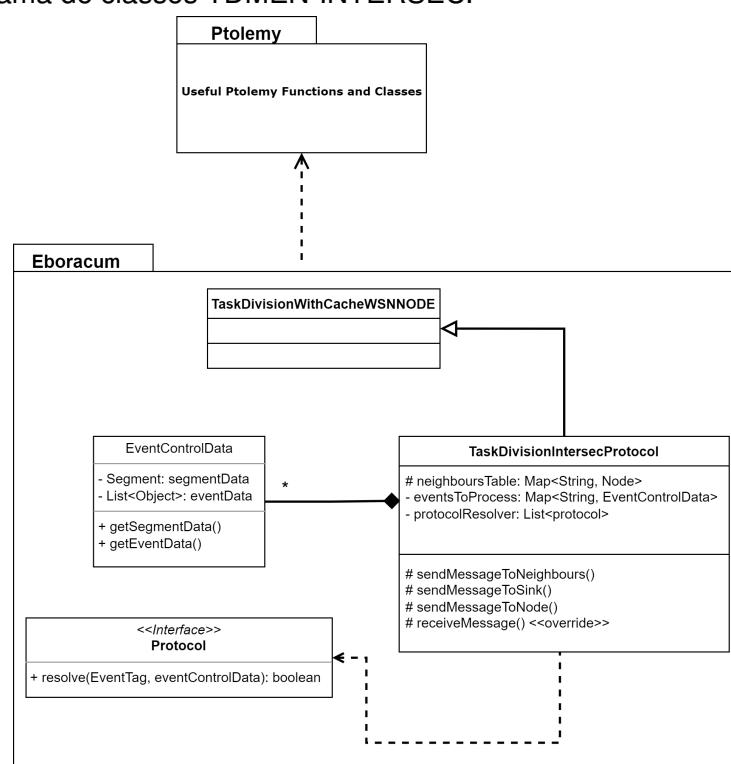
Apesar das simulações e testes de intensidade de sinal apresentados em (Fakhri; Gharghan; Mohammed, 2018) cobrirem uma ampla gama de ambientes externos, ainda podem ocorrer fatores adversos e desconhecidos que impactem na qualidade do sinal. Contudo, como identificamos com base no trabalho de (Yin; Huang; Chen, 2016) e com suporte da equação 1, a perda de pacotes está diretamente relacionada à intensidade do RSSI recebida. Deste modo, caso a intensidade do sinal entre os nós da rede varie muito da média esperada de 75, ou ainda caso haja alguma necessidade, seria possível ajustar o espaçamento dos nodos da rede de modo que o RSSI se adeque à perda de pacotes correspondente desejada, tomando como base a equação 1.

Figura 13 – Fluxo de comunicação TDMEN-INTERSEC.



Fonte: Elaborada pelo autor.

Figura 14 – Diagrama de classes TDMEN-INTERSEC.



Fonte: Elaborada pelo autor.

5 ESTUDO DE CASO 1: DIVISÃO DE TAREFAS EM RSSFM SEM REDUNDÂNCIA DE SENSOREAMENTO

Neste capítulo, as abordagens propostas TDMEN e TDMEN-Cache serão avaliadas no contexto de uma RSSFM para manejo de pragas em pomares. Esta rede dedicada não possui sobreposição de áreas de sensoriamento, sendo um cenário de aplicação de ambas abordagens citadas. Na Seção 5.1 esta aplicação será apresentada, bem como serão detalhados aspectos dos nós empregados nesta rede. A modelagem empregada em nossas simulações será detalhada na Seção 5.2. Nas seções 5.3 e 5.4 serão apresentados os resultados experimentais obtidos com as duas abordagens propostas por este estudo, a TDMEN e a TDMEN-CACHE, ambas voltadas para redes de sensores sem fio sem sobreposição de sensoreamento. Por fim, na Seção 5.5 são apresentados e discutidos os resultados obtidos com a inclusão da simulação de perda de pacotes aos testes executados.

5.1 RSSF para o manejo de pragas em pomares

Uma área bastante promissora para o emprego de RSSF é a agricultura de precisão, onde nodos sensores são integrados aos pomares e lavouras visando coletar e monitorar informações das culturas. Dentre os dados comumente sensoriados por estes dispositivos neste meio estão: os aspectos do solo como umidade e acidez; do clima como a temperatura e o índice da radiação solar; dá saúde das plantas como o controle e monitoramento de pragas que atacam estes cultivos (Velusamy; Rajendran; Mahendran; Naseer; Shafiq; Choi, 2021).

No contexto da agricultura, o manejo e combate das doenças que acometem as lavouras tem gerado uma crescente e constante demanda de inúmeras pesquisas e soluções mais tecnológicas e menos nocivas ao meio ambiente (Velusamy; Rajendran; Mahendran; Naseer; Shafiq; Choi, 2021). Neste meio, o uso de armadilhas eletrônicas para o controle e monitoramento de pragas nas lavouras tem se mostrado bastante relevante e promissor (Preti; Verheggen; Angeli, 2021).

Nesta tese, experimentos foram realizados usando a aplicação de manejo de pra-

gas com armadilhas eletrônicas inteligentes. Estas armadilhas são capazes de capturar e identificar insetos de interesse em um pomar, permitindo deste modo monitorar e detectar possíveis infestações. As armadilhas usam feromônios para atrair insetos e capturá-los usando um piso adesivo (Martins; Freitas; Aguiar; Brisolara; Ferreira, 2019). Um sensor de passagem (infravermelho) detecta a passagem de um inseto e ativa a captura de uma imagem (Martins; Freitas; Aguiar; Brisolara; Ferreira, 2019). Após, a imagem será processada e uma técnica de aprendizado de máquina será empregada para identificação e contagem dos insetos.

A abordagem adotada para a identificação de insetos empregada nas armadilhas foi anteriormente proposta em Martins; Freitas; Aguiar; Brisolara; Ferreira (2019). Esta abordagem usa uma técnica de Processamento Digital de Imagens para determinar regiões de interesse (ROI) na imagem, que são prováveis insetos e uma Rede Neural Artificial para analisar e classificá-los na classe certa de moscas. Esta identificação é feita essencialmente por uma Rede Neural Convolucional, como detalhado por Freitas; Martins; Aguiar; Brisolara; Ferreira jr. (2022).

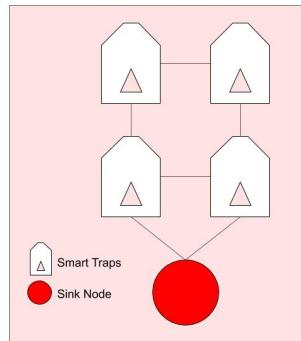
As armadilhas são conectadas por módulos de radiofrequência (RF), assim quando um nodo sensor detecta insetos, e faz a identificação e contagem dos mesmos, esta contagem é então transmitida usando conexões *multihop* até à estação base. A Figura 15 ilustra uma pequena RSSFM composta por quatro armadilhas inteligentes (*smart traps*) conectadas a um nó de estação base (*Sink node*) usando *multi-hop*. Devido à captura de imagens requerida, esta RSSF é classificada como multimídia. Na forma de captura realizada, classifica-se esta RSSF como reativa, pois novas imagens só serão capturadas se o sensor infravermelho detectar a passagem de um inseto.

Nesta tese, a RSSFM simulada baseia-se na prova de conceito da armadilha inteligente apresentada anteriormente por Martins; Freitas; Aguiar; Brisolara; Ferreira (2019) e detalhada em Freitas; Martins; Aguiar; Brisolara; Ferreira jr. (2022). A Figura 16 ilustra os principais módulos que são utilizados no protótipo de armadilha inteligente, os quais são: CPU, Módulo RF, Câmera, sensor infravermelho e uma bateria para alimentar todo o sistema.

As armadilhas inteligentes contêm um sensor infravermelho usado para ativar o nó quando um inseto entra na armadilha. Este módulo é usado para dar um comportamento reativo para nossos nós sensores sem fio, ativando a câmera para capturar uma nova imagem. A capacidade de sensoriamento para o nó da RSSF é representada pela composição de sensor infravermelho e câmera. No entanto, o raio de sensoriamento é dado pelo feromônio usado para atrair os insetos. A CPU é um Raspberry Pi III sendo usada para executar as técnicas de processamento de imagem e aprendizado de máquina propostas por Freitas; Martins; Aguiar; Brisolara; Ferreira jr. (2022).

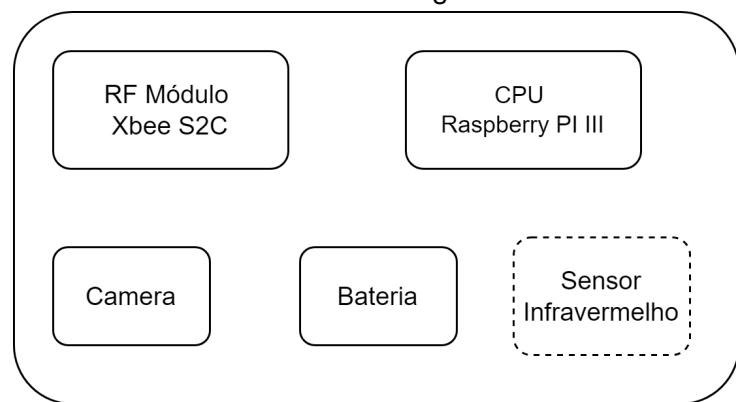
Para comunicação, as armadilhas são equipadas com módulos de RF modelo XBee S2C (Inc., 2020). O raio de comunicação empregado baseou-se em experi-

Figura 15 – Armadilhas Eletrônicas compondo uma RSSF para manejo de pragas.



Fonte: Elaborada pelo autor.

Figura 16 – Diagrama de blocos da Armadilha inteligente



Fonte: Elaborada pelo autor.

mentos em campo realizados em nosso grupo de pesquisa que indicaram um alcance máximo de 160m em região de pomares (Freitas; Martins; Ferreira júnior; Brisolara, 2018). O protocolo empregado é o Zigbee: uma camada de rede e aplicação sobre as camadas MAC e PHY especificadas no padrão IEEE 802.15.4. O Zigbee é preferido em aplicações de RSSF no domínio agrícola devido ao seu baixo custo, confiabilidade, baixo consumo de energia, auto-organização, autocorreção, topologia de malha e características de segurança (Sahitya; Balaji; Naidu; Abinaya, 2017).

Como uma restrição importante em RSSF é o tempo de vida, para nosso estudo de caso, fora caracterizado o consumo de energia para os módulos da armadilha. A Tabela 2 detalha a corrente requerida por cada módulo, usada para modelar custos energéticos e simular a descarga da bateria dos nodos da rede. Esses valores são usados nos experimentos para determinar a caracterização dos modelos de simulação empregados.

Anteriormente, um estudo de caso empregando também esta aplicação alvo foi conduzido pelo presente grupo de pesquisa, porém empregando uma RSSF proativa. Este estudo anterior, publicado em (Zanini; Souza; Brisolara; Ferreira jr, 2021), auxiliou

Tabela 2 – Corrente gasta pelos módulos da Armadilhas

Modo	Corrente (mA)
RX/Idle corrente	28
Tx corrente	33
Raspberry	800
Infra-red sensor	10

Fonte: Elaborada pelo autor.

nas definições empregadas aqui para raio de sensoriamento, rádio de comunicação, densidade da rede e algumas das caracterizações de consumo.

5.2 Modelagem do Estudo de Caso

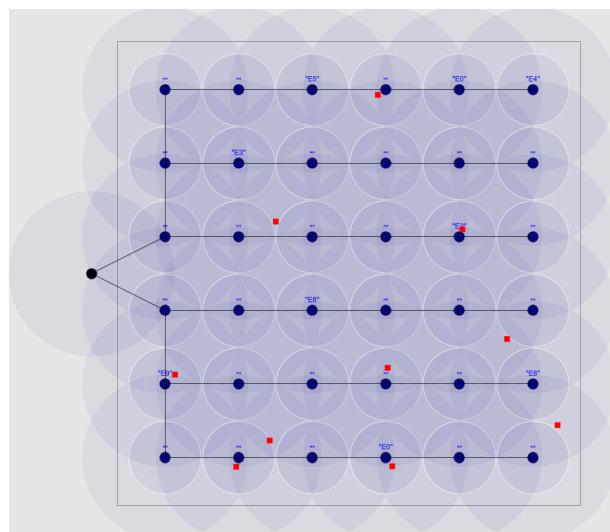
No modelo de simulação base construído no *framework* Eboracum para este estudo de caso, cada nó sensor representa uma armadilha eletrônica. Cada nodo possui um raio de sensoriamento e um raio de comunicação. Neste cenário, a cobertura do sensoriamento adotada é de 70 m, considerando o uso do feromônio *Bio Trime-dlure*. O cenário emprega-se uma armadilha por hectare, que representa um espaçamento entre armadilhas de 100m. Um evento no simulador representa um inseto detectado pelo sensor infravermelho. Assim, a cobertura de cada nó sensor nunca se sobrepõe, pois o mesmo inseto não pode estar em uma armadilha diferente simultaneamente. Nas simulações realizadas, esses nós são modelados por instâncias de classes do nosso *framework*. A classe *TaskDivisionWSNNode*, *TaskDivisionWithCacheWSNNode* e *SimpleWSNNode* são adotadas nos experimentos com TDMEN, com TDMEN-Cache, e sem nenhuma técnica de平衡amento de carga, respectivamente.

