



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

SWPTMAC: Protocolo MAC de Programação Sono-Vigília e Transferência de Energia

Luan Borges dos Santos

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador

Prof. Dr. Marcelo Antonio Marotta

Brasília
2024

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

Bs Borges dos Santos, Luan
 SWPTMAC: Protocolo MAC de Programação Sono-Vigília e
Transferência de Energia / Luan Borges dos Santos;
orientador Marcelo Antonio Marotta. -- Brasília, .
 75 p.

 Dissertação(Mestrado Profissional em Computação Aplicada)
-- Universidade de Brasília, .

 1. REDE DE SENSORES SEM FIO SUBTERRÂNEOS. 2. INTERNET DAS
COISAS. 3. VIDA ÚTIL DA REDE. I. Antonio Marotta, Marcelo,
orient. II. Título.

Dedicatória

Dedico este trabalho aos meus familiares, que serviram de base para o meu desenvolvimento em todo o tempo. Dedico também à minha esposa, que abraçou os meus sonhos.

Agradecimentos

Agradeço primeiramente a Deus, que sempre me concedeu mais do que mereço. Agradeço à minha família, que se doou para me ajudar em tudo. Agradeço de maneira especial aos meus pais, Luiz e Jânia, e a minha avó, Ludumira, que foram fundamentais na minha formação. Agradeço ao meu irmão, que sempre foi o meu companheiro. Agradeço ao meu tio Jurandi, que voluntariamente custeou boa parte dos meus estudos. Agradeço minha tia Jandira, meus demais tios, primos e amigos que me apoiaram ao longo da minha jornada acadêmica. Agradeço minha esposa que permaneceu ao meu lado me incentivando nos momentos bons e difíceis ao longo do mestrado. Agradeço ao Clerimar que me motivou a inscrever para o mestrado e entrar no mundo da pesquisa científica. Agradeço a Universidade de Brasília (UnB) e ao Programa de Pós-Graduação em Computação Aplicada (PPCA) que facilitaram o meu estudo direto do interior de Minas Gerais realizando disciplinas online durante a pandemia. Agradeço a empresa COAGRIL que me liberou do serviço nos dias de aula. Agradeço aos professores do PPCA pelo conhecimento transferido, com ênfase, ao meu orientador Marcelo Antonio Marotta e ao grupo de professores que realizam pesquisas em conjunto com ele pela ajuda de valor inestimável que me deram ao longo da minha pesquisa.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

As Redes de Sensores Sem Fio Subterrâneos (RSSFS) são redes formadas por sensores enterrados que utilizam conexão sem fio para comunicarem. As RSSFS são fundamentais para o monitoramento de ambientes subterrâneos, mas enfrentam um problema de baixa Vida Útil da Rede (VUR), por serem limitadas por fontes energéticas baseadas em baterias. A VUR é definida como o tempo máximo para o qual a rede está operacional, ou seja, o tempo em que é possível obter cobertura de rede em uma região observada. Portanto, diante da importância de maximizar a VUR para garantir a cobertura de uma região observada, é apresentado nessa dissertação um protocolo de programação sono-vigília adaptado à transferência sem fio de energia (TSFE) nas RSSFS para a maximização da vida útil destas redes. A pesquisa apresentada nessa dissertação foi realizada através de simulações com o simulador Omnet, utilizando o framework Castália. O resultado alcançado permite estabelecer que o protocolo proposto, chamado SWPTMAC, oferece um ganho médio de cerca de 24% em relação ao estado da arte.

Palavras-chave: Redes de Sensores Sem Fio Subterrânea, Internet das Coisas, Algoritmos de Sono-Vigília

Abstract

Wireless Underground Sensor Networks (WUSN) are networks made up of sensors that are buried and use a wireless connection to communicate. WUSNs are fundamental for monitoring underground environments, but face the problem of having a limited Network Lifetime (NL). The NL is defined as the maximum time for which the network is operational, that is, the time it is possible to obtain network coverage in an observed region. Therefore, given the importance of maximizing the NL to ensure coverage of an observed region, this dissertation presents a sleep-wakeup protocol adapted to wireless power transfer (WPT) in WUSNs to maximize the lifetime of these networks. The evaluation of the proposal that will be presented in this dissertation was carried out through simulations with the Castália simulator. The result achieved of the proposed SWPTMAC protocol presented 24% NL gain in average when compared to state-of-the-art protocols.

Keywords: Wireless Underground Sensor Networks, Internet of Things, Wake-Sleep Algorithm

Sumário

1	Introdução	1
2	Fundamentação Teórica	4
2.1	Internet das Coisas	4
2.2	IoT na Agricultura de Precisão	5
2.3	Redes de Sensores	6
2.3.1	Tipos de Comunicação das RSSF	6
2.3.2	Consumo de Energia das RSSF	7
2.3.3	Tempo de Vida das RSSF	7
2.4	Redes de Sensores Sem Fio Subterrâneas	11
2.5	Aplicações de RSSFS	12
2.5.1	Tubulações	12
2.5.2	Infraestrutura	12
2.5.3	Agricultura	12
2.6	Desafios das RSSFS	13
2.6.1	Ambientes Adversos das RSSFS	14
2.6.2	Posicionamento dos Dispositivos nas RSSFS	14
2.6.3	Degradação da Comunicação em RSSFS	14
2.6.4	Limitação Energética nas RSSFS	15
2.7	VUR das RSSFS	17
3	Trabalhos Relacionados	18
4	Solução proposta	26
4.1	SWPTMAC: Protocolo MAC de Programação Sono-Vigília e Transferência de Energia	26
4.2	Cenários	28
4.3	Ferramentas	29
4.3.1	OMNeT++	29
4.3.2	Castália	30

5	Protótipo	32
5.1	Códigos usados igualmente por RIMAC, EERIMAC e SWPTMAC	32
5.2	Códigos usados pelo RIMAC	35
5.3	Códigos usados pelo EERIMAC	40
5.4	Códigos usados pelo SWPTMAC	44
5.5	Códigos da implementação do drone	50
5.6	Códigos da implementação do pivô central	51
6	Simulação e Resultados	53
6.1	Métricas de Desempenho	54
6.2	Simulação 1	55
6.3	Simulação 2	57
6.4	Simulação 3	59
6.5	Simulação 4	60
6.6	Simulação 5	62
7	Considerações Finais	64
	Referências	66
	Apêndice	72
A	Fichamento de Artigo Científico	73

Lista de Figuras

2.1	Máquinas de estados demonstrando o protocolo RIMAC.	9
2.2	Máquinas de estados demonstrando o protocolo EERIMAC.	10
2.3	TSFE em RSSFS. Fonte: [1].	11
4.1	Representação da solução.	27
4.2	Máquinas de estados demonstrando o protocolo SWPTMAC.	28
4.3	Cenário Proposto.	29
4.4	Estrutura Básica do Castália.	30
4.5	Módulos de um Nó no Castália.	30
6.1	Avaliação do Impacto da Quantidade de Pacotes Enviados de Um Dispositivo para Outro em uma RSSFS.	56
6.2	Avaliação do Impacto do Tamanho dos Pacotes Enviados de Um Dispositivo para Outro em uma RSSFS.	58
6.3	Avaliação do Impacto da Quantidade de Sensores em uma RSSFS.	60
6.4	Avaliação do Impacto da Quantidade de Drones em uma RSSFS Usando o Protocolo SWPTMAC.	61
6.5	Comparação da Estimativa de VUR Alcançada com Cada Protocolo.	63

Lista de Tabelas

2.1	Tabela comparando os trabalhos que tratam da questão energética em RSSFS.	16
3.1	Trabalhos relacionados.	19
6.1	Especificações da Máquina.	53
6.2	Pilha de Comunicação das Simulações 1 ao 3.	54
6.3	Pilha de Comunicação da Simulação 4.	54
6.4	Pilha de Comunicação da Simulação 5.	54
6.5	Parâmetros da Simulação 1.	56
6.6	Parâmetros da Simulação 2.	57
6.7	Parâmetros da Simulação 3.	59
6.8	Parâmetros da Simulação 4.	61
6.9	Parâmetros da Simulação 5.	62

Capítulo 1

Introdução

As Redes de Sensores Sem Fio (RSSF) são redes utilizadas para realizar monitoramento, as quais são baseadas na interconexão de um grande número de nós sensores aptos a realizarem sensoriamento e processamento de dados [2]. Há um tipo de RSSF que é capaz de realizar o monitoramento de ambientes subterrâneos: as Redes de Sensores Sem Fio Subterrâneos (RSSFS), mais conhecidas pela expressão em inglês *Wireless Underground Sensor Networks* (WUSN) [3][4]. O monitoramento de ambientes subterrâneos é importante para setores que trabalham com tubulações, infraestrutura e agricultura, uma vez que obtém informações úteis sobre água, conteúdo mineral do solo, integridade das infraestruturas, deslizamentos de terra e ocorrência de terremotos [5][6][7]. As RSSFS são redes formadas por sensores que ficam completamente enterrados e utilizam comunicação sem fio para a transferência de informação. O fato dos sensores enterrados não utilizarem fios simplifica a sua implantação, no contexto do monitoramento subterrâneo, quando comparada com soluções tradicionais [8]. Assim como as demais redes de sensores, os dispositivos das RSSFS tem recursos energéticos escassos, pois dependem de baterias para operar. Mas, diferentemente das redes de sensores tradicionais, as RSSFS tem o acesso aos dispositivos dificultada, por estarem enterrados no solo. Dessa forma, seus recursos energéticos são ainda mais custosos para serem renovados e mantidos.

O uso com parcimônia dos recursos energéticos providos por baterias, para a alimentação dos sensores que formam uma RSSF, é um problema estudado na literatura e desafiador [9]. No contexto de RSSFS, a localização física dos nós dificulta e, muitas vezes, impossibilita o seu acesso e substituição [2]. Naturalmente, prolongar ao máximo o tempo de vida útil da rede (VUR) torna-se fundamental para a definição do tempo de disponibilidade máxima operacional, bem como para se obter as regiões de efetiva cobertura da rede ao longo de sua operação [10].

Nesse caso, a TSFE é explorada para aumentar a VUR dos dispositivos de uma RSSFS [11], [12],[13]. Inclusive, para melhorar a transferência de energia, pode-se utilizar

esquemas de adaptação de bobina para carregamento do sensor enterrado baseado nas características do cenário, como profundidade do nó e penetração da comunicação no solo para aumentar a VUR do dispositivo[14]. Além disso, outros fatores também podem ser levados em consideração, como a alocação otimizada de recursos considerando o QoS das aplicações [15] e [16]. Além do carregamento dos dispositivos, outras técnicas podem ser exploradas para aumentar a VUR dos dispositivos, como estratégias para minorar o uso dos recursos energéticos com a comunicação.

Dentre os fatores que mais influenciam no consumo energético de uma rede RSSFS, está o uso de comunicação via rádio. Os estados distintos: transmitindo, recebendo e dormindo possuem taxas diferentes de consumo energético. Protocolos de comunicação como o apresentado em [17] exploram essas fases distintas de consumo e trabalham com a abordagem conhecida como programação sono-vigília (*Sleep-wakeup scheduling*). Nesta abordagem, o objetivo é permanecer com o rádio em estado adormecido pelo maior período de tempo possível, acordando-o somente nos momentos de recebimento e transmissão de informação. O desafio nesta abordagem é manter o sincronismo lógico de conexão entre os sensores, de forma a manter a área de cobertura de comunicação operacional e permitir o escoamento da informação através da infraestrutura de comunicação. Entretanto, os protocolos existentes de sono-vigília para redes de sensores tradicionais, não acomodam um estado para realizar o carregamento do dispositivo. Dessa forma, no melhor dos nossos conhecimentos, até o momento nenhum protocolo foi proposto para explorar ambos, uma boa técnica de programação sono-vigília que ao mesmo tempo otimize o uso da transferência de energia, objetivando a maximização da VUR.

A solução apresentada nessa dissertação é de um protocolo que utilize a técnica de programação sono-vigília em conjunto com a TSFE para maximização da VUR [18]. A solução visa a agricultura de precisão como área de aplicação, a qual é importante para auxiliar na alimentação da crescente população mundial através de uma produção agrícola em maior quantidade e com melhor qualidade [19]. A avaliação da solução foi feita no simulador Omnet em conjunto com o framework Castália. Como a VUR de uma RSSFS é limitada pela fonte de energia independente de cada dispositivo [8], a métrica utilizada para analisar os protocolos de programação sono-vigília será a estimativa de VUR, que é calculada no simulador dividindo a carga de energia remanescente dos dispositivos pelo consumo médio destes. Além disso, restrições quanto a taxa de transferência e perda de pacotes serão consideradas na avaliação. As principais contribuições apresentadas nessa dissertação são resumidas a seguir:

- Arquitetura de RSSFS com protocolos otimizados para maximização da VUR e aplicada a agricultura;

- Desenvolvimento de um novo protocolo de programação de sono-vigília em redes que recebem transferência de energia chamado SWPTMAC (Sleep-Wakeup Power-Transfer Media Access Control);
- Comparação com o estado da arte de protocolos típicos de sono-vigília, como o RIMAC e EERIMAC, para maximização da VUR;

O restante desse trabalho foi organizado da seguinte forma. O Capítulo 1 traz a fundamentação teórica. O Capítulo 2 traz os trabalhos relacionados. O Capítulo 4 apresenta a solução proposta. O Capítulo 5 apresenta os códigos desenvolvidos na implementação da solução. O Capítulo 6 mostra as simulações realizadas e os resultados que foram alcançados. Na sequência, é apresentada as considerações finais do trabalho. Ao final de tudo, é apresentado um fichamento do artigo baseado na pesquisa apresentada nesta dissertação que foi aceito na conferência NOMS 2024.

Capítulo 2

Fundamentação Teórica

2.1 Internet das Coisas

A Internet das Coisas, do inglês Internet of Things (IoT), é a ideia de conectar bilhões ou trilhões de objetos à Internet. A IoT transforma objetos tradicionais em dispositivos inteligentes capazes de obter e compartilhar informações, atuar e tomar decisões. As áreas que a IoT atua são as seguintes: transporte, saúde, indústria, comércio, escola, veículos, casas inteligentes e agricultura. O desenvolvimento da IoT é importante porque pode proporcionar uma melhor qualidade de vida para as pessoas e um crescimento para a economia [20].

Segundo [20], os serviços de IoT podem ser organizados em quatro categorias: (i) identificação, (ii) agregação de informações, (iii) situação colaborativa e (iv) onipresença. Nos serviços de identidade, cada aplicativo precisa trazer os objetos do mundo físico para o mundo virtual identificando esses objetos. Nos serviços de agregação de informações, medições sensoriais brutas são coletadas e processadas para os aplicativos de IoT. Já nos serviços de situação colaborativa informações são agregadas e geridas para tomar decisões e reagir de acordo com essas decisões. Por fim, há os serviços do nível de onipresença, que estão disponíveis para quem precise deles em qualquer lugar, que é o objetivo final de toda aplicação de IoT. A solução apresentada nesse trabalho, irá focar principalmente na agregação de informações.

Ainda de acordo com [20], a utilização de IoT vem crescendo e, proporcional a esse crescimento, devem ser exploradas tecnologias, inovações e aplicações para atenderem a demanda do mercado e as necessidades dos consumidores. Para atender a demanda do mercado e a necessidade dos consumidores é necessário que os dispositivos de IoT tenham disponibilidade em todos os lugares e em todos os momentos. Para obter essa disponibilidade dos dispositivos de IoT são requeridos protocolos, arquiteturas e padronizações que suportem a heterogeneidade desses objetos inteligentes. A solução apresentada nessa

dissertação busca combinar soluções tecnológicas com a finalidade de contribuir com a área da IoT.

2.2 IoT na Agricultura de Precisão

Uma das possíveis áreas de aplicação de IoT é na agricultura. A IoT pode contribuir para prever, processar, analisar e melhorar as atividades agrícolas em tempo real. O monitoramento em tempo real das fazendas é importante por ser parte fundamental da chamada agricultura de precisão. A melhoria de processos agrícolas e a otimização da utilização da água, do solo, dos minerais e de outros recursos utilizando a tecnologia é o que forma a agricultura de precisão [19].

A agricultura de precisão é importante porque, segundo [19], o aumento populacional no mundo fez com que surgisse a necessidade de uma produção agrícola em maior quantidade e com melhor qualidade. A previsão é que a população mundial chegue em 8 bilhões em 2025 e em 9,6 bilhões até 2050. Como a agricultura de precisão busca a eficiência e a otimização de recursos ela pode contribuir para atender a demanda crescente por alimentos. A melhora na qualidade da produção que a agricultura de precisão pode oferecer também é fundamental pois pode ajudar a diminuir a emissão de gases que contribuem para o efeito estufa e a agricultura de precisão é o segundo maior responsável pela emissão desses gases.

Segundo [19], a IoT contribui com a agricultura de precisão nos seguintes aspectos: (i) gerenciamento da fazenda; (ii) irrigação; (iii) monitoramento de colheita; (iv) detecção de pragas. Primeiro, o sensoriamento através de IoT contribui para a execução de previsões que facilitam o gerenciamento de fazendas. Segundo, o monitoramento em tempo real através de IoT pode contribuir para diminuir o consumo energético, o trabalho e o custo da irrigação de fazendas. Em terceiro lugar, o monitoramento de colheita através de IoT pode aumentar a produtividade e reduzir custos. Por fim, a IoT pode contribuir na detecção de pragas, o que evita crises econômicas e de alimentos nos países.

Ainda de acordo com [19], os desafios da implantação de IoT na agricultura são: custo, padronização, heterogeneidade, acessibilidade, adaptabilidade, otimização energética, compatibilidade, confiabilidade, mobilidade, condições ambientais e implantações em tempo real. A questão mais emergente em sistemas de IoT é a questão de otimização energética. As seguintes soluções convencionais e não convencionais vem sendo testadas para solucionar a questão da otimização energética na IoT: energia solar, eólica, hídrica e sistemas de transferência de energia. A solução apresentada nessa dissertação tem como objetivo atuar na otimização energética de IoT na agricultura de precisão.

2.3 Redes de Sensores

As Redes de Sensores Sem Fio (RSSF), ou WSN do inglês Wireless Sensor Networks, são redes estabelecidas pela IoT e formadas por sensores autônomos e distribuídos. A comunicação desses sensores é feita sem a utilização de fios e é feita máquina a máquina sem a interferência de humanos. As RSSF geralmente passam dados cooperativamente através da rede para um local centralizado. Esses sensores são utilizados para monitorar condições físicas ou ambientais como temperatura, som e pressão [21][2].

As RSSF possuem várias possíveis aplicações de monitoramento em diversas áreas. As áreas de aplicações dessas redes são: militar, engenharia civil, monitoramento ambiental, logística, rastreamento de animais, transporte, robótica, automação industrial, saúde, entretenimento, escritórios inteligentes, esportes, automação de construção e controle doméstico. Além disso, as RSSF podem ser categorizadas de acordo com as suas características. Os parâmetros que definem como uma RSSF deve ser categorizada são: serviço, comunicação e tráfego, componente de serviço, rede, nó e ambiente [22].

Apesar das muitas aplicações e da importância das RSSF, essas redes ainda possuem problemas a serem enfrentados como a redução da taxa de perda de pacotes, a diminuição latência, o aumento da cobertura da rede e o fim do esgotamento energético. A solução desses problemas é fundamental para melhorar o desempenho da rede [23]. A proposta apresentada nessa dissertação tem como objetivo tratar o problema de esgotamento energético em um tipo específico de RSSF, as Redes de Sensores Sem Fio Subterrâneas (RSSFS).

2.3.1 Tipos de Comunicação das RSSF

Há três tipos de padrões de comunicação das RSSF que são *broadcast* (difusão), *converge cast* (transmissão convergente) e *local gossip* (fofocas locais) [24]. No tipo *broadcast*, que é usado para enviar quaisquer dados para todos os nós sensores da rede. Nas comunicações do tipo *local gossip* os sensores comunicam localmente uns com os outros, ou seja, quando um sensor envia uma mensagem dentro da mesma região para os nós vizinhos. Já no caso do *converge cast*, onde um grupo de sensores interage com um único sensor, que é o centro de informações para onde os sensores enviam os dados coletados. O nó de destino pode ser o chefe de um cluster - um centro ou estação base para fusão de dados. Na solução apresentada, será utilizado o tipo de comunicação *converge cast* e todos os sensores enviarão os dados para um único dispositivo coletor.

2.3.2 Consumo de Energia das RSSF

O consumo de energia das RSSF pode ser dividido em três domínios: sensoriamento, processamento de dados e comunicação [25]. Em primeiro lugar, o consumo de energia com sensoriamento varia de acordo com a aplicação, o sensor utilizado, a frequência dos eventos e a complexidade das detecções. Já o processamento de dados é feito pelas habilidades computacionais obtidas através de microprocessadores e o consumo de energia com processamento dos dados é semelhante ao sensoriamento. Por fim, a comunicação é feita pelos circuitos do transceptor durante a transmissão e recepção dos dados. A comunicação é o domínio que realiza o maior consumo de energia e por isso é o campo de foco da pesquisa apresentada nessa dissertação.

2.3.3 Tempo de Vida das RSSF

A vida útil da rede (VUR), ou em inglês Network Lifetime (NL) é o tempo máximo para o qual a rede está operacional [25]. De acordo com [18], um dos desafios das RSSF é a maximização da VUR. Há quatro categorias de maximização do VUR: baseado no tempo de vida do nó, que prolonga a longevidade dos sensores; baseado em cobertura e conectividade, que fornece cobertura para uma área-alvo específica; baseado em transmissão: baseado na entrega de informações antes da falha; parametrizado: utiliza parâmetros como probabilidade de disponibilidade do nó e probabilidade de conectividade. Afim de maximizar o VUR algumas das técnicas existentes são:

- Alocação otimizada de recursos (*Resource allocation*): algoritmos que otimizam várias restrições como confiabilidade de transmissão, roteamento, controle de energia, implantação ótima de nós e maximização da taxa de transferência;
- Esquemas de transmissão oportunista (*Opportunistic transmission schemes*): quando a informação é coletada pelos sensores, sua transmissão para o sorvedouro deve considerar diferentes restrições, incluindo informações de estado do canal (channel state information - CSI), programação de transmissão, informações de energia residual (residual energy information - REI), atraso fim-a-fim, rendimento e alocação de energia;
- Programação sono-vigília (*Sleep-wake scheduling*): como o rádio é o componente que mais consome energia no nó sensor, o objetivo das abordagens de agendamento sleep-wake é colocar o rádio no modo adormecimento pelo maior tempo possível e acordá-lo apenas quando necessário com garantia de eficiência na entrega de pacotes;
- Roteamento e agrupamento (*Routing and clustering*): algoritmos que auxiliem na tomada de decisões de roteamento e agrupamentos;

- Conectividade de cobertura e implantação ótima (*Coverage connectivity and optimal deployment*): cobertura e a conectividade são as restrições mais importantes de uma RSSF e uma implantação de sensores otimizada que fornece cobertura total confiável de um campo de detecção específico e retransmite todos os dados detectados para o coletor pode estender o VUR;
- Coleta e agregação de dados (*Data gathering and aggregation*): dados de vários sensores são agregados evitando a transmissão redundante de dados;
- Correlação de dados (*Data correlation*): correlação espaço-temporal dos dados de sensores vizinhos levam a redução de transmissões;
- Formação de feixe (*Beamforming*): evita que um sensor transmita informações com sua potência máxima;
- Colheita de energia (*Energy harvesting*): mecanismos que permitem fornecer uma fonte de alimentação para algum sensor. A primeira abordagem de colheita de energia possível converte diretamente a energia ambiente em energia elétrica para alimentar o sensor. A segunda abordagem é baseada no armazenamento de energia elétrica antes de alimentar energeticamente o sensor. Com a técnica de colheita de energia a RSSF pode chegar a energia ilimitada e, portanto, potencialmente pode ter a VUR ilimitada [26];
- Transferência Sem Fio de Energia (*Wireless Power Transfer*): é fornecida uma fonte de alimentação sem fio estável e controlável para dispositivos que consomem muita energia, é uma área relacionada ao *Energy harvesting* [27].

Programação Sono-vigília

São exemplos de protocolos que implementam a programação sono-vigília o RIMAC (*A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol*) [17] e as suas evoluções como o EERIMAC (*A Energy-Efficient Receiver-Initiated Asynchronous Duty Cycle MAC Protocol*) [28] e o COSFI-RIMAC (*A Cooperative Short Frame Identifier Receiver Initiated MAC Protocol*) [29]. O RIMAC é um protocolo da camada de enlace que utiliza um mecanismo de ciclo de trabalho assíncrono para resolver o problema da escuta ociosa que é uma das fontes mais significativas de consumo de energia em nós sensores sem fio. O mecanismo de ciclo de trabalho consiste em ligar o rádio do dispositivo apenas periodicamente, economizando assim o consumo energético. O funcionamento do protocolo RIMAC é demonstrado na Figura 2.1.

O protocolo RIMAC pode ser desenvolvido nos moldes de uma máquina de estados finita (MEF). A Figura 2.1(a) demonstra uma MEF que representa o comportamento

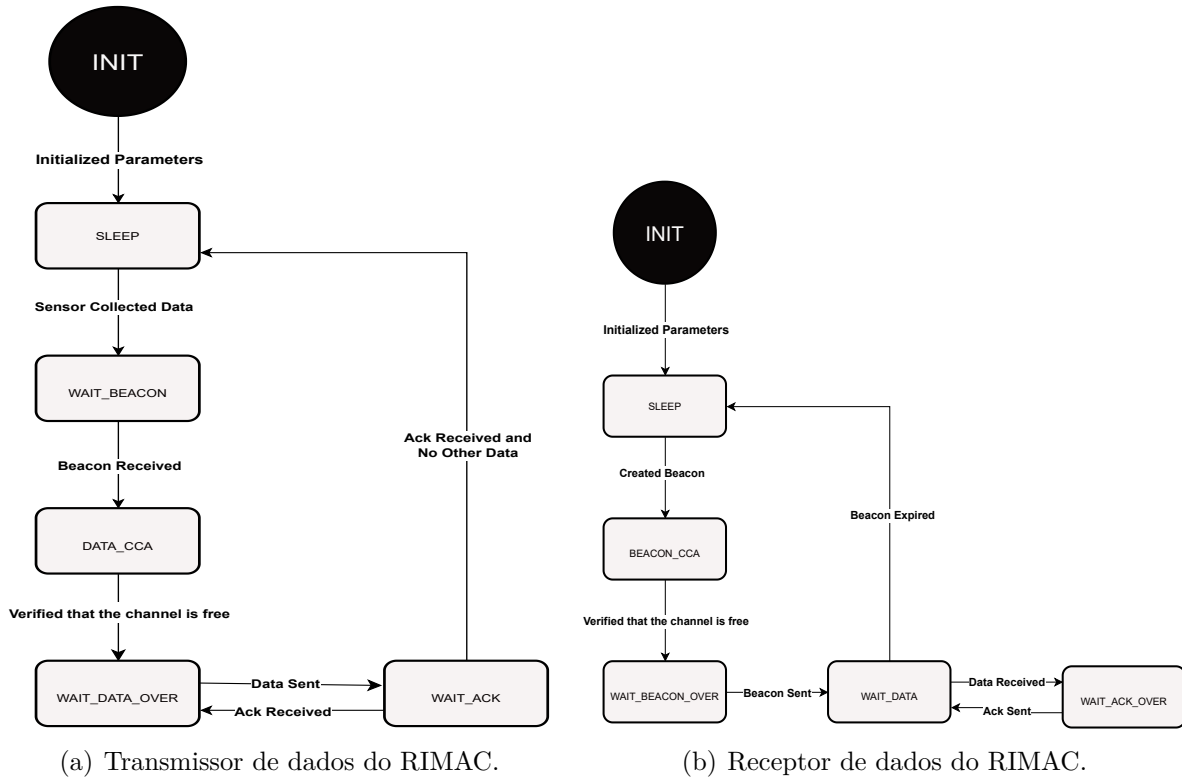


Figura 2.1: Máquinas de estados demonstrando o protocolo RIMAC.