Neste estudo considerou-se um cenário com uma rede Ad Hoc em uma simulação composta por 36 nós sensores distribuídos uniformemente em uma área quadrada de 810.000 m² (900 m x 900 m) com um nodo Sink posicionado na lateral, conforme ilustrado na Figura 17. Os nodos são conectados a um de seus quatro vizinhos (o mais próximo ao Sink) ou diretamente ao Sink. O raio de comunicação é de 160 m, permitindo sempre que as mensagens dos nodos alcancem apenas os quatro nós próximos, que chamamos de vizinhos imediatos.

A Figura 17 representa uma captura da tela do Eboracum durante uma simulação, onde os nós sensores podem ser identificados (pequeno círculo azul-escuro), o raio de sensoriamento (círculo azul-claro intermediário) e o raio de comunicação (círculo maior externo). Também é possível observar a posição do Sink utilizado nas simulações, que está localizado à esquerda, fora do retângulo que representa a área de interesse. Em um cenário de agricultura de precisão, é comum que o Sink esteja fora

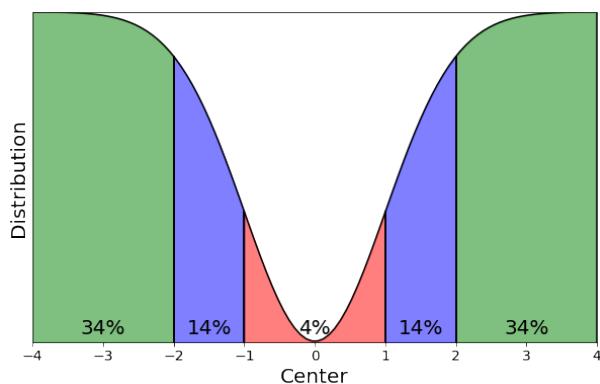
do pomar, pois este deverá estar em local com fonte de energia ilimitada e conectividade com a internet disponível. Os pontos vermelhos representam os eventos, ou seja, a captura de insetos pelas armadilhas.

Figura 17 – Cenário de simulação para manejo de pragas, pontos vermelho representam eventos e círculos azul, nodos sensores.



Fonte: Elaborada pelo autor.

Figura 18 – Distribuição de probabilidade espacial Normal Invertida.



Fonte: Elaborada pelo autor.

Usando a classe *StochasticDailyJumperEvent* fornecida pelo Eboracum, foi modelada a ocorrência de insetos em pomares. Como a presença desse tipo de inseto em um pomar é consideravelmente maior ao final do dia (Drew; Yuval, 1999), as simulações realizadas geram eventos uma vez ao dia. A frequência de eventos é modelada por uma distribuição de Poisson com intervalo médio de 68399 segundos (19 h), desvio padrão de 199 s (2 h) e intervalo de 0 a 86399 segundos (24 h). Por dia, o modelo de simulação gera dez eventos como carga de trabalho diária.

Para modelar aspectos espaciais da chegada dos insetos ao pomar, definimos um modelo de dispersão dos eventos baseado em uma distribuição Normal Invertida, onde quanto mais próximo da borda da rede, maior a probabilidade de ocorrência dos eventos. A rede de sensores está localizada em um pomar e cada evento é acionado pela presença de um inseto. Como esses insetos vieram de fora do pomar, há maior probabilidade de que armadilhas nas bordas da área de interesse capturem mais insetos (Clemente-orta; Madeira; Batuecas; Sossai; Juárez-escario; Albajes, 2020).

A Figura 18 apresenta o gráfico de dispersão de probabilidade utilizado neste estudo de caso, a probabilidade apresentada no centro do gráfico é equivalente à probabilidade de um evento ocorrer no centro da rede, o mesmo vale para as arestas do grafo e da rede. Os resultados deste estudo de caso são apresentados na Seção 5.3.

O método adotado nas armadilhas de acordo com (Freitas; Martins; Aguiar; Brislara; Ferreira jr., 2022) tem o custo de processamento das armadilhas dependente do número de regiões de interesse (NROI) na imagem, ou seja, o número de segmentos a serem processados. Assim, em nosso modelo, um evento representa um novo inseto detectado, e sua ocorrência acarreta um incremento no NROI da armadilha. Ao adotar nossa abordagem proposta, esse número é útil para determinar como os segmentos devem ser distribuídos entre os nós da rede.

Cada um dos nós da rede tem disponível uma bateria de 25200000 mAs. Para avaliar o consumo energético das diferentes soluções de RSSFM para manejo de pragas, foi necessário caracterizar as soluções de rede para esta aplicação alvo. Os custos de energia usados nestas caracterizações foram definidos com base no protótipo da armadilha e seus módulos detalhados na Seção 5.1, mas também considerando os protocolos propostos pelas abordagens de divisão de tarefas.

A primeira caracterização realizada considera a rede sem nenhuma abordagem de divisão de tarefas de processamento ou balanceamento de carga. Neste caso há três custos a serem empregados, referente ao modo de operação dos nodos, o de processamento da imagem, o de envio da mensagem com a contagem e do modo de inoperante (*idle*). Estes custos podem ser observados na Tabela 3.

Os custos de IDLE_COST refere-se ao consumo do nodo quando este não está processando e nem transmitindo mensagens, mas habilitado a capturar eventos e receber mensagens. O SEND_APP_MESSAGE representa o consumo referente ao envio da mensagem de contagem de insetos entre dois nodos vizinhos. Este refere-se ao consumo estimado para o módulo RF enviar mensagens de 36 Bytes, utilizando 250 Kbps como taxa de transmissão e uma corrente de 33 mA durante a transmissão. Nesta caracterização (Tabela 3), como não há divisão de tarefas, o custo de processamento é de todos os segmentos da imagem. O tempo estimado para a Raspberry processar um ROI é de 0,9 segundos usando uma corrente de 800 mA, totalizando 720 mAs. Assim, o custo total do processamento 720 mAs multiplicado pelo NROI

Tabela 3 – Custo de energia da tarefa sem balanceamento de carga.

Nome da Tarefa	Custo
SEND_APP_MESSAGE	0.038016 mAs
IDLE_COST	0.1 mAs
PROCESSING	420 + 720*NROI mAs

Fonte: Elaborada pelo autor.

e somado ao custo de pré-processamento (420 mAs). Neste caso, este pré-processamento é a segmentação realizada internamente pelo algoritmo de aprendizado de máquina. O custo deste pré-processamento foi dado pelo tempo estimado de execução (0,525 segundos) para esta tarefa na CPU consumindo uma corrente de 800 mA. Estes custos são os mesmos empregados nas simulações realizadas no trabalho de Zanini; Souza; Brisolara; Ferreira jr (2021).

Os custos de IDLE_COST e SEND_APP_MESSAGE são os mesmos nas três caracterizações empregadas em nossos experimentos. No entanto, para avaliar o consumo usando nossas abordagens de divisão de carga, foi necessário estender os modos de operação dos nodos no Eboracum. A Tabela 4 e a Tabela 5 apresentam os custos adotados para nossos experimentos usando as abordagens TDMEN e TDMEN-CACHE, respectivamente.

Com a divisão de tarefas proposta na TDMEN, foi necessário redefinir o custo do envio de um segmento da imagem ao nodo vizinho imediato. Esta operação foi nomeada SEND_SEGMENT. Agora o protocolo requer também a mensagem de descoberta e montagem da tabela de vizinhos, a SEND_DISCOVERY, que representa o custo do nodo na troca de mensagens de descoberta e/ou atualizar sua tabela de vizinhos. Os custos de comunicação utilizados para o SEND_DISCOVERY é equivalente ao custo de SEND_APP_MESSAGE já que possuem a mesma quantidade de bytes. Já para o envio do SEND_SEGMENT foi estimado considerando a quantidade de cabeçalhos usada por pacote a ser enviado mais uma mensagem de 3,87 KBytes, representando um segmento de imagem.

Por fim, a divisão de tarefas requer fragmentar o processamento, separando o processamento do segmento, do pré-processamento de toda a imagem. O PROCESS_SEGMENT representa o custo de processamento de um segmento da imagem, onde foi considerado os 720 mAs, mantendo a estimativa do custo de processamento de um ROI pela CPU da armadilha. O custo de PRÉ-PROCESSAMENTO foi estimado em 420 mAs.

Já na caracterização adotada nos experimentos com TDMEN-CACHE, summarizada na Tabela 5, duas novas operações tiveram de ser estimadas COMPARING_EQUALS e COMPARING_DIF. Estas operações são responsáveis por definir se um segmento precisará ser processado ou não, com base na análise de intersecção entre os seg-

Tabela 4 – Custo de energia de cada tarefa com TDMEN.

Nome da Tarefa	Custo
SEND_SEGMENT	5.0012 mAs
SEND_DISCOVERY	0.038016 mAs
SEND_APP_MESSAGE	0.038016 mAs
IDLE_COST	0.1 mAs
PREPROCESSING	420 mAs
PROCESSING_SEGMENT	720 mAs

Fonte: Elaborada pelo autor.

Tabela 5 – Custo de energia de cada tarefa com TDMEN-CACHE.

Nome da Tarefa	Custo
SEND_SEGMENT	5.0012 mAs
SEND_DISCOVERY	0.038016 mAs
SEND_APP_MESSAGE	0.038016 mAs
IDLE_COST	0.1 mAs
PREPROCESSING	420 mAs
PROCESSING_SEGMENT	720 mAs
COMPARING_EQUALS	5 mAs
COMPARING_DIF	5 mAs

Fonte: Elaborada pelo autor.

mentos de imagem: os segmentos a serem processados do evento atual e os segmentos salvos na cache do nodo.

O custo COMPARING_EQUALS equivale ao custo de uma intersecção entre dois segmentos que são exatamente iguais, onde serão comparados todos os pixels da imagem com o do segmento em questão, de modo a determinar que este segmento já está contido em cache e neste caso não precisará ser processado. Já o COMPARING_DIF equivale ao custo da execução da intersecção entre dois segmentos distintos, onde cada pixel dos segmentos em questão serão comparados até que uma diferença seja encontrada. Ao encontrar a diferença, a comparação é parada, determinando que este novo segmento ou não está contido na cache, ou somente não é igual ao comparado neste momento. Para simplificar o processo e visando evitar uma subestimativa de custos, foi adotado o custo de pior caso para ambas as situações, o que correspondente ao custo de comparar todos os pixels de um segmento. Assim, os custos COMPARING_EQUALS e COMPARING_DIF foi estimado em 5mAs para a caracterização adotada nos experimentos com TDMEN-CACHE (Freitas; Martins; Aguiar; Brisolara; Ferreira, 2022).

Este estudo de caso foi empregado em vários experimentos, com objetivo de avaliar as abordagens propostas. Todas as simulações foram repetidas trinta vezes, devido à variabilidade dos modelos estocásticos usados para modelar a ocorrência dos

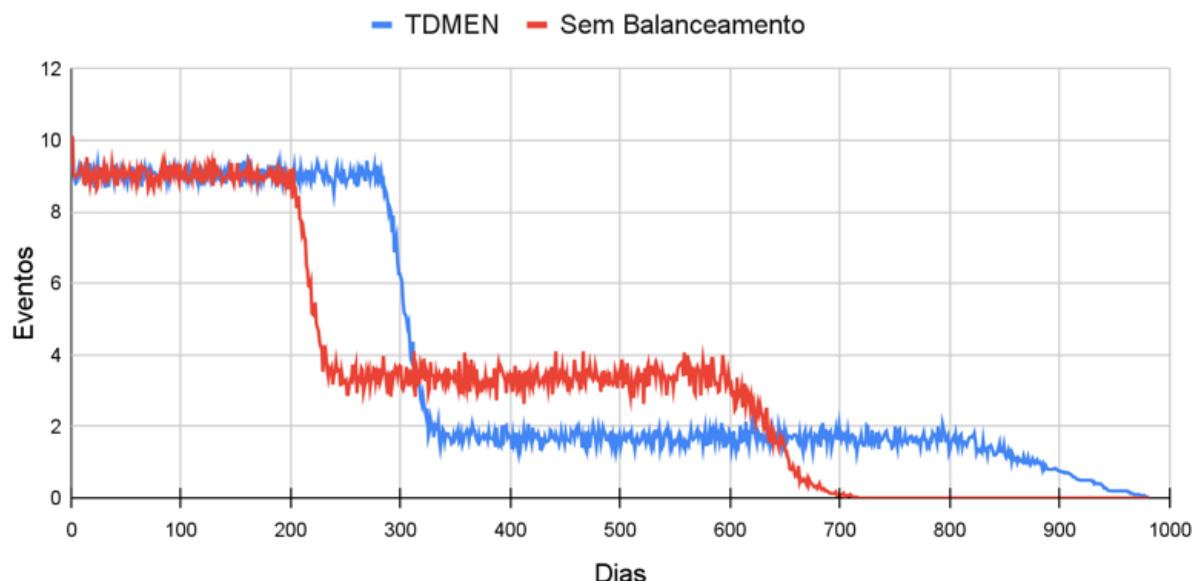
eventos. Valores médios das trinta execuções foram empregados nas comparações e o Test T de Student foi empregado para analisar se as melhorias eram significativas (Sánchez turcios, 2015), quanto a comparação de duas médias.

5.3 Resultados Comparativos TDMEN contra Rede sem Balanceamento de Carga

Os experimentos apresentados nesta seção comparam a abordagem TDMEN com uma estratégia que não usa balanceamento de carga. Na Figura 19 é possível observar a eficiência de ambas as soluções (nas 30 execuções), onde o eixo X mostra o número de eventos detectados em média por dia, e o eixo Y apresenta uma linha do tempo, ilustrando o número de dias que a rede se manteve viva.

Observando a Figura 19, é possível notar que a medida em que os nós da rede ficam sem energia e morrem, o número de eventos detectados pela rede como um todo diminui, embora a rede se mantenha viva (ao menos um nodo possui comunicação direta com o Sink). Notavelmente, a rede que não utiliza estratégias de balanceamento de carga experimenta uma queda significativa no número de eventos detectados mais cedo em comparação com a rede que emprega a nossa abordagem TDMEN.

Figura 19 – Estudo de Caso 1: Média de Eventos por dia - rede com TDMEN versus rede Sem balanceamento



Fonte: Elaborada pelo autor.

Esta linha média constante inicial de eventos detectados pelas redes que fica entre 8 e 10 eventos é o período que ambas estão com todos os nodos da sua rede ativos, ou seja, o momento em que ambas têm a capacidade total de detectar todos os eventos

Tabela 6 – Estudo de Caso 1: Resultados de Tempo de duração em dias da rede com TDMEN e Sem Balanceamento

	TDMEN	Sem Balanceamento
Rede Completa (Media - Variância)	282 - 5	200 - 9
Tempo Total (Media - Variância)	923 - 20	665 - 34

Fonte: Elaborada pelo autor.

que ocorrem em seu raio de alcance. Assim, neste gráfico, pode-se perceber que nossa estratégia TDMEN consegue manter por um período consideravelmente maior a capacidade total da rede, isto é, manter todos os nodos da rede vivos por mais tempo, mantendo a cobertura total de sensoreamento da rede e evitando consequentemente a perda prematura de eventos.

Após a primeira queda no número de eventos detectados por ambas as estratégias, ou seja, posterior a morte da primeira leva de nodos, ambas as abordagens se mantêm em um segundo patamar constante de média de processamento de eventos. No caso da estratégia sem balanceamento, a média deste segundo patamar fica entre 3 e 4 consideravelmente maior que a média da estratégia que faz uso do balanceamento de carga, este resultado se deve pelo número de nodos mortos em ambas serem diferentes. Na rede com TDMEN há um consumo uniforme de energia por quase todos os nodos da rede, causando a morte de vários nodos simultaneamente, ao contrário da estratégia sem balanceamento, na qual os nodos que ficam em locais mais propensos a detectarem eventos ou que estão em contato direto com o *Sink* morrem mais prematuramente. Consequentemente, a abordagem TDMEN prolonga o tempo total de vida da rede.

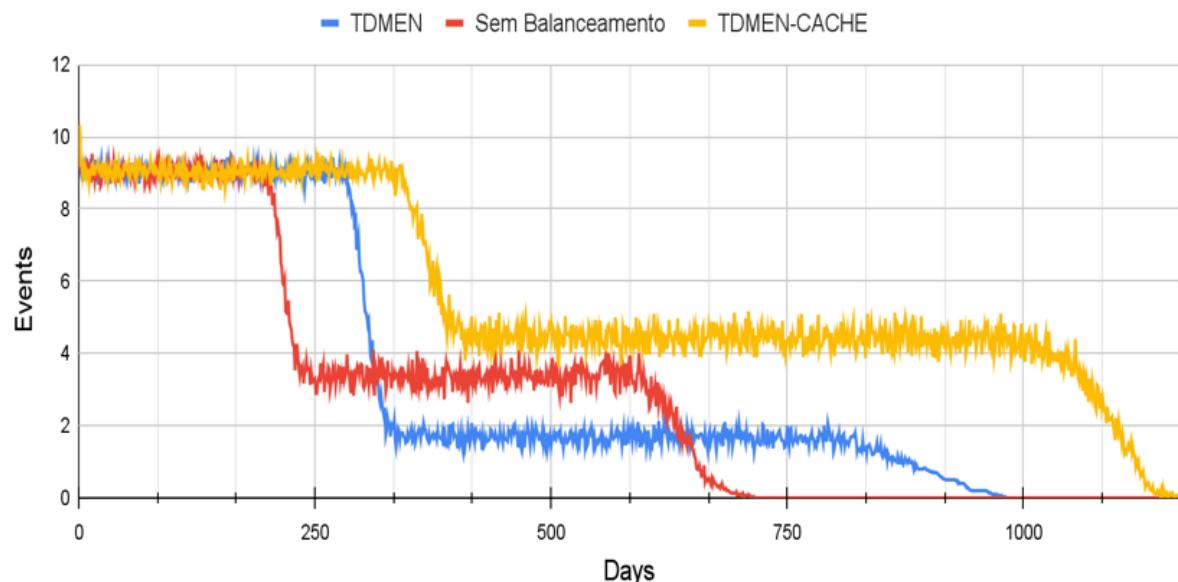
A Tabela 6 resume os resultados obtidos pelos experimentos comparativos, apresentando a média em dias de duração da rede para as trinta execuções feitas com as duas diferentes abordagens, bem como a variância encontrada. No comparativo de duração da rede completa, isto é, a duração em dias até o primeiro nodo da rede morrer, a estratégia TDMEN teve uma melhora aproximadamente de 41%. Já quando comparado com a duração total da rede, a qual é representada pela morte do último nodo que possui comunicação com o *Sink*, a melhora do TDMEN foi de aproximadamente 38%. Estes resultados foram previamente publicados em (Souza; Brisolara; Ferreira, 2022).

5.4 Resultados Comparativos TDMEN-CACHE x TDMEN x Sem Balanceamento de Carga

Esta seção analisa o impacto do emprego da TDMEN-cache no estudo de caso 1. Na Figura 20 os resultados dos experimentos realizados com as três abordagens, são

integrados em um único gráfico comparativo. Pode-se identificar neste um considerável aumento no tempo de vida total da rede quando TDMEN-CACHE foi empregado, bem como no tempo de funcionamento completo da rede, isto é, no tempo em que nenhum nodo da rede parou de funcionar por falta de bateria.

Figura 20 – Estudo de Caso 1: Média de eventos por dia obtidos com as três abordagens



Fonte: Elaborada pelo autor.

O primeiro ponto de deflexão do gráfico (Figura 20) corresponde ao início das mortes dos nodos mais requisitados da rede por falta de bateria, este ponto é retardado pela abordagem TDMEN-CACHE, prolongando o funcionamento da rede completa por mais dias. Esta melhora na eficiência da RSSFM com a TDMEN-CACHE, pode ser melhor analisada comparando a média de dias e a variância encontrada, dados sumarizados na Tabela 7. Estes resultados apontam uma ampliação dos dias de aproximadamente 21% se comparado a TDMEN, abordagem também proposta neste trabalho. Este ganho é ainda mais significativo quando comparado os resultados da RSSF com TDMEN-Cache com uma RSSF sem que haja qualquer balanceamento de carga com seus vizinhos, chegando a aproximadamente 71%.

Analizando o segundo e último ponto de deflexão do gráfico (Figura 20), onde os demais nodos restantes começam a morrer até que toda a rede fique inativa, também observa-se um ganho considerável na duração total da rede com o emprego da TDMEN-CACHE. Pelos dados sumarizados na Tabela 7, o ganho na duração total da rede alcançado pelo TDMEN-CACHE chega a aproximadamente 27% quando comparado ao TDMEN. Porém, se compararmos com a rede que executa apenas o processamento de forma local, sem nenhuma estratégia de balanceamento de carga ou divisão de tarefas, o ganho da TDMEN-CACHE sobre a rede sem balanceamento

Tabela 7 – Estudo de Caso 1: Resultados do tempo de vida da rede em dias para as três abordagens

	TDMEN-CACHE	TDMEN	Sem Balanceamento
Rede Completa (Med - Var)	341 - 3	282 - 5	200 - 9
Tempo Total (Med - Var)	1164 - 31	923 - 20	665 - 34

Fonte: Elaborada pelo autor.

chega a aproximadamente 75%.

5.5 Resultados Comparativos TDMEN-CACHE com simulação de perda de pacotes

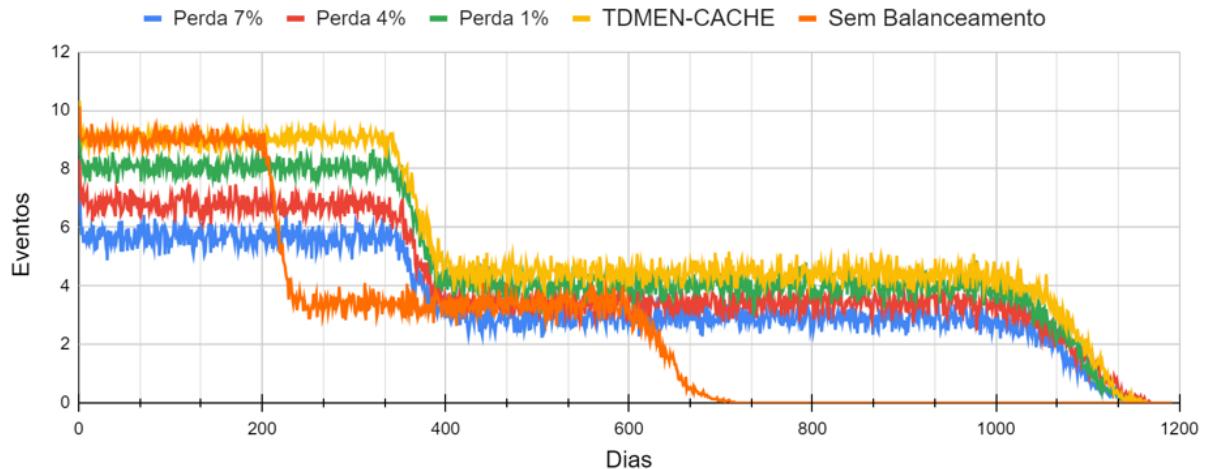
As perdas médias de pacotes esperadas para as RSSF são de aproximadamente 0.5%, conforme discutido no Capítulo 4.4 e referenciada por (Yin; Huang; Chen, 2016). Em nossos experimentos, um valor médio de 1% de perdas de pacotes foi considerado, a fim de garantir uma margem de segurança e verificar a capacidade de nossas abordagens de operar em redes com perda de dados. Além disso, foram realizados também experimentos com 4% e 7% de perdas para destacar os resultados em comparação com as estratégias apresentadas nesta tese. Esses dados também podem representar eventos climáticos extremos ou outras condições extraordinárias que podem afetar a qualidade da conexão e consequentemente o RSSI.

A RSSFM com o TDMEN-CACHE foi a que obteve os melhores resultados no estudo de caso. Visando analisar o impacto das perdas de pacotes no funcionamento da TDMEN-CACHE, esta rede foi avaliada nesta seção considerando quatro cenários distintos: TDMEN-Cache sem perdas, com perdas de 1%, com perdas de 4% e de 7%. Para fins comparativo, esta rede foi comparada a uma solução sem balanceamento de carga (e sem perdas de pacotes). A Figura 21 ilustra os resultados deste comparativo, considerando a média de eventos detectados por dia em cada cenário simulado.

Com base nos dados obtidos e apresentados na Figura 21, é perceptível que as quatro simulações distintas executadas com o TDMEN-CACHE foram consideravelmente superiores à estratégia sem balanceamento. Quando comparado, as simulações TDMEN-CACHE com e sem perdas, pode-se identificar que os resultados do tempo de vida da rede permaneceram praticamente inalterados. Sendo considerado nas simulações como a rede viva enquanto ao menos um nodo possuir comunicação com o Sink.

Ainda, mesmo na simulação do TDMEN-Cache com 7% de perdas, observou-se uma melhora de aproximados de 71% para a rede completa e 66% para a duração total quando comparada à RSSFM sem balanceamento. Esses resultados podem ser mais facilmente identificados na Tabela 8. Essa consistência nos resultados é

Figura 21 – Estudo de Caso 1: Resultados de eventos detectados por dia pela rede com TDMEN-CACHE com e sem perdas de pacotes e pela rede sem balanceamento.



Fonte: Elaborada pelo autor.