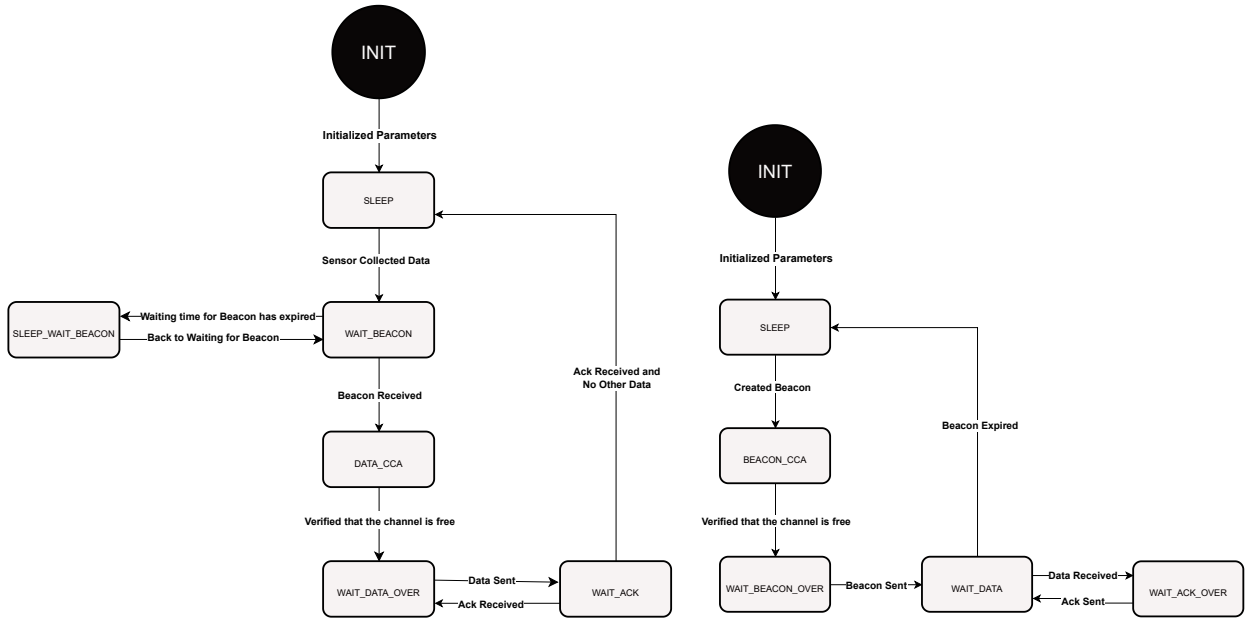
do dispositivo que envia as informações no protocolo RIMAC. O rádio do dispositivo fica no modo *sleep* no estado SLEEP (adormecido), no modo TX (transmitindo) no estado WAIT_DATA_OVER e no modo RX (recebendo) nos estados WAIT_BEACON, DATA_CCA E WAIT_ACK. O receptor das informações é representado na Figura 2.1(b). O rádio do dispositivo fica no modo *sleep* no estado SLEEP (adormecido), no modo TX (transmitindo) nos estados WAIT_BEACON_OVER e WAIT_ACK_OVER e no modo RX (recebendo) nos estados WAIT_DATA e BEACON_CCA. As setas representam as mudanças de estados e o texto em cada seta explica o porquê de cada mudança.

São pontos fundamentais da implementação do RIMAC: Beacon Frames, Dwell Time for Queued Packets, DATA Frame Transmissions From Contending Senders, Collision Detection e Retransmissions. Beacon Frames são pacotes enviados para sinalizar que um dispositivo está disponível para receber dados. Dwell Time é o tempo esperado após enviar um ack para que o dispositivo que recebeu o ack possa enviar mais algum pacote de dados.

Outro exemplo de protocolo que implementa a técnica de sono-vigília é o EERIMAC [28], que é baseado no RIMAC e foi usado como base para o desenvolvimento do SWPT-MAC. O que o EERIMAC faz de diferente do RIMAC é que, caso o dispositivo transmissor fique muito tempo esperando o BEACON (no estado WAIT_BEACON), ele entra no modo *sleep* (no estado SLEEP_WAIT_BEACON) para economizar energia. A Figura 2.2

traz uma representação da MEF do protocolo EERIMAC, o transmissor de informações sendo representado na Figura 2.2(a) e o receptor pela Figura 2.2(b)

São pontos que diferenciam o EERIMAC do RIMAC: Wakeup Period e Sleep Period.



(a) Transmissor de dados do EERIMAC.

(b) Receptor de dados do EERIMAC.

Figura 2.2: Máquinas de estados demonstrando o protocolo EERIMAC.

Transferência Sem Fio de Energia (TSFE)

Devido a capacidade limitada da bateria e a necessidade de recarga frequente nas RSSF, é preciso encontrar novas maneiras de fornecimento de energia sem o incômodo de conectar cabos [27]. Uma das maneiras é a colheita de energia do ambiente *energy harvesting* e outra maneira é a transferência sem fio de energia (TSFE) (*Wireless Power Transfer - WPT*). A colheita de energia busca eliminar o desperdício energia do meio ambiente utilizando fontes renováveis com energia solar, iluminação, calor, som, movimento, vibração, vento e radiação eletromagnética. No entanto, a energia colhida é considerada insuficiente para atender a computação intensa e por isso é utilizada a WPT que fornece uma estável e controlável fonte de alimentação sem fio implantando um foco de energia dedicado.

As RSSFS são de um tipo de rede que possibilita a utilização da TSFE. Em [1], é apresentada uma utilização de TSFE em RSSFS através de indução magnética acoplada e ressonante. A indução magnética é utilizada para fazer a transferência de energia sem utilização de fios, já que os campos magnéticos são muito menos afetados pela natureza imprevisível e mutável do subsolo do que outras técnicas. A quantidade de energia que pode ser obtida com a utilização de TSFE em RSSFS é demonstrada na Figura 2.3.

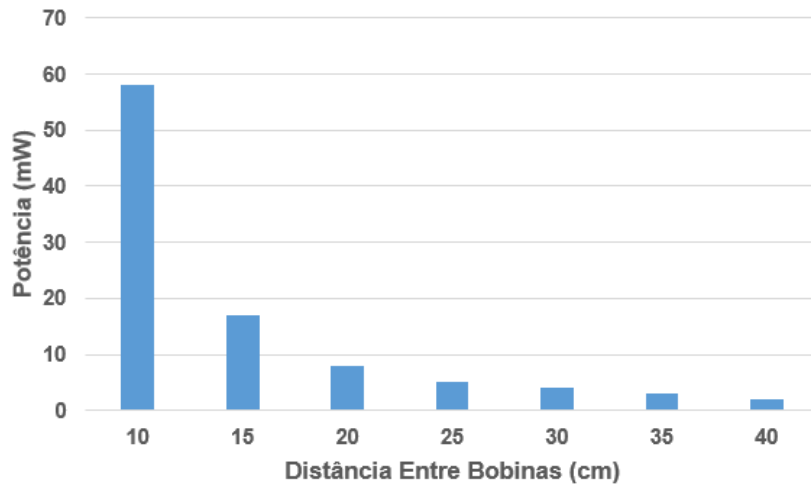


Figura 2.3: TSFE em RSSFS. Fonte: [1].

2.4 Redes de Sensores Sem Fio Subterrâneas

As RSSFS são uma extensão importante das redes de sensores sem fio terrestres, onde os nós sensores são enterrados no subsolo e se comunicam sem fio através do solo [3][4]. Nessas redes, além dos sensores enterrados, existem os coletores de dados que ficam acima do solo que recebem os dados capturados pelos sensores no subsolo. Esses coletores podem ser fixos ou móveis, transportados por pessoas ou máquinas. O trabalho proposto nessa dissertação aborda RSSFS com coletores de dados móveis transportados por máquinas agrícolas.

As RSSFS são utilizadas no monitoramento subterrâneo. Os sensores utilizados nas RSSFS coletam informações sobre a água e conteúdos minerais do solo, sobre a integridade das infraestruturas, sobre deslizamentos de terra e ocorrências de terremotos [8]. As áreas de aplicação das RSSFS são: tubulações subterrâneas, infraestruturas subterrâneas e na agricultura [5][6][7]. Cada uma dessas aplicações serão explicadas com mais detalhes na Seção 2.5. O trabalho proposto nessa dissertação usa com área de aplicação a agricultura.

As RSSFS são fundamentais para o monitoramento subterrâneo. No entanto, estas redes possuem problemas de comunicação, destacando-se quatro deles: (i) adaptação a ambientes instáveis; (ii) posicionamento ideal dos dispositivos; (iii) degradação da qualidade da comunicação; (iv) recursos energéticos restritos [8]. Esses problemas são abordados com mais detalhes na Seção 2.6. O trabalho proposto nessa dissertação tem como objetivo tratar o problema da restrição dos recursos energéticos nas RSSFS.

2.5 Aplicações de RSSFS

2.5.1 Tubulações

Um exemplo de utilização das RSSFS apresentado no monitoramento de tubulações. Um trabalho que demonstra essa utilização foi apresentado em [5], onde um repetidor é instalado no fundo de um poço. Os dados coletados são enviados via repetidores para gateways ou estações que ficam localizadas na superfície. Existem dois canais de comunicação nesse sistema. O primeiro é entre o sensor e o repetidor e o segundo é entre o repetidor e o gateway. Foi levado em consideração as métricas de indicação de intensidade do sinal recebido (RSSI), relação sinal-ruído (SNR) e relação de entrega de pacote (PDR). Embora traga uma ideia de aplicação na área de tubulações, o foco principal do trabalho foi a análise da comunicação entre o repetidor e o gateway.

2.5.2 Infraestrutura

O trabalho apresentado em [6] demonstra uma implementação de RSSFS no monitoramento da infraestrutura de estádios. A abordagem demonstrada volta-se para o posicionamento desses sensores. A proposta é de uma arquitetura que utiliza um método de estimativa de distância para sinais de som e vibração. Foi feita uma simulação em NS2 para avaliar a proposta.

2.5.3 Agricultura

Outra área de aplicação das RSSFS é a agricultura. Um exemplo de implementação de RSSFS foi demonstrado em [7], onde apresentada uma arquitetura de utilização de sensores subterrâneos sem fio para acionamento de pivôs na agricultura de precisão autônoma. Foi apresentado que é importante a utilização desse tipo de sensor porque ele pode oferecer informações mais precisas sobre o solo se comparado com o sensoriamento aéreo ou via satélite. Outro ponto é que esses sensores são instalados de maneira profunda o suficiente, de maneira que não interfira na operação das máquinas agrícolas, o que evita que eles tenham de ser retirados e replantados com frequência. Nessa abordagem, o pivô também é utilizado como estrutura física dos sensores que estão acima do solo para a comunicação com os sensores enterrados.

Um outro exemplo de utilização de RSSFS na agricultura foi apresentado em [30]. Um modelo matemático foi proposto onde é apresentada uma ideia de utilização de sensor acústico para detecção de pragas de insetos. Nessa dissertação, são analisados a intensidade do sinal recebido e a perda do caminho. A conclusão alcançada é a necessidade de utilização de transmissão de sinal com diferentes potências de transmissão.

A previsão da perda de caminho devido à atenuação do sinal no subsolo é um ativo importante para o bom funcionamento da RSSFS na agricultura de precisão. Em [31], é proposto um modelo baseado em uma previsão precisa da Constante Dielétrica Complexa (CDC). A atenuação das ondas refletiva e refrativa de acordo com a profundidade de enterramento do nó sensor são levadas em consideração em três tipos de comunicação (subterrâneo-subterrâneo, subterrâneo-aéreo e aéreo-subterrâneo). O modelo foi validado em medições feitas em um jardim botânico.

As telhas de drenagem, sistema utilizado na agricultura que remove o excesso de águas subterrâneas, são utilizadas em um experimento como guias de ondas para transportar energia eletromagnética (EM) por longas distâncias [32]. As ondas propagadas são convertidas em energia que alimenta os sensores. Nos experimentos, são avaliadas as telhas totalmente ou parcialmente cobertas, bem como telhas na presença de água. A técnica proposta é apontada como uma solução para o monitoramento de líquidos e gases ao longo de quilômetros de distância.

Uma arquitetura de utilização de RSSFS na agricultura é apresentada em [33]. Nesse trabalho, também é apresentado um projeto de antena que utiliza informações de conteúdo de água no solo e mecanismos de direcionamento de feixe para melhorar a eficiência e o alcance das comunicações subterrâneas sem fio. A refração da interface solo/ar também é levada em consideração no projeto. Ainda são levadas em consideração as variações nos efeitos adversos que o canal de rádio fica exposto no ambiente subterrâneo.