Tabela 8 – Estudo de Caso 1: Resultados do tempo de vida da rede em dias para TDMEN-CACHE com e sem perdas e para a rede sem balanceamento de carga

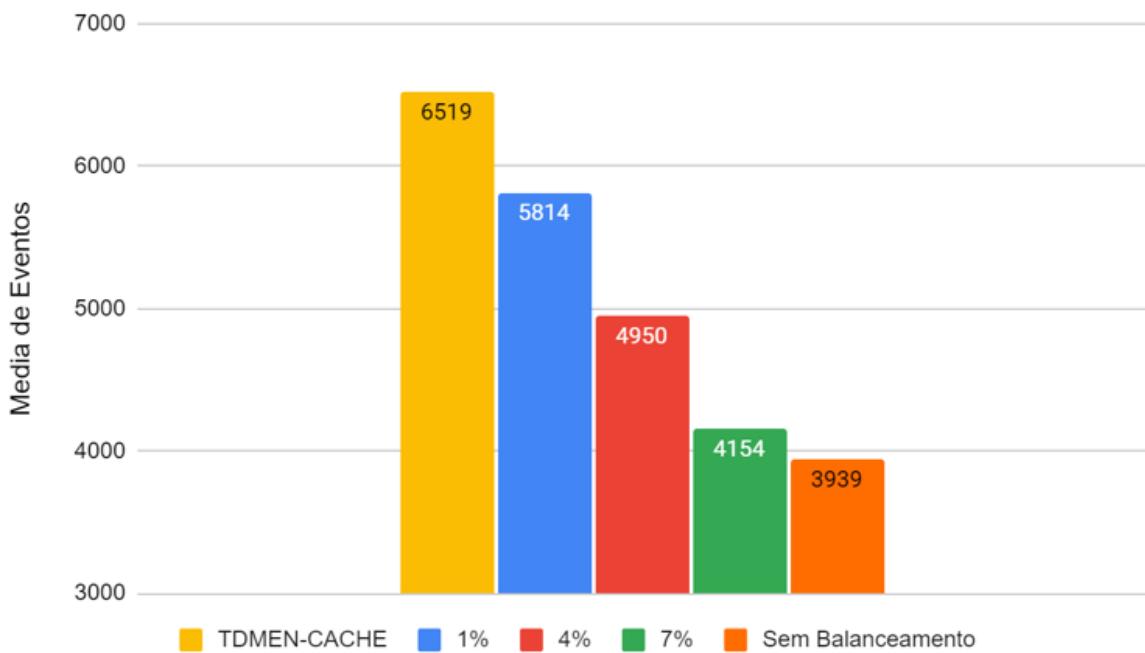
Método	Rede Completa (Med - Var)	Duração Total (Med - Var)
TDMEN-CACHE	341 - 3	1164 - 31
TDMEN-CACHE com 1%	341 - 4	1100 - 31
TDMEN-CACHE com 4%	340 - 4	1114 - 31
TDMEN-CACHE com 7%	338 - 5	1101 - 32
Sem Balanceamento	200 - 9	665 - 34

Fonte: Elaborada pelo autor.

justificada pelo comportamento da rede quando ocorre a perda de um pacote. Na simulação de perdas, o nó não realiza o reenvio do pacote perdido, resultando em uma perda parcial ou total do evento. Ainda diferentes tipos de mensagens, onde algumas, como o hello, não geram a perda do evento, dado que mais de um nodo recebe essa mesma mensagem, o que aumenta a chance de sempre existir algum nodo que receba o comando e responda ao nodo mestre, executando o processamento do dado e consumindo energia do nodo. Contudo, para que o resultado do evento seja contabilizado, ele deve chegar até o Sink via MultiHop, onde o dado pode ser perdido e impactando na contabilidade do resultado final de eventos processados.

Assim, para uma análise mais clara do impacto real no TDMEN-CACHE causado pela simulação de perda de pacotes, foram realizados testes comparativos do número total de eventos detectados pela rede. A Figura 22 ilustra os resultados obtidos com a contagem dos eventos detectados durante as simulações. É possível observar que, mesmo nas simulações em que a perda de pacotes aplicada ao TDMEN-CACHE foi de 7%, os resultados permaneceram consideravelmente superiores em comparação

Figura 22 – Estudo de Caso 1: Número de eventos detectado na rede com TDMEN-CACHE (com sem perdas) em comparativo com a rede sem balanceamento (sem perdas de pacote).



Fonte: Elaborada pelo autor.

com uma estratégia que não utiliza nenhuma técnica de平衡amento de carga.

Na Tabela 9, os resultados referentes ao número total de eventos detectados são detalhados, evidenciando um considerável aumento em sua contagem quando se utiliza o TDMEN-CACHE em todas as simulações com perda de pacotes, comparado a versão sem balanceamento de carga. Enquanto isso, na Tabela 10, é resumida a quantidade de eventos detectados a menos em comparação com uma simulação do TDMEN-CACHE sem perdas. É possível observar que, em uma simulação com uma média de perda de pacotes de 1%, a quantidade de eventos detectados a menos chega a 11%. Essa discrepância ocorre porque, como ilustrado na Figura 17, o resultado do evento, mesmo após ser processado, deve ser encaminhado até o Sink

Tabela 9 – Estudo de Caso 1: Resultados de eventos detectados pelo TDMEN-CACHE (com perdas e sem perdas de pacote) e pela rede sem balanceamento (sem perdas de pacote).

Método	Med - Var
TDMEN-CACHE	6519 - 81
1%	5814 - 78
4%	4950 - 74
7%	4154 - 75
Sem Balanceamento	3939 - 80

Fonte: Elaborada pelo autor.

Tabela 10 – Estudo de Caso 1: Porcentagem de eventos perdidos pela rede com TDMEN-CACHE, variando a porcentagem de perdas.

Perdas de Pacotes	Perdas de Eventos
TDMEN-CACHE 1%	11%
TDMEN-CACHE 4%	24%
TDMEN-CACHE 7%	36%

Fonte: Elaborada pelo autor.

Tabela 11 – Estudo de Caso 1: porcentagem de eventos detectados a mais pela rede com TDMEN-CACHE, com perdas de pacote variadas, em comparativo com a rede sem balanceamento (e sem perdas de pacotes).

Perdas de Pacotes	TDMEN-CACHE X Sem Balanceamento
TDMEN-CACHE 1%	48% mais eventos
TDMEN-CACHE 4%	26% mais eventos
TDMEN-CACHE 7%	5% mais eventos

Fonte: Elaborada pelo autor.

para ser efetivamente contabilizado. Assim, uma perda de pacote durante qualquer das retransmissões no caminho até o sink pode representar a perda de um evento, assim maiores taxas de perdas, representam uma maior perda de eventos.

Além disso, na Tabela 11, é feita uma comparação entre as abordagens do TDMEN-CACHE com perdas e a estratégia que não utiliza balanceamento, onde é calculada a porcentagem de eventos adicionais detectados por cada uma. Dessa forma, é possível identificar que, na simulação com perdas de 1%, que representa o cenário normal esperado de perda de pacotes, o número médio de eventos adicionais detectados foi de 48% em comparação com a estratégia sem balanceamento. Mesmo na simulação com perdas de 7%, houve um considerável aumento no número de eventos detectados, chegando a 5% a mais de eventos. É importante destacar que o percentual de 7% de perdas utilizado é um caso de comparação para situações de possíveis quedas extremas de sinal ou eventos climáticos raros que possam causar considerável interferência de sinal. Mesmo nessas condições, a estratégia TDMEN-CACHE demonstrou resultados satisfatórios.

Por fim, para obter um comparativo mais preciso, também foi avaliada a rede sem balanceamento de cargas quando empregado perdas de 1% e então os resultados forma comparados aos resultados do TDMEN-CACHE com a mesma taxa de perda. As Figuras 23 e 24 resumem os dados que ilustram esta comparação. Mesmo a abordagem sem balanceamento de carga deve enviar os resultados do processamento do evento pela rede até o Sink através de esquema multihop, possibilitando assim a ocorrência de perda do mesmo.

Pela Figura 23, é notável que na estratégia TDMEN-CACHE, o primeiro ponto de

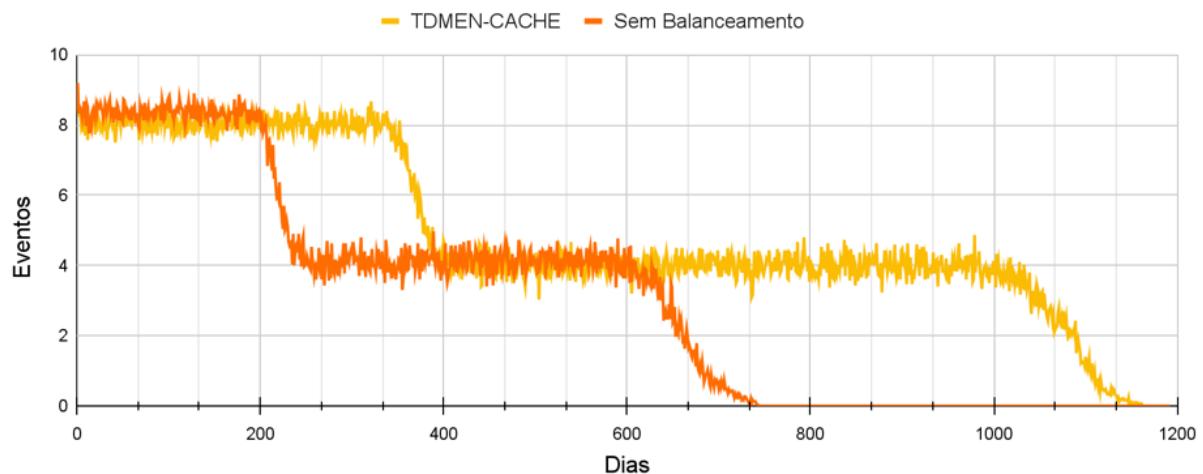
Tabela 12 – Estudo de caso 1: Número de eventos detectados pela rede usando TDMEN-CACHE e pela rede sem balanceamento, ambas considerando perdas de 1%.

Método	Med - Var
TDMEN-CACHE	5814 - 78
Sem Balanceamento	3655 - 70

Fonte: Elaborada pelo autor.

inflexão no gráfico ocorre significativamente mais tarde. Esse ponto marca a primeira onda de falhas nos nós da rede. A abordagem TDMEN-CACHE proporcionou a esses nós uma duração aproximadamente 69% maior em comparação com os resultados obtidos na rede sem balanceamento de carga. No que diz respeito à duração total da rede, a abordagem TDMEN-CACHE apresentou um aumento significativo, proporcionando 61% mais dias em comparação com a estratégia sem balanceamento.

Figura 23 – Estudo de Caso 1: Eventos sensoreados por dia pela TDMEN-CACHE em comparativo com a RSSF sem balanceamento, ambas considerando perdas de 1%.

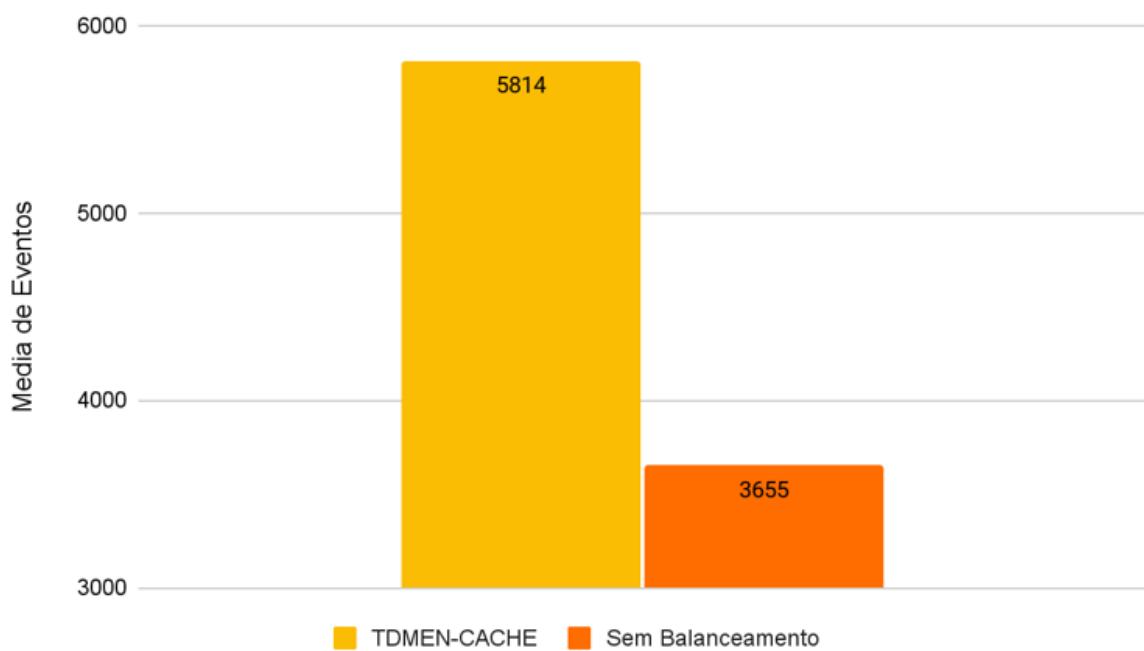


Fonte: Elaborada pelo autor.