Outra aplicação das RSSFS na agricultura, é no monitoramento da fertilidade do solo. Além da importância dessas informações de fertilidade para o aumento da produção agrícola, essas informações são importantes para diminuir a emissão de gases do efeito estufa e proteger o meio ambiente. Em [34], é feita uma revisão de como esse monitoramento de fertilidade tem sido feito na agricultura. A taxonomia que é avaliada no artigo inclui diferentes tipos de sensores de solo subterrâneo com tecnologias de comunicação e técnicas de *Machine Learning (ML)* para monitoramento da fertilidade do solo. Segundo o artigo, o objetivo do trabalho apresentado é que as soluções de monitoramento da fertilidade do solo propostas sejam acessíveis e sustentáveis para que todos os agricultores do mundo possam aproveitar a alavanca que não apenas melhorará seu investimento, mas também protegerá o clima através da aplicação precisa de fertilizantes.

2.6 Desafios das RSSFS

A utilização das RSSFS é dificultada por causa das condições ambientais adversas. As RSSFS precisam de sensores com baixo consumo energético e tamanho pequeno, comunicação e soluções de rede eficientes, bem como técnicas de localização precisas. Vários

trabalhos vêm sendo desenvolvidos para superar tais desafios e alguns desses trabalhos são apresentados nessa seção. O desafio de lidar com ambientes adversos em RSSFS é apresentado na Subseção 2.6.1, a localização dos dispositivos na Subseção 2.6.2, a degradação da comunicação é apresentada na Subseção 2.6.3 e a restrição energética em RSSFS é apresentado na Subseção 2.6.4.

2.6.1 Ambientes Adversos das RSSFS

Uma das dificuldades enfrentadas pelas RSSFS é a mudança das condições ambientais. A composição, a eletricidade e a umidade do solo não são constantes. Diante disso, surge a necessidade de que os dispositivos adaptem-se às condições do momento sem a necessidade de serem desenterrados para realização de adaptações. Afim de resolver o problema de adaptação a diferentes características do solo em [35] é apresentado um mecanismo para seleção adaptativa dos parâmetros da camada física baseado na umidade do solo em cada momento.

2.6.2 Posicionamento dos Dispositivos nas RSSFS

É importante que as informações extraídas de sensores tenham grande precisão espaço temporal, surge então a necessidade de uma identificação precisa da localização dos sensores. Tal identificação da localização do nó sensor foi tratada em [36]. O contexto abordado no trabalho citado foi de uma rede híbrida formada por sensores acima e abaixo do solo. Foi feita a simulação de um modelo de localização baseado na intensidade do sinal recebido (Received Signal Strength - RSS).

2.6.3 Degradação da Comunicação em RSSFS

Outro problema enfrentado nesse tipo de cenário é quanto ao sinal na comunicação, já que ocorre uma absorção e um espalhamento do sinal no solo. Uma análise do impacto da textura e da umidade do solo na transmissão do sinal é feita em [37]. Em [38], foi analisada a intensidade do sinal recebido e os parâmetros de perda de caminho e verificado o quanto a transmissão é prejudicada à medida que aumenta a profundidade dos dispositivos. Um modelo de previsão da perda de caminho devido à atenuação do sinal subterrâneo é feita em [39] utilizando um modelo baseado na Constante Dielétrica Complexa (CDC). Várias são as soluções que vêm sendo pesquisadas para o problema de comunicação, por exemplo, [40] apresenta como solução a utilização de comunicação acústica em até 50 metros a taxas de dados de 20 bps. Em [41], é estudada a transmissão via LoRa em conjunto com o protocolo LoRaWAN em solos com cascalho, areia ou argila.

2.6.4 Limitação Energética nas RSSFS

O canal subterrâneo utilizado nas RSSFS exige uma maior potência na transmissão dos rádios, mas com as limitações energéticas dos dispositivos redes acabam surgindo os problemas de comunicação dessas redes. O primeiro motivo para a escassez de recursos energéticos nas RSSFS é porque as fontes de energia interna aos dispositivos subterrâneos podem não ter uma grande duração devido ao esforço para envio das informações para a superfície. O segundo motivo da limitação energética das RSSFS é porque os sensores ficam enterrados e muitas vezes é difícil o desenterramento dos sensores para troca ou recarregamento de baterias [8]. Existem várias pesquisas que buscam solucionar esse problema de limitação energética nas RSSFS, a Tabela 2.1 traz uma comparação das métricas utilizadas nos trabalhos que tratam desse tema.

Para lidar com o problema de limitação energética nas RSSFS tem sido utilizada a indução magnética, que serve tanto para a transferência de energia como para a comunicação [42]. O estudo da utilização de indução magnética em WUSN investiga a modelagem do canal, estimativa do canal, desempenho de link único, estratégia de implantação, carga de bateria, soluções de camadas cruzadas de rede, localização e validação. No artigo citado, são apontadas algumas direções de trabalhos futuros. Uma das direções é a otimização da camada cruzada para atender os objetivos de melhorar a taxa de transferência da rede, tornar o carregamento mais eficiente e melhorar a acurácia da localização. Outra direção é tratar das informações de estado do canal imperfeito. Outras direções são a combinação de relés ativos e passivos, a utilização de múltiplas bobinas e integração com outras redes de sensores sem fio.

Tratando-se de indução magnética nesse tipo de cenário o carregamento foi abordado em [11] e [12]. Nos trabalhos citados foi apresentada uma rede de relés magnéticos virtuais baseados em múltiplas entradas e múltiplas saídas, (multiple-input and multiple-output - MIMO). Além disso, um protocolo de transmissões de sinais simultâneas de vários nós sensores foi desenvolvido no trabalho para reduzir as perdas de energia. Como trabalho futuro, foi sugerida a melhoria dessa técnica, otimizando os parâmetros do sistema como parte da otimização da camada cruzada mencionada, de modo que a eficiência energética aumente.

Outro exemplo da utilização da indução magnética pode ser visto em [14]. No artigo citado, foi apresentado um trabalho onde foi desenvolvido um modelo que identifica a melhor faixa de frequência de operação. Além disso, foi desenvolvido um esquema de adaptação da bobina que torna o projeto eficiente em termos de energia e largura de banda. Foram feitos experimentos validando o modelo e garantindo que ele pode operar mesmo diante das variações típicas das propriedades do solo, como em caso de chuva.

Tabela 2.1: Tabela comparando os trabalhos que tratam da questão energética em RSSFS.

Artigo	Métricas
[42]	Taxa de transferência (bps)
[11]	Eficiência energética
[12]	Eficiência energética
[13]	Taxa de transferência (mbps), déficit de energia no receptor
[14]	Atenuação da tensão induzida na entrada do RX
[15]	Valor ótimo (bps/Hz), Sinal para ruído de rádio - SNR (db), Energia consumida (J), Taxa de transferência (mbps)
[16]	Energia transmitida (dBm), taxa de transferência (mbps), sinal para ruído de rádio - SNR (db)
[43]	Consumo total de energia (J), porcentagem de energia economizada
[44]	Taxa de transferência (bps)
[45]	Perda do caminho (dB)

Uma das questões que interferem no consumo energético é o posicionamento dos dispositivos. Na utilização de indução magnética, é possível utilizar uma arquitetura de agrupamento (cluster) em uma forma de comunicação 3D [43], onde é levada em consideração uma forma que permita uma boa distribuição de energia entre os nós. O modelo proposto pode ser utilizado na água, em solo seco e em rocha sedimentar úmida. Os resultados encontrados é que com essa técnica é possível reduzir em até 84,37% o consumo energético.

Existe um campo de estudo com foco nesses dispositivos capazes de transmitir informação e energia sem a necessidade de fios. Em [13], o problema de transferência simultânea de energia e informações sem fio em um canal de acesso múltiplo com desvanecimento e ruído gaussiano aditivo é tratado. Nesse cenário, o dispositivo transmissor envia informações para o receptor, que pode ser utilizado para enviar energia para o transmissor. A proposta é que a transmissão de energia e de informação ocorra de maneira simultânea.

A otimização da utilização dos recursos em redes de sensores subterrâneos com alimentação de energia por meio sem fio é tratada em [15] com o objetivo de maximizar a taxa de transferência levando em consideração a qualidade do serviço (QoS). Em [16], esse estudo avança com o propósito de otimizar a alocação simultânea de energia e tempo sob as influências do solo. A proposta apresentada é de utilização da técnica MIMO.

Um dos problemas decorrentes da limitação energética é a falta de confiabilidade da rede. Em [44], esse problema é tratado com um mecanismo que possui três tipos de canais: de comunicação por indução magnética, por ondas eletromagnética e de transferência de energia por meios sem fio. Então, é proposto um protocolo para um acesso confiável e energeticamente eficiente a esses canais. Nesta abordagem, são utilizadas quatro formas de rede e sete tipos de dispositivos, dentre eles, veículos aéreos não tripulados.

2.7 VUR das RSSFS

Assim como as demais redes de sensores, as RSSFS enfrentam o desafio de maximização da VUR [18]. Os problemas das RSSFS já mencionados de necessidade de funcionamento em ambientes instáveis, de posicionamento ideal dos dispositivos, de degradação da qualidade da comunicação e de recursos energéticos limitados diminuem a VUR e, portanto, influenciam na perda de cobertura desse tipo de rede. Em primeiro lugar, os ambientes instáveis das RSSFS podem vir a danificar algum dispositivo, o que interromperia a cobertura da região que aquele dispositivo encontrava-se. Em segundo lugar, o posicionamento dos dispositivos interfere na cobertura da rede, já que dispositivos distantes podem consumir muita energia para transmitir informações ou podem até mesmo não conseguirem fazer essa transmissão, ficando dessa forma sem cobertura na região que o dispositivo distante encontra-se. Em terceiro, a degradação da qualidade da comunicação no solo impede o envio de dados e limita a cobertura da rede. Por fim, como um dispositivo sem energia não funciona, a restrição energética desse tipo de dispositivo impede a cobertura da região que aquele dispositivo está. A solução apresentada nessa dissertação é de utilizar um protocolo de programação sono-vigília que atue em conjunto com TSFE nas RSSFS para a maximização da vida útil destas redes.

Capítulo 3

Trabalhos Relacionados

Afim de relacionar a solução apresentada nesta dissertação com as soluções já propostas por outros autores, foram buscados termos relacionados com redes de sensores subterrâneos como *wusn*, *sensor* e *underground*, e também foram buscados termos relacionados a VUR como *energy* e *power*. Usou-se a seguinte string de busca na base da Scopus: (TITLE-ABS-KEY ((sensor AND underground AND power AND transfer AND magnetic AND induction))) OR (TITLE-ABS-KEY (mac AND duty-cycle AND receiver-initiated AND asynchronous AND wireless AND sensor AND network AND dynamic AND traffic)) OR (TITLE-ABS-KEY (wireless AND sensor AND network AND underground AND agriculture AND energy AND (powering OR powered OR transfer))) AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018)). O resultado dessa consulta é apresentado na Tabela 3.1, onde traz o nome do artigo, se é uma RSSFS, se utiliza *Multiple Inputs, Multiple Outputs* (MIMO), se relaciona-se ao agro, se usa drone na solução, se está relacionado a maximização da vida útil da rede, se utiliza alguma técnica de recarregamento de bateria (RB) e se utiliza a técnica de programação sono-vigília.

Todos os trabalhos relacionados são sobre RSSFS. O foco desses trabalhos é maximizar a VUR e todos utilizam alguma forma de RB. Dentre esses trabalhos, alguns utilizam de alguma forma técnicas MIMO ou drones. Tal como acontece com a avaliação do protocolo apresentado neste artigo, alguns trabalhos são aplicados à agricultura.

Em [46], a abordagem para maximizar a VUR foi através da técnica de transferência sem fio de energia (TSFE) usando radiofrequência (RF). O trabalho apresentado em [47] trata a transferência de energia como um sistema de transferência de energia por indução (TEI) em uma RSSFS e a transmissão deste sistema de energia sem fio foi adaptado para missões periódicas de carregamento rápido com um drone. Em [48], as telhas de drenagem agrícola são utilizadas como guias de onda para transportar energia eletromagnética (EE)

Tabela 3.1: Trabalhos relacionados.

Artigo	RSSFS	MIMO	Agro	Drone	VUR	RB	Sono-Vigília
[46]	✓		✓		✓	✓	
[47]	✓		✓	✓	✓	✓	
[48]	✓		✓		✓	✓	
[49]	✓				✓	✓	
[50]	✓			✓	✓	✓	
[51]	✓				✓	✓	
[52]	✓		✓		✓	✓	
[53]	✓				✓	✓	
[54]	✓		✓		✓	✓	
[55]	✓	✓	✓		✓	✓	
SWPTMAC	✓	✓	✓	✓	✓	✓	✓

por longas distâncias.

Além das técnicas TSFE já mostradas, em [53] e [49] uma frequência adaptativa é usada na TSFE para maximizar a transferência de energia através de areia ou pedras. Em [50], é feito um estudo sobre o uso da indução magnética (IM) na maximização da VUR, e um protocolo Q-Learning é proposto. Em [51], os sinais acústicos são usados para transferência de energia sem fio. Em [54] e [52], os sensores são alimentados pela diferença de temperatura entre o ar e o solo subterrâneo. Por fim, [55] apresenta o uso da técnica MIMO.

Detalhando o que cada trabalho relacionado apresenta, o artigo [46] combina a tecnologia de comunicação por retroespalhamento (*backscatter*) com TSFE. A tecnologia utilizada usa a energia recebida por RF para alimentar suas operações de retroespalhamento. Isso significa que a energia recebida por RF pode ser aproveitada tanto para a comunicação quanto para o fornecimento de energia ao sensor.

Uma contribuição de [46], é a eficiência energética possibilitada pela tecnologia utilizada. Além disso, a eficiência energética contribui para uma maior autonomia dos dispositivos. Outro ponto de contribuição do artigo é um modelo sofisticado para otimização da taxa de transferência da RSSFS. A QoS dos usuários também é levada em consideração.

Além das contribuições já apresentadas já citadas de [46], os resultados da solução foram comparados com os resultados de outras soluções. Os resultados mostram que a solução proposta supera significativamente outras soluções de referência em termos de desempenho da rede. O artigo também identifica vários fatores que afetam o desempenho da rede e fornece orientações para a implementação prática de RSSFS assistidas por retroespalhamento.

O trabalho apresentado em [46] relaciona com o trabalho apresentado na dissertação pois ambos os trabalhos estão focados em melhorar a eficiência e prolongar a VUR de RS-

SFS, cada um com abordagens e técnicas específicas para atingir esse objetivo. Enquanto [46] combina TSFE com retroespalhamento, o trabalho apresentado na dissertação combina TSFE e programação sono-vigília. Ambos os trabalhos contribuem para o avanço da tecnologia de rede de sensores aplicada à agricultura inteligente e sustentável.

O trabalho apresentado em [47] utiliza TEI para uma RSSFS. Eles demonstram o projeto e construção desse sistema. Além disso, são apresentados os resultados de experimentos de campo e laboratório. O artigo aborda um problema significativo na agricultura moderna: como monitorar o solo sem a necessidade de hardware acima do solo ou substituição periódica de baterias.

As contribuições apresentadas em [47] incluem a demonstração de eficiência em carregamento, a adaptação de um drone para recargas periódicas e a significativa redução do consumo de energia dos sensores. Esses avanços são relevantes para a agricultura, pois fornecem uma solução para o monitoramento do solo, ajudando na conservação da água e no controle da irrigação.

Os resultados alcançados por [47] mostram que a técnica de maximização da VUR foi capaz de realizar a recarga e fornecer energia para um sensor em uma RSSFS agrícola. Durante os experimentos de campo, foi observada uma eficiência média de carga de 33,5% com uma taxa de carga máxima de 35 W, apesar de um aumento significativo das perdas no final para 5%. Além disso, o artigo aborda os desafios técnicos enfrentados durante a integração do sistema, como o gerenciamento de dados, a impermeabilização do equipamento e a interferência eletromagnética.

Relacionando [47] com o trabalho nesta dissertação verificamos que ambos buscam maximizar a VUR de RSSFS aplicadas na agricultura. Além disso, ambos os trabalhos utilizam drones para realizar a TSFE. A diferença é que no artigo existe um foco maior no hardware implementado, enquanto o trabalho da dissertação é focado no protocolo desenvolvido.

No que diz respeito ao trabalho apresentado em [48], o foco principal é a apresentação de uma nova técnica para transferência de energia para sensores em uma RSSFS. Essa técnica se utiliza de canalizações de drenagem agrícola como guias de ondas para transmitir energia eletromagnética a longas distâncias. A energia eletromagnética é então convertida em energia contínua (DC) para recarregar as baterias dos sensores enterrados. O artigo descreve detalhadamente o projeto e a implementação de um sistema retificador de 1 GHz em um experimento com canalizações de drenagem agrícola de 8 polegadas. Além disso, são medidas diversas variáveis do sistema, como eficiência de conversão e resposta à presença de água nas canalizações.

O trabalho apresentado em [48] oferece contribuições significativas para a área de RSSFS, ao introduzir uma abordagem inovadora para o fornecimento de energia a sensores

agrícolas enterrados. Ao utilizar os sistemas de drenagem agrícola, a técnica supera os desafios de perda de transmissão e eficiência energética associados ao carregamento sem fio através do solo. Além disso, as medições experimentais realizadas evidenciaram a viabilidade e eficácia dessa abordagem para a área agrícola.

O estudo de [48] apresenta ganhos significativos em relação a algumas soluções para o fornecimento de energia a sensores agrícolas enterrados. Além disso, a eficiência energética alcançada pelo sistema foi de até 29,64%, o que é considerado um valor bastante elevado para esse tipo de aplicação. Esses resultados indicam que a técnica proposta pode ser uma solução viável e eficaz para o fornecimento de energia a RSSFS.

A proposta apresentada por [48] assemelha-se com a solução proposta nesta dissertação pelo objetivo de maximizar a VUR de uma RSSFS. No entanto, a forma de fornecimento de energia para os sensores enterrados é diferente. A proposta apresentada no artigo abre uma possibilidade de pesquisa futura, onde a técnica de programação sonovigília atue em conjunto com essa outra técnica de carregamento dos sensores enterrados.

Os artigos [53] e [49] apresentam o desenvolvimento de uma nova tecnologia de sensores sem fio, chamada FracBot, para mapear fraturas hidráulicas e monitorar reservatórios não convencionais. Esses dispositivos são injetados durante a operação de fraturamento hidráulico e utilizam comunicação baseada em IM para transmitir dados coletados de volta para a estação base. O artigo discute a arquitetura do sistema, incluindo o design do FracBot, a rede de comunicação, a estrutura de energia e a maximização da VUR. Mostra ainda os resultados de testes experimentais e futuras direções de pesquisa.

O trabalho apresentado em [53] e [49] busca maximizar a VUR da rede de FracBots de várias maneiras. Primeiramente, os FracBots são projetados com unidades de colheita de energia que lhes permitem obter energia do ambiente, reduzindo a dependência de baterias com vida útil limitada. Além disso, o artigo propõe um modelo de carregamento de energia eficiente, no qual a estação base transmite energia para os FracBots de forma a superar as restrições de condutividade do óleo no ambiente de fratura. Isso garante que os FracBots recebam energia suficiente para operar, mesmo em ambientes desafiadores. Adicionalmente, o artigo descreve um protocolo de comunicação de camadas cruzadas que otimiza o consumo de energia e a eficiência de transmissão, garantindo que a rede opere de forma eficiente.

Em [53], que é a parte do trabalho focada em conectividade, foi possível alcançar uma taxa de transferência de dados de 1 kbps em um ambiente de extrema dificuldade de comunicação. Já o protocolo de camada cruzada, explorado em [49], reduziu o consumo energético em 50%. Além disso, foi possível obter uma taxa de 95% de localização dos sensores subterrâneos. Os resultados foram promissores para a tecnologia apresentada.

Quando relacionamos o trabalho apresentado em [49] com o trabalho apresentado

nesta dissertação, encontramos diferenças nas formas que os sensores enterrados recebem energia. No caso da tecnologia Fracbot, a energia é obtida através de uma colheita de energia no próprio ambiente. Já no trabalho apresentado nesta dissertação, a energia é transferida por IM de um drone para os sensores enterrados. Além disso, ambos os trabalhos propõem protocolos de comunicação como parte da solução adotada.

O artigo [50] aborda o conceito de RSSFS alimentadas por energia sem fio com assistência de IM e seu potencial para enfrentar desafios de confiabilidade nestas redes. O autor propõe integrar técnicas de comunicação por IM com mecanismos de TSFE para melhorar a confiabilidade e permitir uma coleta de dados eficiente. O artigo fornece uma revisão bibliográfica no que diz respeito a implantação de sensores, controle de acesso a múltiplos canais e estabelecimento de roteamento seletivo de frequência. O autor descreve os desafios e questões futuras relacionadas em cada direção de pesquisa. O artigo também apresenta a arquitetura vislumbrada para RSSFS que utilizam IM e TSFE, descrevendo os dispositivos envolvidos, como coletores de dados, provedores de energia, sensores subterrâneos e agentes aéreos. O modelo de rede para este cenário também é apresentado, mostrando o fluxo de energia, dados de ondas eletromagnéticas e dados de IM entre os componentes.

O autor de [50] apresenta um estudo de caso relacionado à implantação de múltiplos sensores em locais subterrâneos usando uma nova topologia baseada em projeção bidirecional. Os resultados mostraram que essa abordagem permitiu uma melhoria significativa na conectividade entre os sensores ativos e os centros de dados. Além disso, o documento discute os resultados alcançados com o controle de acesso a múltiplos canais e o estabelecimento de roteamento seletivo de frequência. Os resultados indicaram que essas técnicas contribuíram para uma maior confiabilidade do sistema e eficiência energética.

Assim como o estudo apresentado em [50], o trabalho apresentado nesta dissertação também aborda os desafios de confiabilidade e eficiência energética nas redes de sensores subterrâneos. A integração das técnicas de IM e TSFE proposta no artigo é uma solução também utilizada no trabalho da dissertação. Os trabalhos também estão relacionados por usarem drones como uma solução possível de se integrar com RSSFS. Além disso, os trabalhos se relacionam ao permitir a coleta confiável de dados e o prolongamento da VUR de RSSFS.

O artigo [51] trata do projeto de um sistema de comunicação sem fio e transferência de energia para robôs em tubulações e sensores subterrâneos. O ambiente extremamente subterrâneo apresenta desafios para as tecnologias tradicionais de comunicação sem fio, por isso, os autores propõem o uso de um sistema híbrido que combina sinais acústicos para transferência de energia sem fio e campos magnéticos de baixa e alta frequência para comunicação. O artigo descreve a simulação dos sinais acústicos e campos magnéticos

usando simulações em ondas completas, bem como o projeto de um laboratório de testes.

O trabalho apresentado em [51] é de grande relevância para a área de monitoramento subterrâneo de tubulações. O trabalho ainda contribui com a comunicação acústica em RSSFS. Além disso, há uma combinação de sinais acústicos com campo magnético. O trabalho traz explicações sobre os experimentos e resultados que podem ser de grande utilidade para outros pesquisadores que desejarem investigar mais profundamente as áreas citadas.

Quanto aos resultados apresentados em [51], é possível concluir que uma taxa de transferência atingida no final de um tubo de metal de 8 mm de espessura é de 50 kbps, com uma distância de 2 cm e uma frequência portadora de 100 kHz. Os autores realizaram experimentos para medir a potência recebida e a taxa de perda de pacotes em diferentes configurações. Foi observado que a potência recebida diminuiu significativamente quando a bobina transmissora estava dentro do tubo de metal, indicando que o sinal é fortemente afetado pelo ambiente de metal. Além disso, foi constatado que a taxa de perda de pacotes aumenta com a taxa de símbolos e que a comunicação confiável é possível com uma taxa de transferência inferior a 20 kbps. Esses resultados mostram a viabilidade do sistema proposto para a comunicação e transferência de energia sem fio em ambientes subterrâneos extremos.

O trabalho apresentado em [51] relaciona-se com o trabalho apresentado na dissertação no que diz respeito à busca por uma maximização da VUR. Além disso, a busca por avaliar as taxas de transferências na RSSFS é um objetivo de ambos os trabalhos. Os trabalhos se diferenciam nas técnicas utilizadas na maximização da VUR. O protocolo proposto na dissertação poderia ser adaptado para um cenário onde a energia fosse recebida pelos sensores da mesma forma que no artigo citado.

O trabalho mostrado em [54] e [52] apresenta um sensor de monitoramento de solo sem bateria para agricultura que utiliza a diferença de temperatura entre o ar próximo à superfície e o solo raso subterrâneo. Com tecnologias inovadoras de colheita de energia, este sensor pode operar semipermanentemente em grandes áreas agrícolas onde a fonte de energia é indisponível. O artigo discute os fatores-chave que determinam o desempenho do sistema acionado pelo gerador termoelétrico e avalia o desempenho do coletor de energia, bem como as potenciais aplicações dessa tecnologia em RSSFS agrícolas.

A abordagem apresentada em [54] e aperfeiçoada em [52] oferece uma solução sustentável e de baixo custo para o monitoramento subterrâneo do solo em áreas extensas. A capacidade do sensor de operar semipermanentemente em ambientes remotos e de difícil acesso, sem a necessidade de substituição frequente de baterias, representa um avanço significativo no desenvolvimento de tecnologias para a agricultura de precisão e a otimização das práticas agrícolas. Além disso, a tecnologia apresentada poderá ser utilizada

em outras áreas além da agricultura.

Os resultados alcançados por [54] e [52] mostram que o sensor de monitoramento de solo sem bateria, impulsionado pela diferença de temperatura entre o ar próximo à superfície e o solo raso subterrâneo, pode gerar uma quantidade significativa de energia em diferentes condições climáticas. O estudo revelou que o sistema proposto pode colher algumas centenas de microwatts em todos os campos e condições testadas, com um aumento de cerca de 20% na geração de energia em ventos de 3 m/s em comparação com ventos de 0,25 m/s. Além disso, a relação entre a saída de energia e a diferença de temperatura não é linear, e a saída de energia pode variar em diferentes estações do ano e locais. Embora seja uma boa alternativa tecnológica, a solução não pode ser considerada universal pelas restrições de local e momentos do ano. No entanto, os resultados demonstrados no artigo validam a eficácia do sensor em coletar energia de forma eficiente a partir da diferença de temperatura e fornecem informações importantes para a otimização do projeto do sensor em várias condições climáticas e locais.

A abordagem inovadora mostrada tanto em [54], quanto em [52] para a colheita de energia a partir da diferença de temperatura pode ser aplicada em conjunto com o protocolo apresentado nesta dissertação para maximizar a vida útil dos sensores subterrâneos, permitindo a monitorização semipermanente de condições do solo em ambientes remotos e de difícil acesso. A combinação dessas tecnologias pode fornecer uma solução eficiente e sustentável para a monitorização de condições do solo em larga escala. Essa é uma possível proposta de pesquisa futura.

O artigo [55] trata dos avanços recentes em RSSFS baseadas em IM. O artigo discute os desafios e soluções para estabelecer conexões sem fio eficientes nessas condições, incluindo modelagem de canal, transmissão e processamento de sinal digital, sincronização, design de rede, TSFE e localização. Além disso, o artigo menciona as tentativas de implementação e validação por alguns grupos de pesquisa inovadores. O foco da pesquisa futura nesta área parece estar em técnicas de otimização de camadas cruzadas mais avançadas, incorporando múltiplas funcionalidades das RSSFS baseadas em IM, como o design de redes de alta taxa de transferência e eficientes em energia.