Na Figura 24, o número total de eventos detectados pela rede é ilustrado por meio de um gráfico de barras. É possível observar que na abordagem TDMEN-CACHE, o número de eventos detectados foi consideravelmente maior, chegando a detectar 69% mais eventos do que na abordagem sem balanceamento. Esses valores podem ser mais claramente identificados na Tabela 12, onde esse significativo aumento no número de eventos detectados se deve ao fato de a abordagem TDMEN-CACHE prolongar a vida útil da rede, permitindo que os nós detectem eventos por um período mais longo.

Por meio dos resultados obtidos nos experimentos de simulação com perdas da abordagem TDMEN-CACHE, observou-se uma tendência decrescente linear na redução do número de eventos detectados à medida que o percentual de perda de pacotes

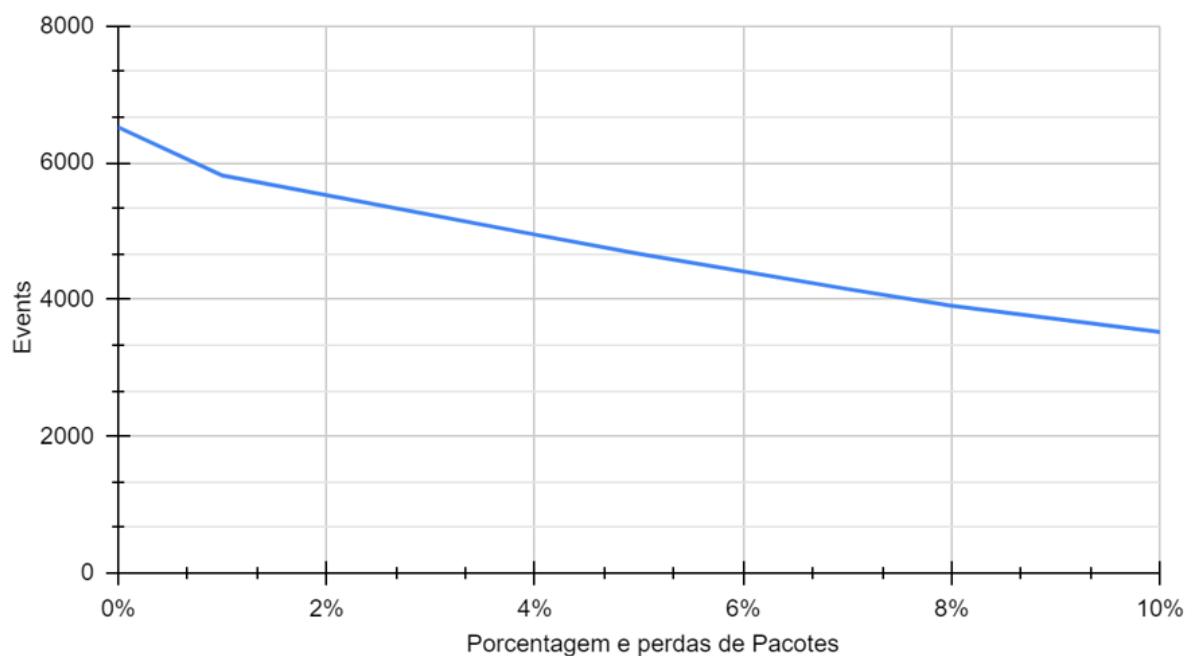
Figura 24 – Estudo de caso 1: Número de eventos detectado na rede com TDMEN-CACHE em comparativo com a RSSF sem balanceamento, ambas considerando perdas de 1%.



Fonte: Elaborada pelo autor.

aumenta. Essa tendência pode ser observada no gráfico da Figura 25 onde resultados com percentuais de perdas adicionais aos usados nas demais simulações foram executados e sumarizados nesse gráfico. Ainda, com base nos resultados obtidos e analisando a linha de tendência do gráfico, pode-se inferir que quando o nível de perda de pacotes atinge 10%, o número de eventos perdidos são aproximadamente 46%.

Figura 25 – Estudo de Caso 1: Impacto da perda de pacotes na detecção de eventos na rede usando TDMEN-CACHE.



Fonte: Elaborada pelo autor.

6 ESTUDO DE CASO 2: DIVISÃO DE TAREFAS EM RSSFM COM REDUNDÂNCIA DE SENSOREAMENTO

Neste capítulo será apresentado um estudo de caso envolvendo redes com sobreposição de sensoreamento (Redundância espacial). Este estudo de caso será empregado para avaliar a TDMEN-INTERSEC. Na Seção 6.1 será detalhada a aplicação alvo, bem como será descrita sua caracterização empregada nas simulações. Já na Seção 6.2, a abordagem TDMEN-INTERSEC será comparada com outras abordagens estado-da-arte em RSSF reativas com redundância de sensoriamento.

6.1 Modelagem do Estudo de Caso TDMEN-INTERSEC

Este segundo estudo de caso considera uma RSSFM empregada no monitoramento e identificação de pássaros por meio do canto dos mesmos. O evento que gatilha os nós reativos da rede de sensores agora representa um canto de um pássaro. A carga de processamento desta aplicação refere-se ao processamento e áudio e a identificação de padrão em sons de pássaros.

Para auxiliar na caracterização destas operações de processamento, foi empregada a API BirdNET-Analyzer¹ (Pérez-granados, 2023), a qual permite o uso de Inteligência Artificial para a identificação de mais de 6000 espécies de pássaros ao redor do mundo. Esta API provê ainda funcionalidades adicionais úteis para a nossa abordagem TDMEN, como a segmentação de áudio, a qual poderá ser utilizada para dividir e balancear a carga de processamentos dos eventos.

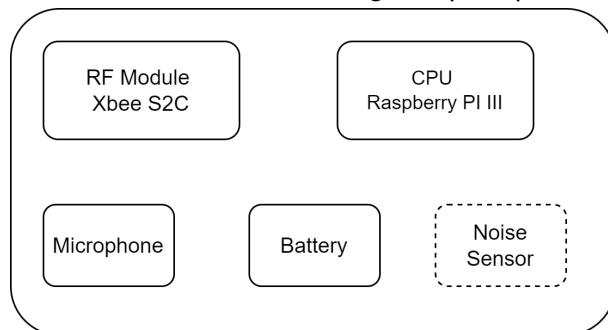
Para estimar os custos da aplicação, similar ao que foi executado no estudo de caso 1, foram realizados experimentos práticos considerando a arquitetura ilustrada na Figura 26. Essa arquitetura possui a mesma bateria, CPU e módulo de comunicação apresentados no primeiro estudo de caso. O novo design de hardware se diferencia apenas pela substituição do módulo de câmera por um de microfone, este utilizado para gravar os cantos de pássaros, e pela troca do sensor de passagem por um sensor de ruídos, necessário para sinalizar a ocorrência de algum som ambiente e ativar

¹<https://github.com/kahst/BirdNET-Analyzer>

posteriormente o microfone para gravação.

Para possibilitar a identificação e avaliação dos sons de pássaros, a API BirdNET-Analyzer foi previamente instalada na CPU Raspberry Pi III. Esta API foi então utilizada para estimar o custo energético referente ao processamento de áudio. Os experimentos executados para avaliar o custo de processamento consideraram uma amostra de áudio que representava 120 segundos, durante os quais haviam 7 pássaros cantando, totalizando 320 000 bytes de informação.

Figura 26 – Diagrama de blocos da Armadilha inteligente para passaros.



Fonte: Elaborada pelo autor.

Para estimar o consumo referente a transmissão, também foi considerado o mesmo modelo de rádio empregado no estudo de caso 1, bem como o mesmo protocolo de comunicação. O tamanho do segmento usado foi de 8000 Bytes, sendo este o tamanho exato obtido após o preprocessamento do áudio base de 120s usado nos experimentos. O preprocessamento deste som base gera 7 segmentos, um para cada pássaro, onde a duração do canto de pássaro contido em cada segmento é de 3 segundos. Considerando o payload máximo (104 Bytes), foi determinado o número de pacotes (77) a serem enviados, a fim de que todo o segmento pudesse ser transmitido. Este número de pacotes foi multiplicado pelo número de bytes do cabeçalho gerando o volume de dados a ser transmitido (9848 Bytes). Considerando que a taxa total de transmissão de cada nó é de 250.000 bites por segundo ou convertendo 31.250 bytes por segundo, e o custo para utilizar completamente essa taxa é de 33 mA, dividindo o valor total de bytes do segmento identificado (9848 bytes) por essa taxa de transmissão, obtemos um percentual de uso da largura de banda de 31%, onde, dessa porcentagem do custo de energia, resultam os 10,23 mA. O cálculo desse custo é sumarizado na Equação 3.

Para contabilizar os custos da abordagem TDMEN-INTERSEC operando neste novo cenário, considerando o uso da API BirdNET-Analyzer, a abordagem foi previamente instalada em um Raspberry Pi III a mesma usada na arquitetura prática apresentada na Seção 5.1. Isso foi feito para contabilizar o custo energético associado à execução de cada uma das operações. Os resultados obtidos durante

Tabela 13 – Estudo de Caso 2: Custo de energia de tarefas para nova abordagem.

Nome da Tarefa	Custo
SEND_SEGMENT	10.23 mAs
SEND_DISCOVERY	0.038016 mAs
SEND_APP_MESSAGE	0.038016 mAs
IDLE_COST	0.1 mAs
PREPROCESSING	6560 mAs
PROCESSING_SEGMENT	4963 mAs

Fonte: Elaborada pelo autor.

Tabela 14 – Estudo de Caso 2: Custo de energia de tarefas para PS, ANT e Sem Balanceamento.

Nome da Tarefa	Custo
SEND_APP_MESSAGE	0.038016 mAs
IDLE_COST	0.1 mAs
PROCESSING_EVENT	41304 mAs

Fonte: Elaborada pelo autor.

os testes de custos das operações estão resumidos na Tabela 13. Os custos de SEND_DISCOVERY e SEND_APP_MESSAGE mantiveram-se os mesmos dos demais estudos de caso devido ao tamanho da mensagem em bytes ser igual em ambos os casos, apesar de as estruturas e a frequência com que ocorrem serem distintas. O mesmo ocorre para o IDLE_COST, onde o valor permanece idêntico em todos os experimentos devido ao uso do mesmo hardware em todas as simulações.

Já na Tabela 14, são apresentados os custos utilizados para as demais estratégias (PS, Ant e Sem balanceamento), usadas como comparativos de desempenho para a abordagem proposta. Nessas estratégias, como um único nodo executa todo o processamento do evento multimídia, o custo de PROCESSING_EVENT mostrado na tabela engloba toda a tarefa de preprocessar o áudio, identificar os sons das aves e catalogar a quais espécies esses cantos pertencem. Este custo foi calculado considerando o tempo de CPU requerido para executar esta tarefa (51,63 segundos) e a corrente consumida (800 mA), totalizando os 41304 mAs. Ainda, os custos de SEND_APP_MESSAGE e IDLE_COST se mantiveram os mesmos devido ao tamanho da mensagem e arquitetura serem as mesmas respectivamente.

Tamanho do Segmento:	8000Bytes	
Tamanho do Header:	24Bytes	
Tamanho do Payload:	104Bytes	
Taxa de Transmissão:	250Kbps	
Custo Por Uso Total da Transmissão:	$33mA$	
	$8000/104 = 77\text{pacotes}$	(Total de pacotes)
Gasto com Header	$77\text{pacotes} * 24b\text{Bytes} = 1848\text{Bytes}$	
Total a ser transmitido	$8000 + 1848 = 9848\text{Bytes}$	
Percentual de Uso da Taxa	$9848/((250 * 1024)/8) = 0.31$	
Custo Total por Percentual Usado	$33mA * 0.31 = 10.23mA$	(Por Segmento)
		(3)

O custo estimado para realizar o pré-processamento do áudio é de 6560 mAs. Os cálculos desses custos podem ser visualizados em 4, onde o áudio utilizado nas simulações tinha uma duração de 120 segundos, contendo o canto de 7 pássaros. O tempo identificado para segmentar/pré-processar este áudio foi de 8.2 segundos com o uso da Raspberry Pi III e com a integração da API BirdNET ao TDMEN. O custo para cada segundo de processamento nesta arquitetura é de 800 mA, resultando em um custo total identificado de 6560 mAs.