O trabalho apresentado em [55] é de grande relevância para a área de TSFE por abordar a aplicação desta técnica em ambiente de difícil acesso. Além disso, os avanços no que diz respeito aos ajustes de parâmetros para a TSFE pode ser de grande utilidade para pesquisas futuras nesta área. Outra parte que o artigo citado contribui é na área de comunicação utilizando IM. A comunicação usando IM é explorada em diferentes ambientes como no espaço livre e em líquidos. Por fim, o artigo identifica direções promissoras para futuras investigações e aplicações, incluindo técnicas de otimização de camadas cruzadas e o uso de relés de IM ativos e passivos com diferentes padrões de operação.

O artigo [55] apresenta uma grande contribuição em termos de investigação teórica nas áreas de RSSFS. No entanto, as contribuições do artigo não ficaram restritas as áreas teóricas. A equipe de pesquisa conseguiu estabelecer uma comunicação de até 35 metros em uma RSSFS utilizando IM. Esses resultados práticos indicam que essas tecnologias têm o potencial de oferecer soluções viáveis e eficientes para aplicações de monitoramento e previsão em ambientes subterrâneos.

O trabalho apresentado em [55] apresenta a IM como uma solução viável para a TSFE. Diante do que foi apresentado neste artigo, e em alguns outros trabalhos, foi tomada a decisão de utilizar IM na pesquisa apresentada nesta dissertação. O artigo também apresenta uma direção de pesquisa futura sobre a utilização de MIMO em RSSFS que foi seguida pela pesquisa apresentada na dissertação. Além disso, tanto o trabalho apresentado no artigo como o trabalho apresentado na dissertação tratam da importância da maximização da VUR quando se trata de RSSFS.

Portanto, os trabalhos relacionados apresentados trazem diferentes melhorias para RSSFS e maximizações da VUR destas redes. Porém, nenhum dos trabalhos relacionados explora uma técnica de programação sono-vigília que ao mesmo tempo otimize o uso da transferência de energia, visando maximizar a VUR, permanecendo como uma lacuna aberta para investigação. Portanto, neste trabalho, exploramos esta lacuna de pesquisa, fornecendo uma solução que pode maximizar o VUR, introduzindo uma técnica aprimorada de programação sono-vigília que otimiza o uso da TSFE em RSSFS.

Capítulo 4

Solução proposta

Neste capítulo será apresentada a solução usada na maximização da VUR de uma RSSFS. Com a finalidade de estender a VUR, o trabalho aqui apresentado utiliza a técnica de programação sono-vigília [18] em conjunto com a técnica de TSFE[27]. A programação sono-vigília é feita colocando os rádios em estado de adormecimento e ativando-os apenas para enviar alguma informação, realizando a implementação de um novo protocolo baseado no EERIMAC chamado SWPTMAC e que é comparado neste trabalho com os já existentes protocolos de programação sono-vigília. Já a TSFE será feita através de um drone que irá até próximo aos sensores enterrados.

A Figura 4.1 ilustra todos os elementos envolvidos na solução desenvolvida. A área de aplicação caracterizada foi a agricultura, onde os sensores enterrados coletam informações do solo e enviam através de rádios de ondas eletromagnéticas para o coletor de dados que fica instalado no braço de um pivô central. O rádio caracterizado na simulação foi o CC2420 que já foi explorado em RSSFS [56] e na programação sono-vigília [17]. A TSFE será feita por um drone que viajará de sua base até o local que estiver o sensor enterrado para realizar a transferência de energia para esse sensor através de bobinas de indução magnética. A técnica de avaliação da solução foi simulação, baseada na implementação feita por [57].

4.1 SWPTMAC: Protocolo MAC de Programação Sono-Vigília e Transferência de Energia

Em vez de propor um novo protocolo de programação sono-vigília do zero, preferimos selecionar como base um protocolo de programação sono-vigília já popular, nomeadamente o protocolo EERIMAC. No EERIMAC os horários dos emissores são diferentes, com menor probabilidade de contenção, conseguindo maior conservação de energia. Os

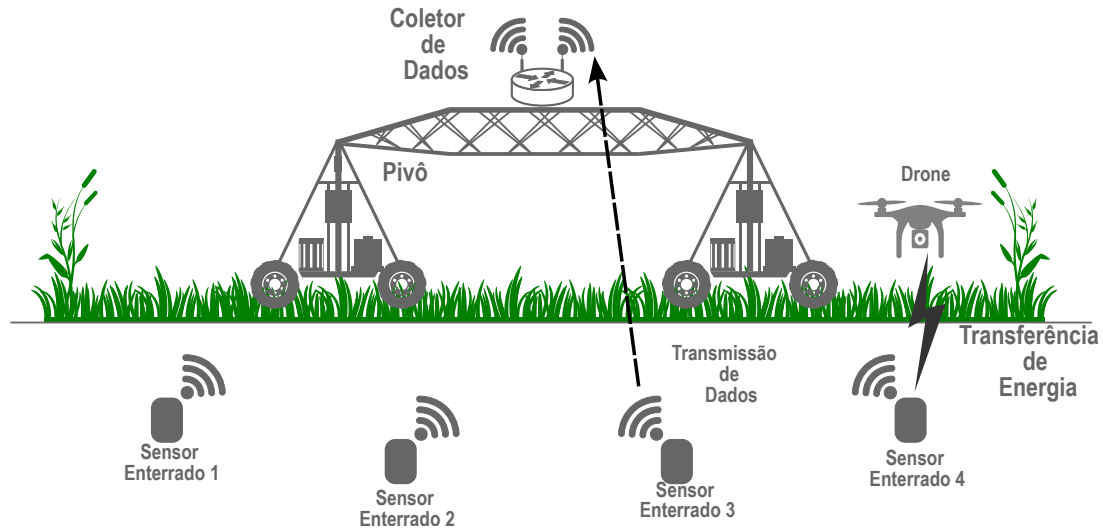


Figura 4.1: Representação da solução.

intervalos de tempo de espera e suspensão são parâmetros críticos para determinar o desempenho do protocolo em relação ao ciclo de trabalho médio, taxa de transferência e atraso ponta a ponta. EERIMAC realiza a comunicação considerando o envio ao estado SLEEP_WAIT_BEACON caso um BEACON esteja demorando para chegar.

Diferente do EERIMAC, propomos o protocolo SWPTMAC, que funciona de forma semelhante ao EERIMAC, mas introduzindo o recurso de TSFE. SWPTMAC inclui o estado SLEEP_WAIT_BEACON caso um BEACON demore muito para chegar, considerando também os intervalos de espera e de suspensão como parâmetros primordiais. Os demais estados utilizados na comunicação pelo SWPTMAC são semelhantes aos estados do protocolo RIMAC, que serviu de base para surgimento do EERIMAC.

A diferença entre SWPTMAC e EERIMAC é o estado adicional denominado RECEIVING_ENERGY e suas ações derivadas necessárias para cessar as atividades de comunicação do dispositivo para receber carga do equipamento que está transferindo energia. Neste caso, um drone desempenha o papel de um equipamento de recarga capaz de realizar a TSFE. O estado adicional RECEIVING_ENERGY é acessado quando chega um drone com energia para transmitir conforme ilustrado no lado direito da Figura 4.1, entre o drone e o Sensor Subterrâneo 4.

Para realizar a transmissão de dados e reagir a TSFE, na Figura 4.2, apresentamos as máquinas de estado desenvolvidas para explicar o SWPTMAC. Um dispositivo implementado com SWPTMAC pode estar em duas situações: ser um dispositivo aguardando recebimento de dados ou realizando transferência de dados. No primeiro caso, os parâmetros do dispositivo receptor são configurados e seu rádio entra em modo de adormecimento conforme representado pela máquina de estados na Figura 4.2(b). Caso receba algum dado, o receptor envia uma confirmação confirmando a chegada dos dados. Após

o recebimento dos dados, o dispositivo retorna ao estado de espera ficando ocioso para aguardar mais dados. Caso chegue um estímulo de transferência de energia, o dispositivo só iniciará a transferência de energia nos casos em que os dados forem totalmente recebidos ou quando o dispositivo estiver ocioso no estado de adormecimento, passando para o estado RECEIVING_ENERGY. O transmissor segue uma estratégia diferente conforme explicado a seguir.

Um sensor subterrâneo SWPTMAC fica pronto para enviar dados inicializando os parâmetros do transmissor conforme ilustrado na Figura 4.2(a). Após a inicialização, o transmissor adormece e o rádio só se torna ativo após os módulos sensores incorporados, *por exemplo* Ph ou sensores de umidade do solo, coletarem alguns dados no ambiente monitorado. Depois disso, o dispositivo espera que um receptor demonstre através de um beacon que está pronto para receber dados. Em seguida, o dispositivo segue o processo de transmissão de dados. O transmissor pode ser induzido a receber energia enquanto espera pelos dados do sensor, enquanto espera por um receptor ou após completar a transmissão de dados entrando no estado RECEIVING_ENERGY.

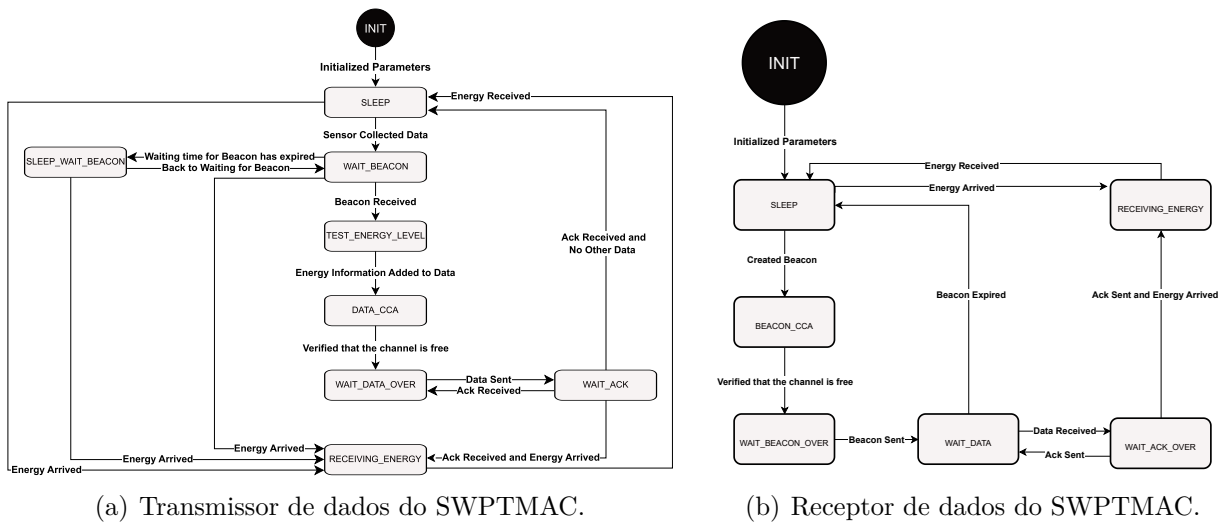


Figura 4.2: Máquinas de estados demonstrando o protocolo SWPTMAC.

4.2 Cenários

A área de aplicação das RSSFS é a importante área da agricultura, que precisa evoluir para atender a demanda da crescente população mundial [19]. Semelhante ao que foi desenvolvido em [7], será utilizado um pivô central na arquitetura da RSSFS. O pivô percorrerá um campo coletando os dados dos sensores enterrados. O cenário é representado na Figura 4.3. Como pode ser visto na figura, os dispositivos de sensoriamento subterrâ-

neo são dispostos em grade. A posição central do pivô é fixa e fica no centro da grade de dispositivos. O dispositivo coletor de dados fica instalado no braço do pivô. Sendo assim, a medida que o braço do pivô faz o movimento circular, o coletor de dados movimentar-se também e percorre o campo recebendo os dados dos sensores.

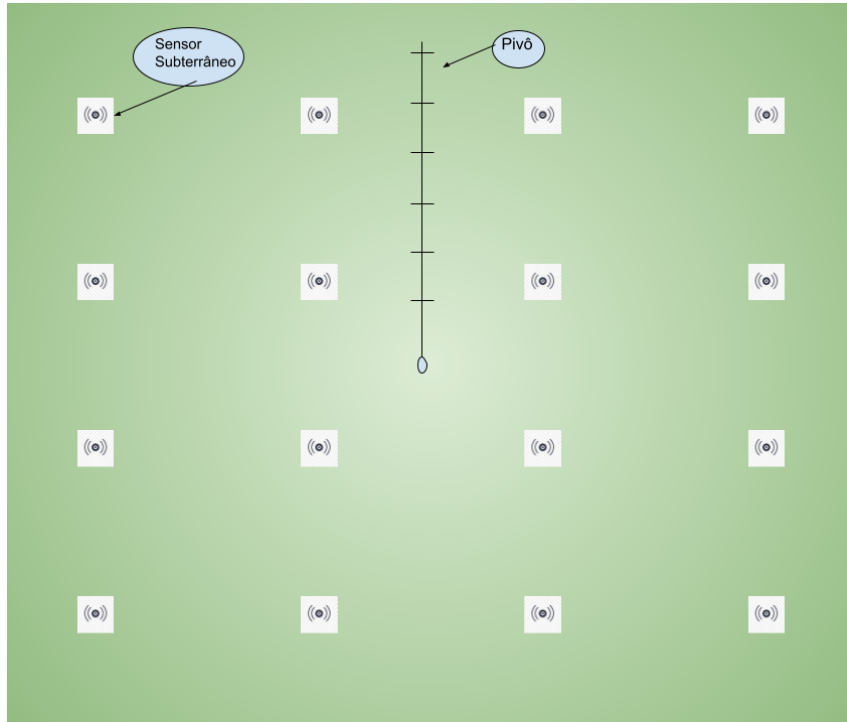


Figura 4.3: Cenário Proposto.

4.3 Ferramentas

Nesta seção, serão apresentadas as ferramentas que serão utilizadas nas simulações. De acordo com [58], o Castália, que é uma extensão do simulador OMNeT++, é adequado para simular RSSF e baseado nisso ele foi escolhido para realização dessa pesquisa.

4.3.1 OMNeT++

O OMNeT++¹ é um simulador desenvolvido em C++ baseado em módulos e componentes. O OMNeT++ é utilizado principalmente para simulação de redes. O simulador fornece uma interface de desenvolvimento baseada no Eclipse. Afim de atender domínios específicos, existem extensões que são integradas ao OMNeT++.

¹<https://omnetpp.org/>

4.3.2 Castália

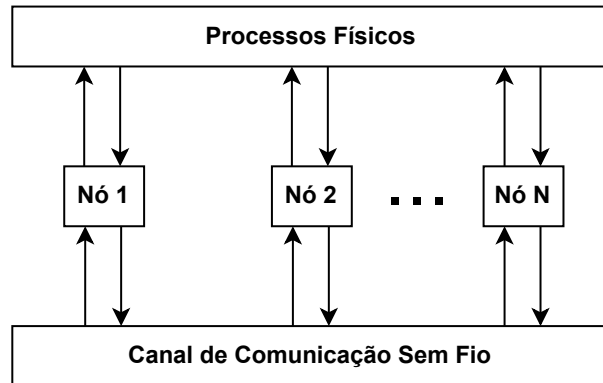


Figura 4.4: Estrutura Básica do Castália.

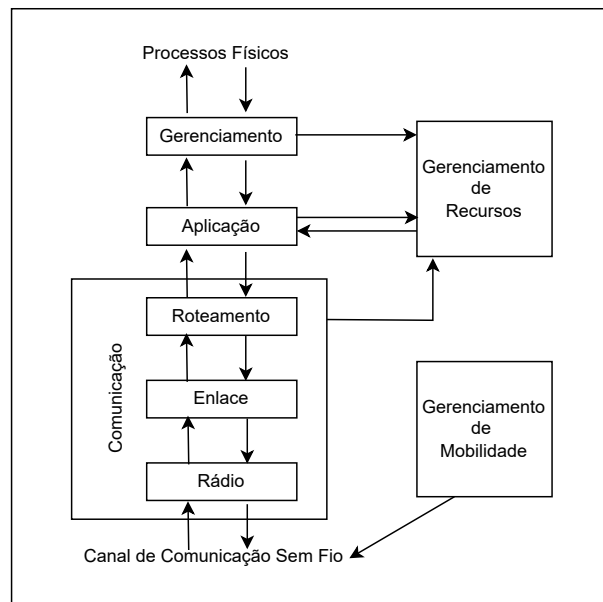


Figura 4.5: Módulos de um Nó no Castália.

Uma das extensões do OMNeT++ é o Castália ². Castália é utilizado principalmente para fazer simulação de RSSF. No Castália, existem modelos já desenvolvidos e altamente paramétricos que facilitam o processo de realização de simulações. O Castália é utilizado por pesquisadores e desenvolvedores para testar algoritmos e protocolos. A Figura 4.4 mostra a estrutura básica do Castália, onde são representados o processo físico (*Physical process*), os nós (*Node*) e o canal sem fio (*Wireless Channel*). O principal módulo do Castália é o nó que é representado na Figura 4.5. As representações são baseadas nas apresentadas em [59].

²<https://omnetpp.org/download-items/Castalia.html>

Como pode ser visto na Figura 4.4 os nós se conectam por meio do módulo de canal sem fio [59]. As setas significam a passagem de mensagens de um módulo para outro. Quando um nó tem um pacote para enviar este vai para o canal wireless que então decide quais nós devem receber o pacote. Os nós também são vinculados por meio dos processos físicos que eles monitoram. Para cada processo físico existe um módulo que contém informações o processo físico no espaço e no tempo (enviando uma mensagem para o módulo correspondente) para obter as leituras de seus sensores.

Como pode ser visto na Figura 4.5, o módulo do nó é composto por outros módulos. As setas sólidas significam passagem de mensagem entre módulos. O módulo *Application* é aquele usado para implementar os algoritmos da camada de aplicação. Os módulos de comunicação *MAC* e *Routing*, bem como o módulo *Mobility Manager*, também são usados para implementar novos protocolos ou padrão de mobilidade, respectivamente. A solução apresentada nessa dissertação é de implementar os protocolos de programação sono-vigília, adaptado à TSFE, usando o módulo *MAC* e caracterizar um pivô central utilizando o módulo *Mobility Manager*.

Tendo em vista que para implementação da solução aqui apresentada são necessárias as caracterizações de pivô e drone, o desenvolvimento foi feito. O pivô foi caracterizado através de uma implementação do módulo *Mobility Manager* do framework. A implementação do módulo *Mobility Manager* foi chamada *CircleMobilityManager* e é responsável pelo movimento circular do dispositivo coletor de dados. Já o drone foi um módulo implementado sem base em algum módulo já existente. O módulo do drone realiza o cálculo de tempo de viagem e da quantidade de energia que pode ser transferida do drone para o sensor enterrado.

Capítulo 5

Protótipo

Este capítulo trará os pseudocódigos que foram desenvolvidos para fazer a implantação dos protocolos RIMAC, EERIMAC e SWPTMAC no simulador OMNET, através do framework Castália. Como base inicial do desenvolvimento da programação sono-vigília foram utilizados os trechos de códigos apresentados em [60]. A primeira seção deste capítulo, trará os pseudocódigos que são iguais nos três protocolos. Já as demais seções trarão os pseudocódigos que diferenciam a implementação de cada protocolo. Além disso, há uma explicação de cada linha do pseudocódigo nos parágrafos a seguir. O objetivo deste capítulo é de trazer uma explicação detalhada que possa contribuir com futuros pesquisadores que forem implementar tais protocolos.

5.1 Códigos usados igualmente por RIMAC, EERIMAC e SWPTMAC

Nesta seção, serão apresentados os códigos que são usados tanto pelo RIMAC, quanto pelo EERIMAC e pelo SWPTMAC. O Algoritmo 1 apresenta o método que envia os beacons. Já o Algoritmo 2 apresenta o método auxiliar que verifica se o canal está livre antes de enviar algum pacote. Enquanto o Algoritmo 3 apresenta o comportamento dos protocolos quando recebem algum pacote via rádio.

O Algoritmo 1 inicia criando as variáveis que serão utilizadas no código. A linha 1 cria um quadro da camada de enlace. Já a linha 2 chama o método que verifica o tipo de beacon que deve ser gerado. Da linha 3 até a linha 13 existe um procedimento para cada tipo de beacon que deve ser gerado. Da linha 3 até a linha 5, é feita a criação de um beacon que inicia a comunicação. Da linha 6 até a linha 9, é criado um beacon que deve ser enviado no momento que um receptor de dados detecta uma colisão. Por fim, o beacon criado é colocado na fila de envio de dados do dispositivo na linha 14.

Algoritmo 1 Algoritmo que implementa as diferentes formas de Beacons que podem ser enviados no RIMAC e EERIMAC e SWPTMAC.

```
1: macFrame ← newMacPacket()
2: typeBeacon ← GETTYPEBEACON
3: if typeBeacon is ON_START then
4:   SETSOURCE(macFrame, SELF_MAC_ADDRESS)
5:   SETDESTINATION(macFrame, BROADCAST_MAC_ADDRESS)
6: else if typeBeacon is COLISION then
7:   SETBACKOFFTIME
8:   SETSOURCE(macFrame, SELF_MAC_ADDRESS)
9:   SETDESTINATION(macFrame, BROADCAST_MAC_ADDRESS)
10: else if typeBeacon is ACK then
11:   SETSOURCE(macFrame, SELF_MAC_ADDRESS)
12:   SETDESTINATION(macFrame, destination)
13: end if
14: PUSHToTXQUEUE(macFrame)
```

Algoritmo 2 Algoritmo que implementa como os protocolos verificam se o canal está livre.

```
1: CCACode ← GETCHANNELISCLEAR
2: if CCACode is CLEAR then
3:   returnTrue
4: else
5:   returnFalse
6: end if
```

O Algoritmo 2 inicia chamando um método comum em simuladores que traz um código do estado do canal na linha 1. Para este cenário dos protocolos simulados, o código do estado do canal que importa é o de canal limpo. Portanto, na linha 2 é verificado se este código é o retornado e, confirmando-se, retorna-se verdadeiro na linha 3, confirmando que o canal está livre. Em caso de ter outro código retornado, o método auxiliar retorna falso na linha 5.

Algoritmo 3 Algoritmo que implementa o comportamento de quando algum pacote chega pelo rádio.

```

1: if DetectCollision() then
2:   SUMBACKOFFTIMES()
3:   SENDBEACON(COLISION)
4: end if
5: macFrame  $\leftarrow$  DynamicCastPackage()
6: typePackage  $\leftarrow$  GETTYPEPACKAGE(macFrame)
7: if typePackage is ON_START then
8:   if HasPackageToSent then
9:     timeNextEvent  $\leftarrow$  simTime()
10:    SCHEDULEAT(timeNextEvent, BEACON_RECEIVED)
11:   end if
12: else if typePackage is COLISION then
13:   timeNextEvent  $\leftarrow$  simTime() + backoffTime
14:   SCHEDULEAT(timeNextEvent, BEACON_RECEIVED)
15: else if typePackage is ACK then
16:   timeNextEvent  $\leftarrow$  simTime()
17:   SCHEDULEAT(timeNextEvent, ACK_RECEIVED)
18: else
19:   SENDTONETWORKLAYER(macFrame)
20: end if

```

O Algoritmo 3 inicia verificando se mais de um dispositivo está tentando transmitir ao mesmo tempo e entre as linhas 1 e 4 faz a lógica para tratar tal situação. Na linha 1, verifica se encontra alguma colisão. Caso a colisão seja confirmada, é feita uma soma no fator de variação do *backoff* na linha 2 para evitar novas colisões. Então, é enviado um beacon para a rede na linha 3 informando que foi detectada alguma colisão. Nas linhas 5 e 6 cria-se as variáveis necessárias para processar os pacotes encaminhados via rádio para a camada de enlace. Da linha 7 até a 11 é feita a lógica executada quando um beacon de início de comunicação é recebido. Na linha 8, verifica se tem algum pacote para enviar para o dispositivo receptor que enviou o beacon. Na linha 9, é recuperado o tempo que está na simulação e na linha 10 é feito um agendamento de envio de pacote para um momento futuro. Da linha 12 até a 14 é feito o processamento que ocorre quando recebe um beacon informando que houve uma colisão, então, verifica e recuperado o tempo da

simulação na linha 13 e é feito um agendamento de um evento de reinício da transmissão de dados na linha 14. Da linha 15 até a 18 é feito o processamento que ocorre quando recebe um beacon informando que um pacote foi recebido. Por fim, na última linha o pacote é encaminhado para a camada de rede.

5.2 Códigos usados pelo RIMAC

Nesta seção, são apresentados os métodos da implementação do RIMAC. O Algoritmo 4 apresenta o método implementado para definir os estados do protocolo RIMAC. O método que trata os eventos do protocolo RIMAC é apresentado no Algoritmo 5. O tratamento que o protocolo RIMAC faz dos pacotes gerados pelo dispositivo e que chegam pela camada de rede até a camada de enlace é apresentado no Algoritmo 6.

Algoritmo 4 Os estados do protocolo RIMAC e o estado de rádio correspondente.

```

1: newState ← GETNEWSTATE()
2: SWITCH newState DO
3:   CASE SLEEP
4:     SETRADIOTO SLEEP
5:   CASE WAIT_BEACON_OVER
6:     SETRADIOTO TX
7:   CASE WAIT_DATA_OVER
8:     SETRADIOTO TX
9:   CASE WAIT_ACK_OVER
10:    SETRADIOTO TX
11:  CASE BEACON_CCA
12:    SETRADIOTO RX
13:  CASE DATA_CCA
14:    SETRADIOTO RX
15:  CASE WAIT_DATA
16:    SETRADIOTO RX
17:  CASE WAIT_BEACON
18:    SETRADIOTO RX
19:  CASE WAIT_ACK
20:    SETRADIOTO RX

```

No Algoritmo 4, quando cada estado é definido no dispositivo, é definido também um estado para o rádio do dispositivo. Para o estado *SLEEP* o rádio entra no estado de adormecimento. Nos estados *WAIT_BEACON_OVER*, *WAIT_DATA_OVER* e *WAIT_ACK_OVER* o dispositivo está com o rádio transmitindo, respectivamente, beacon, dados e confirmação de recebimento. Já nos estados *BEACON_CCA*, *DATA_CCA*, *WAIT_DATA*, *WAIT_BEACON* e *WAIT_ACK*, o dispositivo está com o rádio no estado de recepção. Na linha 1 deste algoritmo, é recuperado qual o estado que está sendo

definido no dispositivo. Na linha 2, inicia um *switch* de escolha para cada possível estado de ser definido. Da linha 3 em diante, em uma linha é verificado se determinado estado foi definido e na linha seguinte é definido o estado do rádio correspondente com aquele estado do dispositivo.