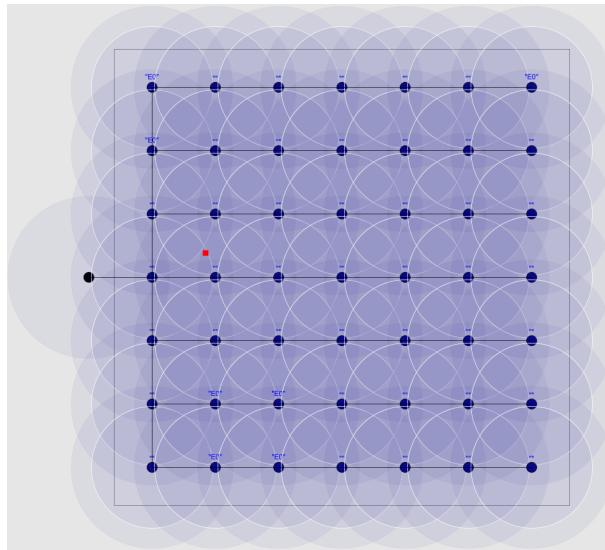
Custo 1s Processamento:	
Tempo para Processar Audio 120s:	$8,2s$
Custo Segmentar audio 120s	$800mA * 8,2s = 6560mA$ (Com 7 cantos) (4)

Através de experimentos, observou-se que o processamento de um segmento para identificar qual pássaro é o responsável pelo som leva aproximadamente 6.2 segundos de execução da Raspberry Pi III, gerando um custo de 4963 mAs, onde os cálculos para identificar este custo estão sumarizados em 5. Por fim, como o tamanho das demais mensagens do TDMEN permanece o mesmo que nos testes realizados no outro estudo de caso, os demais custos permaneceram inalterados.

Custo 1s Processamento:	$800mA$
Tempo para Processar 1 Segmento:	6,2s
Custo Processar Segmento	$800mA * 6,204s = 4963mA\cdot s$ (Cada) (5)

Estes custos estimados foram então incorporados aos modelos de simulação empregados para avaliar as soluções de rede com diferentes balanceamento de carga e sem balanceamento de carga. Os resultados das simulações foram repetidas 30 vezes para cada cenário e os resultados de média foram então comparados.

Figura 27 – Cenário de simulação para monitoramento do canto de pássaros.



Fonte: Elaborada pelo autor.

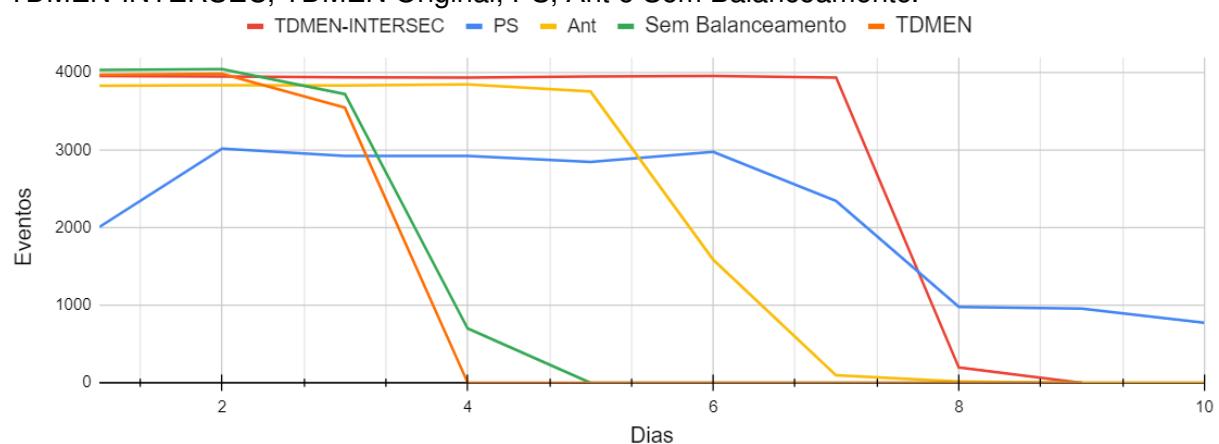
Na Figura 27, é apresentada uma captura de tela da simulação da RSSF abordando o estudo de caso do canto de pássaros. Na imagem, cada um dos 49 nodos são modelados pela classe *TaskDivisionIntersecProtocol* em um hardware conforme apresentado na Figura 26. Esses nodos estão dispostos de modo a permitir que até 3 nodos detectem um mesmo evento, sendo este evento o canto de um pássaro. O raio de comunicação desses nodos é definido pelo atributo do simulador Eboracum *commCover*, com valor 160, já o alcance dos sensores de comunicação é dado pelo atributo *sensorCover*, com valor 120. Neste estudo de caso apresentado, a classe *StochasticDailyJumperEvent* é empregada para a representação do comportamento de aves cantando em uma floresta. Os eventos aparecem no espaço seguindo uma distribuição Uniforme e a frequência dos eventos é dada por uma distribuição de Poisson, com um intervalo de ocorrência entre eventos de [1,100] segundos.

6.2 Comparativo TDMEN-INTERSEC com demais estratégias

Esta seção apresenta os resultados obtidos pela rede dedicada à identificação de pássaros quando usando a abordagem proposta TDMEN-INTERSEC. Para fins de comparação, também foram realizados experimentos empregando uma rede sem balanceamento de carga e redes empregando as abordagens Ant (Ferreira; Brisolara; Indrusiak, 2015) e PS (Caliskanelli; Harbin; Indrusiak; Mitchell; Polack; Chesmore, 2013), os quais serão comparados.

A primeira análise comparativa tem por objetivo visualizar o desempenho da rede ao longo de seu tempo de vida. Na Figura 28, é apresentado um gráfico que representa, ao longo do tempo, o número de eventos detectados por cada uma das abordagens avaliadas. Com base nos resultados experimentais, a nossa abordagem TDMEN-INTERSEC foi a que manteve todos os nós da rede vivos por mais tempo. Como a abordagem TDMEN-INTERSEC manteve todos os nós sensoreando por mais tempo, essa estratégia também foi a que perdeu menos eventos e, consequentemente, detectou um total maior de eventos do que as demais abordagens. Por outro lado, os resultados indicaram que a abordagem PS foi a que manteve a rede viva por mais tempo, no entanto, esta também foi a abordagem que teve o maior percentual de perda de eventos, mesmo nos primeiros dias da rede. Já a abordagem Ant teve um comportamento mais similar ao da TDMEN-INTERSEC, conseguindo um balanceamento da carga entre os nodos, porém limitado. Neste estudo de caso, como há sobreposição na área de sensoriamento, a rede sem balanceamento de carga e a abordagem TDMEN Original sofrem com possíveis processamentos duplicados, o que é evitado nas demais abordagens.

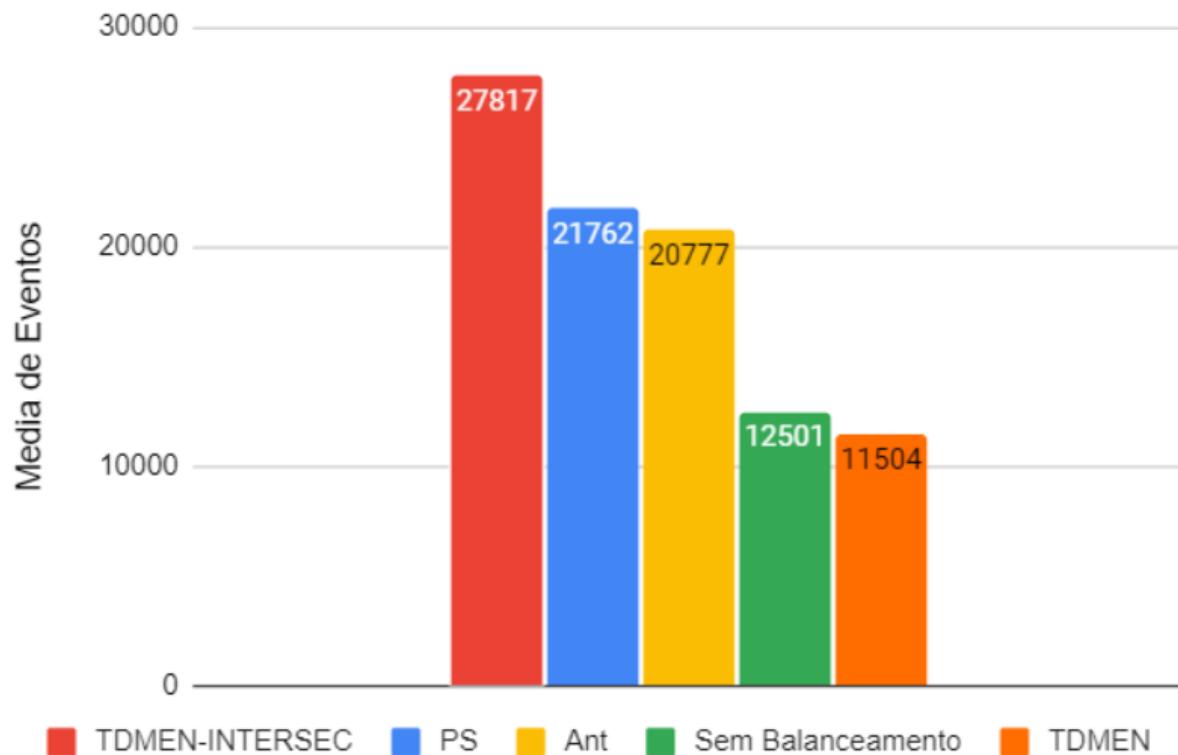
Figura 28 – Estudo de Caso 2: Comparativo de detecção de eventos por dia das estratégias TDMEN-INTERSEC, TDMEN Original, PS, Ant e Sem Balanceamento.



Fonte: Elaborada pelo autor.

Na Figura 29, é apresentada a média de eventos sensoreados por cada uma das abordagens durante as 30 repetições executadas. Pode-se observar que, das cinco

Figura 29 – Estudo de Caso 2: Média de Eventos detectados pelas estratégias TDMEN-INTERSEC, TDMEN Original, PS, Ant e Sem Balanceamento.



Fonte: Elaborada pelo autor.

abordagens estudadas, a TDMEN-INTERSEC foi a que sensoreou mais eventos, detectando aproximadamente 28% mais eventos do que a abordagem PS, 34% mais do que o Ant, 123% mais do que uma rede que não realiza nenhum balanceamento de carga e 142% mais eventos que a abordagem TDMEN original. Com base nestes resultados, identifica-se que a abordagem TDMEN-INTERSEC tem a capacidade de sensorear mais eventos ao longo do tempo em comparação com as demais abordagens estudadas. A Tabela 15 resume os resultados apresentados na Figura 29, incluindo a média do total de eventos e a variância obtida considerando 30 execuções de cada modelo de simulação.

Tabela 15 – Total de eventos detectados TDMEN-INTERSEC, PS, Ant e sem balanceamento.

Método	Med - Var
TDMEN-INTERSEC	27817 - 43
PS	21762 - 69
Ant	20777 - 318
Sem Balanceamento	12501 - 42
TDMEN Original	11504 - 42

Fonte: Elaborada pelo autor.

Pode-se identificar pela Tabela 16, que a abordagem TDMEN-INTERSEC manteve todos os nós da rede vivos por 17% mais tempo do que a estratégia PS, 40% mais do que o Ant e 250% mais do que uma abordagem sem balanceamento ou que a abordagem TDMEN Original. É importante ressaltar que, apesar da abordagem PS manter todos os nós vivos por 6 dias, devido à forma como a estratégia funciona, parte da rede acaba ficando sem cobertura, resultando em perdas de eventos, mesmo sem nenhum nó ter morrido.

Por fim, ao compararmos a duração total da rede, observamos que a estratégia PS manteve a rede viva por aproximadamente 20% mais tempo do que a TDMEN-INTERSEC, 30% mais do que o Ant, 50% mais que a abordagem TDMEN-Original e 80% mais do que uma abordagem sem balanceamento de carga. No entanto, mesmo que a abordagem PS tenha se mantido em funcionamento por mais tempo do que a TDMEN-INTERSEC, ela ainda sensoreou significativamente menos eventos.

Tabela 16 – Tempo de vida da rede TDMEN-INTERSEC, PS, Ant e Sem Balanceamento.

Método	Rede Completa (Med - Var)	Duração Total (Med - Var)
TDMEN	7 - 0.8	7.97 - 0.18
PS	6 - 0.32	10 - 0
Ant	5 - 0.28	7.33 - 0.8
Sem Balanceamento	2 - 0.1	4 - 0

Fonte: Elaborada pelo autor.

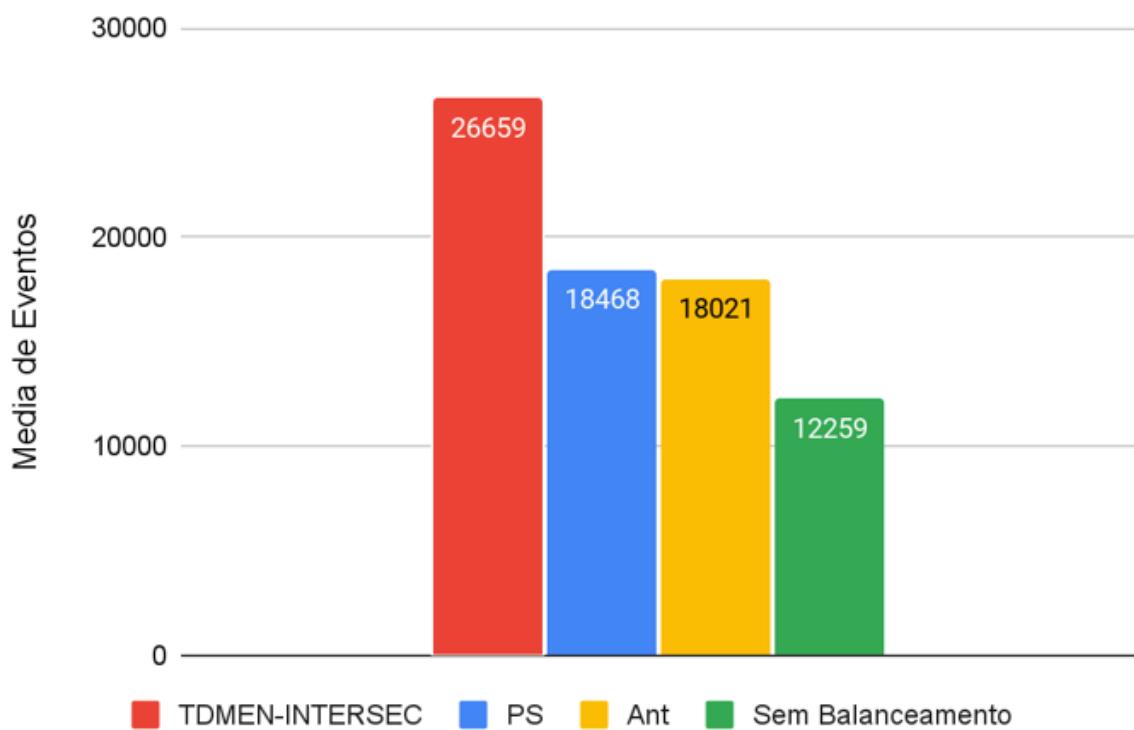
Com o objetivo de se aproximar ao máximo dos resultados de testes em um ambiente real, foram realizadas simulações com perdas de pacotes para identificar o impacto que interferências externas podem gerar em cada um dos algoritmos estudados. O percentual de perda de pacotes utilizado nos testes foi de 1%, o qual foi identificado na Seção 5.5 como sendo o mais adequado.

Na Figura 30, o número de eventos processados pela rede ao longo de sua vida foi contabilizado e sumarizado neste gráfico. Entre as abordagens analisadas neste estudo de caso, aquela que sofreu menos interferência na perda de pacotes foi a que não executa nenhum balanceamento, tendo uma redução de 2% no número de eventos sensoreados. Essa abordagem não emprega nenhuma comunicação adicional entre nodos, visto que não realiza nenhuma divisão de trabalho entre nodos vizinhos, nem mesmo evita que dois nodos realizem o processamento de um mesmo evento. Consequentemente, esta abordagem sofrerá menos impacto quando ocorrem perdas de pacotes, deixando de detectar, em média, 242 eventos.

Por outro lado, as abordagens Ant e Ps sofreram impacto de forma semelhante em relação à perda de pacotes, cada uma com uma redução do número de eventos detectados de 2756 e 3294, respectivamente, o que representa uma redução de 13% de eventos sensoreados pela abordagem Ant e 15% para a abordagem PS. Devido

à necessidade dessas abordagens de realizar troca de mensagens adicionais entre nodos a fim de decidir quem processará e assim evitar o processamento duplicado. Consequentemente, estas abordagens terão um impacto consideravelmente maior em termos de perda de eventos, quando comparado a abordagem sem balanceamento de carga. Já a abordagem TDMEN-INTERSEC proposta nesta tese não sofreu tanto impacto quanto as demais abordagens de balanceamento, perdendo em média 1158 eventos, o que representa 4% a menos. Esse resultado é atribuído ao pelos diferentes tipos de mensagens, onde algumas, como o hello, não geram a perda do evento, dado que mais de um nodo recebe essa mesma mensagem(*broadcast*), o que aumenta a chance de sempre existir algum nodo que receba o comando e responda ao nodo mestre.

Figura 30 – Eventos detectados em um meio com perdas pelas estratégias TDMEN-INTERSEC, PS, Ant e Sem Balanceamento.



Fonte: Elaborada pelo autor.

A Tabela 17 apresenta, de maneira simplificada, o percentual médio de eventos a menos detectados por cada uma das abordagens em comparação com a abordagem proposta TDMEN-INTERSEC. Observa-se que a nova estratégia proposta conseguiu detectar aproximadamente 117% mais eventos do que uma estratégia sem balanceamento e 44% mais do que a estratégia bioinspirada PS, destacando a eficácia da abordagem proposta.

Tabela 17 – Comparativo total de eventos detectados com perdas pelas estratégias: PS, Ant e sem Balanceamento em comparativo a abordagem proposta TDMEN-INTERSEC.

Método	X TDMEN-INTERSEC
PS	44% menos eventos
Ant	48% menos eventos
Sem Balanceamento	117% menos eventos

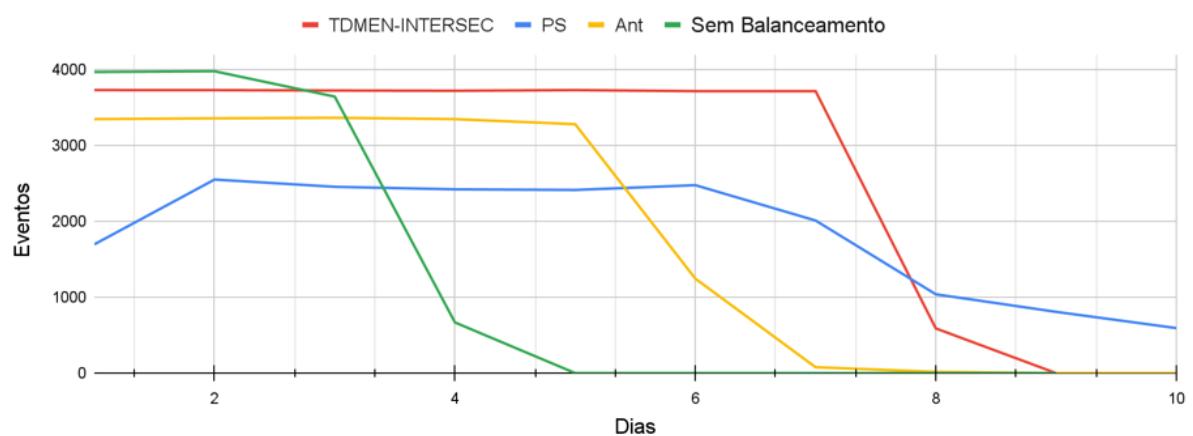
Fonte: Elaborada pelo autor.

Comparação entre as Figuras 31 e 28 revela que, mesmo com a introdução de perdas de pacotes na simulação da rede, o comportamento das estratégias apresentadas permaneceu semelhante. Entre as abordagens estudadas, a estratégia sem balanceamento, como esperado, foi a menos afetada, sendo impactada apenas pelas perdas de pacotes que ocorreram durante o envio do resultado do processamento do evento de um nó até o Sink. As estratégias PS e Ant foram afetadas de maneira similar, com redução equivalente no número de eventos capturados. Mesmo com essa redução no número de eventos detectados, ambas as abordagens PS e Ant continuam sendo consideravelmente superiores à abordagem sem balanceamento, quando comparadas ao número total de eventos sensoreados e ao número de dias que cada abordagem conseguiu manter a rede ativa.

Além disso, ao comparar as Figuras 31 e 28, percebe-se ainda que a abordagem TDMEN proposta foi menos afetada em comparação com as estratégias com balanceamento de carga, pois realiza menos trocas de mensagens do que as estratégias PS e Ant, garantindo assim um impacto menor no número de eventos perdidos devido à perda de pacotes. Dessa forma, entre as abordagens analisadas, a TDMEN foi consideravelmente superior às demais estratégias, quando analisado o número de eventos sensoreados e a capacidade em manter todos os nodos da rede vivos pelo máximo de tempo possível.

Por fim, na Figura 31, mesmo em um ambiente com perdas de pacotes, a abordagem proposta mostrou-se significativamente superior às demais estratégias. A estratégia TDMEN manteve todos os nós da rede ativos por mais tempo e processou e identificou mais eventos do que as demais. Na abordagem sem balanceamento, os primeiros nos falharam já no segundo dia de processamento. A estratégia Ant conseguiu manter todos os nós ativos até o quinto dia, enquanto a abordagem PS conseguiu manter até o sexto dia, porém com um custo significativo em termos de trocas de mensagens e maior número de eventos perdidos.

Figura 31 – Detecção de eventos por dia em um meio com perdas pelas estratégias TDMEN, PS, Ant e Sem Balanceamento.



Fonte: Elaborada pelo autor.

7 CONCLUSÃO

Esta tese propôs a divisão de tarefas de processamento em redes de sensores sem fio multimídia, visando a distribuição mais homogênea da carga de trabalho entre os nodos. Esta abordagem nomeada TDMEN permite que tarefas sejam quebradas em subtarefas e distribuídas entre os nodos vizinhos, levando em consideração a bateria dos nodos na distribuição das subtarefas. Esta abordagem foi publicada em (Souza; Brisolara; Ferreira, 2022) e os resultados experimentais demonstraram melhorias de até 41% no tempo de duração da rede, em comparação com uma rede sem平衡amento de carga.

Nos testes com perdas de pacotes, a abordagem TDMEN-CACHE manteve resultados superiores em comparação com a abordagem sem balanceamento. A abordagem proposta detectou 48% mais eventos no total e proporcionou um aumento de 69% no tempo médio de vida para a rede completa, isto é, sem nenhum nodo inativo.

Tanto a TDMEN quanto a TDMEN-CACHE foram avaliadas em um cenário de rede sem redundância espacial entre os nodos, ou seja, nodos diferentes não poderiam detectar um mesmo evento. Visando também suportar cenários de redes com sobreposição de sensoreamento, esta tese também propôs uma nova abordagem, denominada TDMEN-INTERSEC. Testes de desempenho comparativos com outras três abordagens (PS, Ant e Sem Balanceamento) foram realizados empregando como estudo de caso uma rede dedicada ao monitoramento e reconhecimento de cantos de pássaros.

A estratégia adotada pela TDMEN-INTERSEC explora detecções simultâneas do mesmo evento pelos nós, os quais poderão colaborar para processar o evento sensoreado. Como esses nodos detectaram o mesmo evento, não é necessário gastar energia compartilhando dados do evento; eles precisam apenas cooperativamente determinar quais partes do evento irão processar. Além disso, quando necessário, esses nodos podem enviar segmentos dos eventos a outros vizinhos que possuam mais energia para participarem do processamento.

Nos testes realizados, a abordagem TDMEN-INTERSEC sensoreou 28% mais eventos do que a abordagem PS, 34% mais do que a Ant e 123% mais do que a estratégia Sem Balanceamento. Isso demonstra que a abordagem proposta possui o

potencial de sensorear mais eventos ao longo do tempo em comparação com outras estratégias de balanceamento de carga. Além disso, em relação à duração da rede, a abordagem TDMEN-INTERSEC conseguiu manter todos os nós da rede ativos por mais tempo do que as outras estratégias: 17% mais que a PS, 40% mais que a Ant e 250% mais que a Sem Balanceamento.

Ainda, testes com perdas de pacotes foram executados com objetivo de identificar a capacidade da abordagem TDMEN de executar nestes meios. Os resultados demonstraram que a abordagem proposta obteve resultados ainda mais significativos em comparação com as demais estratégias analisadas. Onde ao comparar o impacto sofrido pela simulação de perda de dados na rede, a abordagem TDMEN teve uma redução de apenas 4% de eventos sensoreados, já as abordagens Ant e PS tiveram uma queda de 13% e 15% respectivamente. Assim, ao comparar o número total de eventos sensoreados, a abordagem TDMEN-INTERSEC detectou 44% mais eventos do que a abordagem PS, 48% mais do que a Ant e 117% mais do que a abordagem Sem Balanceamento.

A TDMEN pode ser empregada em redes com sobreposição ou sem sobreposição, como demonstrado nos estudos de caso, diferentemente dos algoritmos considerados estado-da-arte para balanceamento de carga em redes reativas, Ant e PS, os quais requerem a sobreposição para realizar o balanceamento de carga. Assim, as abordagens propostas neste trabalho apresentam as vantagens:

1. **Aplicabilidade em redes com interseção de sensoriamento e redes sem interseção:** A estratégia *TDMEN* se mostrou eficaz em ambos os cenários, proporcionando flexibilidade de aplicação.

2. **Balanceamento de carga energeticamente mais eficiente:** Comparada às abordagens bioinspiradas, a *TDMEN* apresenta uma distribuição de carga mais eficiente em termos energéticos, prolongando em até 40% a vida útil da rede e consequentemente sensoreando significativamente mais eventos.

3. **Menor impacto em redes sujeitas a interferências externas e perdas de pacotes:** Sofrendo menos impacto do que as abordagens bioinspiradas analisadas, a *TDMEN* mantém ainda uma eficiência maior do que uma abordagem local mesmo com até 7% de perdas de pacotes.

Esses resultados destacam o potencial das abordagens *TDMEN* em melhorar diversos aspectos operacionais em redes de sensores sem fio multimídia, contribuindo para a eficiência e aprimoramento do desempenho do sistema na totalidade.

A distribuição das tarefas de processamento e paralelismo inerente desta cooperação entre nodos, aliada a decisão de processar local e só enviar o resultado à estação base através da rede multihop, pode resultar também em um tempo menor para que o resultado do processamento chegue até a estação base. Nossos experimentos, por limitações dos relatórios oferecidos pelo simulador, não avaliaram o tempo de resposta

da rede. Assim, avaliar os efeitos do emprego de nossas abordagens nesta métrica seria uma discussão relevante como trabalho futuro.

Outro possível efeito positivo de nossas abordagens, seria o aumento da privacidade dos dados, uma vez que evita-se a necessidade do envio dos dados multimídia de nodo em nodo até o *Sink*. Como trabalho futuro, este impacto será discutido com revisão de trabalhos na área de segurança e privacidade.

Como em nossa abordagem as comunicações entre nodos são executadas por meio de mensagens de Broadcast, os nodos vizinhos podem ler as mensagens enviadas uns para os outros. Deste modo, tomando como base o funcionamento de algoritmos de inteligência de enxame, os nodos podem inferir dados e informações de seus vizinhos quando estes executam alguma troca de mensagens. A dedução destes dados pode evitar a necessidade da troca de algumas mensagens e consequentemente, reduzir o consumo de energia da rede.

REFERÊNCIAS

- ABAZEED, M.; FAISAL, N.; ALI, A. Cross-layer multipath routing scheme for wireless multimedia sensor network. **Wireless Networks**, [S.I.], v.25, p.4887–4901, 2019.
- ALKHATIB, A. A. A.; BAICHER, G. S. Wireless sensor network architecture. In: INTERNATIONAL CONFERENCE ON COMPUTER NETWORKS AND COMMUNICATION SYSTEMS (CNCS 2012), 2012., 2012. **Anais...** [S.I.: s.n.], 2012.
- AMUTHA, J.; SHARMA, S.; NAGAR, J. WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues. **Wireless Personal Communications**, [S.I.], v.111, p.1089–1115, 2020.
- BAVARVA, A.; JANI, P. V.; GHETIYA, K. Performance improvement of wireless multimedia sensor networks using MIMO and compressive sensing. **Journal of Communications and Information Networks**, [S.I.], v.3, n.1, p.84–90, 2018.
- CALISKANELLI, I.; HARBIN, J.; INDRUSIAK, L. S.; MITCHELL, P.; POLACK, F.; CHESMORE, D. Bioinspired load balancing in large-scale WSNs using pheromone signalling. **International Journal of Distributed Sensor Networks**, [S.I.], v.9, n.7, p.172012, 2013.
- CHIWARIRO, R.; THANGADURAI, N. Deep Learning-based Prioritized Packet Classification and Optimal Route Selection in Wireless Multimedia Sensor Networks. **IETE Journal of Research**, [S.I.], p.1–19, 2023.
- CLEMENTE-ORTA, G.; MADEIRA, F.; BATUECAS, I.; SOSSAI, S.; JUÁREZ-ESCARIO, A.; ALBAJES, R. Changes in landscape composition influence the abundance of insects on maize: the role of fruit orchards and alfalfa crops. **Agriculture, Ecosystems & Environment**, [S.I.], v.291, p.106805, 2020.
- DEEBAK, B. D.; AL-TURJMAN, F. A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks. **Ad Hoc Networks**, [S.I.], v.97, p.102022, 2020.

DEMETILLO, A. T.; JAPITANA, M. V.; TABOADA, E. B. A system for monitoring water quality in a large aquatic area using wireless sensor network technology. **Sustainable Environment Research**, [S.I.], v.29, p.1–9, 2019.

DREW, R. A.; YUVAL, B. The evolution of fruit fly feeding behavior. In: **Fruit Flies (Tephritidae)**. [S.I.]: CRC press, 1999. p.749–768.

FAKHRI, A. B.; GHARGHAN, S. K.; MOHAMMED, S. L. Path-loss modelling for WSN deployment in indoor and outdoor environments for medical applications. **International Journal of Engineering & Technology**, [S.I.], v.7, n.3, p.1666–1671, 2018.

FERREIRA, P. R.; BRISOLARA, L.; INDRUSIAK, L. S. Decentralised load balancing in event-triggered wsns based on ant colony work division. In: EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS (SEAA), 41., 2015. **Anais...** [S.I.: s.n.], 2015. p.69–75.

FREITAS, L. C.; MARTINS, V. A. M.; AGUIAR, M. S. de; BRISOLARA, L. B. de; FERREIRA JR., P. R. Deep Learning Embedded into Smart Traps for Fruit Insect Pests Detection. **ACM Trans. Intell. Syst. Technol.**, New York, NY, USA, jul 2022. Just Accepted.

FREITAS, L. C.; MARTINS, V. A. M.; FERREIRA JÚNIOR, P. R.; BRISOLARA, L. B. d. Modelagem de Aplicações de Sensoriamento Visando Mapeamento Adaptativo em Redes de Sensores Sem Fio. In: CONGRESSO DE INOVAÇÃO TECNOLÓGICA, 2., 2018, Pelotas, Brasil. **Anais...** [S.I.: s.n.], 2018.

FREITAS, L.; MARTINS, V.; AGUIAR, M. de; BRISOLARA, L. de; FERREIRA, P. Deep learning embedded into smart traps for fruit insect pests detection. **ACM Transactions on Intelligent Systems and Technology**, [S.I.], v.14, n.1, p.1–24, 2022.

GENTA, A.; LOBIYAL, D.; ABAWAJY, J. H. Energy efficient multipath routing algorithm for wireless multimedia sensor network. **Sensors**, [S.I.], v.19, n.17, p.3642, 2019.

HUSSIEN, A. A.; MATLOOB, S. I. The comparative study some of reactive and proactive routing protocols in the wireless sensor network. **Journal of University of Babylon for Engineering Sciences**, [S.I.], v.26, n.4, p.195–207, 2018.

INC., D. I. **XBee/XBee-PRO S2C Zigbee**: RF Module. 2020. Disponível em: <<https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf>>. Acesso em: 2021-07-13.

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

- KETHIREDDY, S. R.; RALLAPALLI, P. V.; CHILAKALA, L. R.; DEVULAPALLI, P. K. Energy Efficient Cooperative Image Transmission in Multi-Hop Wireless Multi-Media Sensor Networks. **International Journal of Intelligent Engineering & Systems**, [S.I.], v.16, n.5, 2023.
- KHALAF, O. I.; SABBAR, B. M. An overview on wireless sensor networks and finding optimal location of nodes. **Periodicals of Engineering and Natural Sciences (PEN)**, [S.I.], v.7, n.3, p.1096–1101, 2019.
- LIMA, M. C. F.; ALMEIDA LEANDRO, M. E. D. de; VALERO, C.; CORONEL, L. C. P.; BAZZO, C. O. G. Automatic detection and monitoring of insect pests—A review. **Agriculture**, [S.I.], v.10, n.5, p.161, 2020.
- LONG, C.; CAO, Y.; JIANG, T.; ZHANG, Q. Edge computing framework for cooperative video processing in multimedia IoT systems. **IEEE Transactions on Multimedia**, [S.I.], v.20, n.5, p.1126–1139, 2017.
- MARQUES, M. R. S.; BRISOLARA, L.; FERREIRA, P. R.; INDRUSIAK, L. S. Eboracum: An extensible framework for high-level modeling and evaluation of reactive and adaptable WSNs. In: IEEE 21ST INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION (ETFA), 2016., 2016. **Anais...** [S.I.: s.n.], 2016. p.1–8.
- MARTINS, V. A. M.; FREITAS, L. C.; AGUIAR, M. S. de; BRISOLARA, L. B. de; FERREIRA, P. R. Deep Learning applied to the Identification of Fruit Fly in Intelligent Traps. In: IX BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING (SBESC), 2019. **Anais...** [S.I.: s.n.], 2019. p.1–8.
- MATHEEN, M.; SUNDAR, S. A Novel Technique to Mitigate the Data Redundancy and to Improvise Network Lifetime Using Fuzzy Criminal Search Ebola Optimization for WMSN. **Sensors**, [S.I.], v.23, n.4, p.2218, 2023.
- MOHAN, P.; SUBRAMANI, N.; ALOTAIBI, Y.; ALGHAMDI, S.; KHALAF, O. I.; ULAGANATHAN, S. Improved metaheuristics-based clustering with multihop routing protocol for underwater wireless sensor networks. **Sensors**, [S.I.], v.22, n.4, p.1618, 2022.
- NABI, F.; JAMWAL, S.; PADMANBH, K. Wireless sensor network in precision farming for forecasting and monitoring of apple disease: a survey. **International Journal of Information Technology**, [S.I.], v.14, n.2, p.769–780, 2022.
- PEREIRA, I. A.; BRISOLARA, L. B. d.; FERREIRA, P. R. Application-Level Load Balancing for Reactive Wireless Sensor Networks: An Approach Based on Constraint Optimization Problems. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS), 2020. **Anais...** [S.I.: s.n.], 2020. p.63–76.

- PÉREZ-GRANADOS, C. BirdNET: applications, performance, pitfalls and future opportunities. **Ibis**, [S.I.], 2023.
- PRETI, M.; VERHEGGEN, F.; ANGELI, S. Insect pest monitoring with camera-equipped traps: strengths and limitations. **Journal of pest science**, [S.I.], v.94, n.2, p.203–217, 2021.
- RAJ, S. N.; BHATTACHARYYA, D.; MIDHUNCHAKKARAVARTHY, D.; KIM, T.-h. Multi-hop in clustering with mobility protocol to save the energy utilization in wireless sensor networks. **Wireless Personal Communications**, [S.I.], v.117, p.3381–3395, 2021.
- SAHITYA, G.; BALAJI, N.; NAIDU, C. D.; ABINAYA, S. Designing a wireless sensor network for precision agriculture using zigbee. In: IEEE 7TH INTERNATIONAL ADVANCE COMPUTING CONFERENCE (IACC), 2017., 2017. **Anais...** [S.I.: s.n.], 2017. p.287–291.
- SALEEM, M. M.; ALABADY, S. A. Energy-efficient multipath clustering with load balancing routing protocol for wireless multimedia sensor networks. **IET Wireless Sensor Systems**, [S.I.], 2023.
- SÁNCHEZ TURCIOS, R. A. t-Student: Usos y abusos. **Revista mexicana de cardioloxía**, [S.I.], v.26, n.1, p.59–61, 2015.
- SISODIA, A.; SWATI, H. H. Incorporation of non-fictional applications in wireless sensor networks. **International Journal of Innovative Technology and Exploring Engineering (IJITEE)**, [S.I.], v.9, n.11, 2020.
- SOUZA, W. S.; BRISOLARA, L. B. de; FERREIRA, P. R. Load Balancing Based on Multimedia Task Division for Reactive WSNs: Case Study for Pest Management. In: XII BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING (SBESC), 2022., 2022. **Anais...** [S.I.: s.n.], 2022. p.1–8.
- THAKUR, D.; KUMAR, Y.; KUMAR, A.; SINGH, P. K. Applicability of wireless sensor networks in precision agriculture: A review. **Wireless Personal Communications**, [S.I.], v.107, p.471–512, 2019.
- USMAN, M.; JAN, M. A.; HE, X.; CHEN, J. A survey on big multimedia data processing and management in smart cities. **ACM Computing Surveys (CSUR)**, [S.I.], v.52, n.3, p.1–29, 2019.
- VARSHOVI, H.; KAVIAN, Y. S.; ANSARI-ASL, K. Design and implementing wireless multimedia sensor network for movement detection using FPGA local co-processing. **Multimedia Tools and Applications**, [S.I.], v.78, p.17413–17435, 2019.

- VELUSAMY, P.; RAJENDRAN, S.; MAHENDRAN, R. K.; NASEER, S.; SHAFIQ, M.; CHOI, J.-G. Unmanned Aerial Vehicles (UAV) in precision agriculture: Applications and challenges. **Energies**, [S.I.], v.15, n.1, p.217, 2021.
- WILLIAMS, H. Birdsong and singing behavior. **Annals of the New York Academy of Sciences**, [S.I.], v.1016, n.1, p.1–30, 2004.
- YIN, L.; HUANG, Y.; CHEN, W. Reliability analysis of wireless communication based on ZigBee. In: INTERNATIONAL CONFERENCE ON ELECTRICAL & ELECTRONICS ENGINEERING AND COMPUTER SCIENCE (ICEEECS 2016), 2016., 2016. **Anais...** [S.I.: s.n.], 2016. p.67–70.
- ZANINI, F. F.; SOUZA, W. S.; BRISOLARA, L. B.; FERREIRA JR, P. R. Modelagem de uma RSSF para manejo de pragas em pomares usando o simulador Eboracum. In: XIII CONGRESSO BRASILEIRO DE AGROINFORMÁTICA (SBIAGRO), 2021. **Anais...** [S.I.: s.n.], 2021. p.197–206.