Algoritmo 5 Código dos comportamentos do protocolo RIMAC mediante cada evento recebido

```

1: eventType ← GETEVENTTYPE()
2: if eventType is WAKEUP then
3:   timeNextEvent ← simTime() + sleepInterval * (0.5 + randomNum())
4:   SCHEDULEAT(timeNextEvent, WAKEUP)
5: end if
6: SWITCH macState DO
7:   CASE SLEEP
8:     if eventType is WAKEUP then
9:       SETSTATE(BEACON_CCA)
10:      timeNextEvent ← simTime() + ccaInterval * (1 + randomNum())
11:      SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
12:     end if
13:   CASE BEACON_CCA
14:     if eventType is CCA_TIMEOUT then
15:       if !isChannelClear() then
16:         timeNextEvent ← simTime() + ccaInterval * (1 + randomNum()) ▷
Calcula o tempo de verificação do canal
17:         SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
18:       else
19:         SENDBEACON(ON_START, BROADCAST_CAST)
20:       end if
21:     end if
22:   CASE DATA_CCA
23:     if eventType is BEACON_RECEIVED then
24:       if !isChannelClear() then
25:         timeNextEvent ← simTime() + ccaInterval * (1 + randomNum())
26:         SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
27:       else
28:         SETSTATE(WAIT_DATA_OVER)
29:         timeNextEvent ← simTime()
30:         SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
31:       end if
32:     end if

```

O Algoritmo 5 começa definindo qual evento foi recebido. Na linha 1 do Algoritmo 5, é recuperado o evento que foi iniciado no dispositivo. Na linha 2, é verificado se o evento é para despertar o dispositivo e caso seja, o tempo do próximo evento de despertar é calculado na linha 3 e este evento é agendado na linha 4. Na sequência inicia a seção de

```

33:  CASE WAIT_BEACON_OVER
34:      if eventType is TRANSMISSION_ENDED then
35:          SETSTATE(WAIT_DATA)
36:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
37:          SCHEDULEAT(timeNextEvent, BEACON_TIMEOUT)
38:      end if
39:  CASE WAIT_DATA_OVER
40:      if eventType is TRANSMISSION_ENDED then
41:          SETSTATE(WAIT_ACK)
42:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
43:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
44:      else if eventType is CCA_TIMEOUT then
45:          if !txQueue.empty() then
46:              SENDFIRSTPACKAGEINQUEUE()
47:              timeNextEvent  $\leftarrow$  simTime() + timeToTX
48:              SCHEDULEAT(timeNextEvent, TRANSMISSION_ENDED)
49:          end if
50:      else
51:          SETSTATE(WAIT_ACK)
52:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
53:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
54:      end if
55:  CASE WAIT_ACK_OVER
56:      if eventType is TRANSMISSION_ENDED then
57:          SETSTATE(WAIT_DATA)
58:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
59:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
60:      end if

```

```

61:  CASE WAIT_DATA
62:      if eventType is BEACON_TIMEOUT then
63:          if txQueue.empty() then
64:              SETSTATE(SLEEP)
65:          else
66:              SETSTATE(WAIT_BEACON)
67:          end if
68:      else if eventType is DWELL_TIMEOUT then
69:          if txQueue.empty() then
70:              SETSTATE(SLEEP)
71:          else
72:              SETSTATE(WAIT_BEACON)
73:          end if
74:      end if
75:  CASE WAIT_ACK
76:      if eventType is ACK_RECEIVED then
77:          if !txQueue.empty() then
78:              SETSTATE(WAIT_DATA_OVER)
79:              SCHEDULEAT(0, CCA_TIMEOUT)
80:          else
81:              SETSTATE(SLEEP)
82:          end if
83:      end if
84:  CASE WAIT_BEACON
85:      if eventType is BEACON_RECEIVED then
86:          SETSTATE(DATA_CCA)
87:          SCHEDULEAT(0, BEACON_RECEIVED)
88:      end if

```

escolha do estado do dispositivo a partir da linha 6. Cada evento que pode ser recebido já tem um tratamento necessário de acordo com o estado em que o dispositivo se encontra. O objetivo deste algoritmo é atender o fluxo de estados das máquinas de estados do RIMAC apresentadas nas Figuras 2.1(a) e 2.1(b).

No Algoritmo 5 existem alguns padrões de processamento que se adequam para cada cenário. O primeiro padrão é observado nas linhas 3 e 4, onde é calculado o tempo que deve ocorrer um evento futuro e o agendamento deste evento. O segundo padrão é observado na linha 7, onde é verificado se o estado que o dispositivo encontra-se é o de adormecimento (*SLEEP*). O mesmo processo é feito para verificar se o dispositivo está em qualquer outro dos possíveis estados. Outro padrão é que, em cada estado do dispositivo, apenas determinados eventos são processados, por exemplo, na linha 8, verifica se um evento de despertar (*WAKEUP*) ocorreu e então inicia o processamento deste evento. Outro padrão que ocorre é o da linha 9, onde é chamado o método de definição de um novo estado para o dispositivo. Ainda existe também o padrão de verificação se o canal está livre como ocorre, por exemplo, na linha 15.

Algoritmo 6 Algoritmo que implementa a forma que o RIMAC processa os pacotes gerados pelo dispositivo.

```
1: macFrame ← newMacPacket()
2: ENCAPSULATEPACKAGE(macFrame, dataPackage)
3: SETSOURCE(macFrame, SELF_MAC_ADDRESS)
4: SETDESTINATION(macFrame, destination)
5: PUSHTOTXQUEUE(macFrame)
6: if macState is SLEEP then
7:   SETSTATE(WAIT_BEACON)
8: end if
```

O momento de geração do pacote é aquele em que o dispositivo realiza o sensoriamento do ambiente. O sensoriamento produz pacotes de dados que serão processados e transmitidos pelo dispositivo. No Algoritmo 6, é ilustrado o processamento que o dispositivo faz quando é gerado algum pacote. No RIMAC, é criado um quadro que encapsula os dados gerados na linha 2. Então, é definido os endereços do remetente e do destinatário nas linhas 3 e 4. Na sequência, o quadro gerado é adicionado na fila de envio do dispositivo na linha 5. Por fim, é verificado se o dispositivo está adormecido na linha 6 e caso isso se confirme o dispositivo entra no estado de espera por um beacon, na linha 7, para iniciar então a transmissão dos dados gerados.

5.3 Códigos usados pelo EERIMAC

Nesta seção, são apresentados os métodos da implementação do EERIMAC. São apresentados os algoritmos 7, 8, 9. Os algoritmos que implementam o EERIMAC são uma evolução em relação aos que foram apresentados referentes ao RIMAC. Além disso, eles serviram como base para o que foi implementado para o SWPTMAC.

Algoritmo 7 Os estados do protocolo EERIMAC e o estado de rádio correspondente.

```
1: newState ← GETNEWSTATE()
2: SWITCH newState DO
3:   CASE SLEEP
4:     SETRADIOToSLEEP
5:   CASE SLEEP_WAIT_BEACON
6:     SETRADIOToSLEEP
7:   CASE WAIT_BEACON_OVER
8:     SETRADIOToTX
9:   CASE WAIT_DATA_OVER
10:    SETRADIOToTX
11:   CASE WAIT_DATA_OVER
12:    SETRADIOToTX
13:   CASE BEACON_CCA
14:    SETRADIOToRX
15:   CASE DATA_CCA
16:    SETRADIOToRX
17:   CASE WAIT_DATA
18:    SETRADIOToRX
19:   CASE WAIT_DATA
20:    SETRADIOToRX
21:   CASE WAIT_ACK
22:    SETRADIOToRX
```

O Algoritmo 7 apresenta o método implementado para definir os estados do protocolo EERIMAC. Todos os estados do protocolo RIMAC, apresentados no Algoritmo 4, estão presentes no protocolo EERIMAC. Além destes estados presente no RIMAC, é acrescentado o estado *SLEEP_WAIT_BEACON* que também coloca o rádio do dispositivo no modo adormecimento. O padrão de comportamento do protocolo EERIMAC neste caso é o mesmo explicado para o protocolo RIMAC.

O método que trata os eventos do protocolo EERIMAC é apresentado no Algoritmo 8. Os eventos do protocolo EERIMAC são baseados no fluxo das máquinas de estados apresentadas das Figuras 2.2(a) e 2.2(b). O padrão explicado para o protocolo RIMAC é adotado também para o protocolo EERIMAC. O tratamento de eventos do EERIMAC possui duas situações que o caracterizam e o diferencia do tratamento de eventos do RIMAC. A primeira situação pode ser vista entre as linha 88 e 92, que retrata o tratamento

Algoritmo 8 Código dos comportamentos do protocolo EERIMAC mediante cada evento recebido.

```
1: eventType ← GETEVENTTYPE()
2: if eventType is WAKEUP then
3:   timeNextEvent ← simTime() + sleepInterval * (0.5 + randomNum())
4:   SCHEDULEAT(timeNextEvent, WAKEUP)
5: end if
6: SWITCH macState DO
7:   CASE SLEEP
8:     if eventType is WAKEUP then
9:       SETSTATE(BEACON_CCA)
10:      timeNextEvent ← simTime() + ccaInterval * (1 + randomNum())
11:      SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
12:    end if
13:   CASE BEACON_CCA
14:     if eventType is CCA_TIMEOUT then
15:       if !isChannelClear() then
16:         timeNextEvent ← simTime() + ccaInterval * (1 + randomNum())
17:         SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
18:       else
19:         SENDBEACON(0, BROADCAST_CAST)
20:       end if
21:     end if
22:   CASE DATA_CCA
23:     if eventType is BEACON_RECEIVED then
24:       if !isChannelClear() then
25:         timeNextEvent ← simTime() + ccaInterval * (1 + randomNum())
26:         SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
27:       else
28:         SETSTATE(WAIT_DATA_OVER)
29:         timeNextEvent ← simTime()
30:         SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
31:       end if
32:     end if
```

```

33:  CASE WAIT_BEACON_OVER
34:      if eventType is TRANSMISSION_ENDED then
35:          SETSTATE(WAIT_DATA)
36:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
37:          SCHEDULEAT(timeNextEvent, BEACON_TIMEOUT)
38:      end if
39:  CASE WAIT_DATA_OVER
40:      if eventType is TRANSMISSION_ENDED then
41:          SETSTATE(WAIT_ACK)
42:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
43:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
44:      else if eventType is CCA_TIMEOUT then
45:          if !txQueue.empty() then
46:              SENDFIRSTPACKAGEINQUEUE()
47:              timeNextEvent  $\leftarrow$  simTime() + timeToTX
48:              SCHEDULEAT(timeNextEvent, TRANSMISSION_ENDED)
49:          end if
50:      else
51:          SETSTATE(WAIT_ACK)
52:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
53:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
54:      end if
55:  CASE WAIT_ACK_OVER
56:      if eventType is TRANSMISSION_ENDED then
57:          SETSTATE(WAIT_DATA)
58:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
59:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
60:      end if

```

```

61:  CASE WAIT_DATA
62:      if eventType is BEACON_TIMEOUT then
63:          if txQueue.empty() then
64:              SETSTATE(SLEEP)
65:          else
66:              SETSTATE(WAIT_BEACON)
67:          end if
68:      else if eventType is DWELL_TIMEOUT then
69:          if txQueue.empty() then
70:              SETSTATE(SLEEP)
71:          else
72:              SETSTATE(WAIT_BEACON)
73:          end if
74:      end if
75:  CASE WAIT_ACK
76:      if eventType is ACK_RECEIVED then
77:          if !txQueue.empty() then
78:              SETSTATE(WAIT_DATA_OVER)
79:              SCHEDULEAT(0, CCA_TIMEOUT)
80:          else
81:              SETSTATE(SLEEP)
82:          end if
83:      end if
84:  CASE WAIT_BEACON
85:      if eventType is BEACON_RECEIVED then
86:          SETSTATE(DATA_CCA)
87:          SCHEDULEAT(0, BEACON_RECEIVED)
88:      else if eventType is WAIT_BEACON_TIMEOUT then
89:          SETSTATE(SLEEP_WAIT_BEACON)
90:          timeNextEvent  $\leftarrow$  simTime() + timeToWakeup
91:          SCHEDULEAT(timeNextEvent, WAKEUP_WAIT_BEACON)
92:      end if
93:  CASE SLEEP_WAIT_BEACON
94:      if eventType is WAKEUP_WAIT_BEACON then
95:          SETSTATE(WAIT_BEACON)
96:      end if

```

do protocolo para o caso de expiração do tempo de espera por um beacon. Na linha 88, é verificado se ocorreu um evento de expiração do tempo de espera por um beacon e então, na linha 89 é definido que o dispositivo entre no estado de adormecimento enquanto espera por um beacon. Nas linhas 90 e 91 é definido o momento e agendado o momento de despertar para continuar a espera por um beacon. Já a segunda situação é encontrada da linha 93 até a 96, onde é demonstrado o comportamento que o dispositivo utilizando o protocolo EERIMAC tem quando está no estado SLEEP_WAIT_BEACON. Na linha 94, é verificado se o dispositivo recebeu um evento de despertar e caso se confirme, o dispositivo volta para o estado de espera por um beacon na linha 95.

Algoritmo 9 Algoritmo que implementa a forma que o EERIMAC processa os pacotes gerados pelo dispositivo.

```

1: macFrame ← newMacPacket()
2: ENCAPSULATEPACKAGE(macFrame, dataPackage)
3: SETSOURCE(macFrame, SELF_MAC_ADDRESS)
4: SETDESTINATION(macFrame, destination)
5: PUSHToTXQUEUE(macFrame)
6: if macState is SLEEP then
7:   timeNextEvent ← simTime() + sleepInterval * 1.5
8:   SCHEDULEAT(timeNextEvent, WAIT_BEACON_TIMEOUT)
9:   SETSTATE(WAIT_BEACON)
10: end if

```

O tratamento que o protocolo EERIMAC faz dos pacotes gerados pelo dispositivo e que chegam pela camada de rede até a camada de enlace é apresentado no Algoritmo 9. O que diferencia o método semelhante a este que diz respeito ao RIMAC, apresentado no Algoritmo 6, é que o método implementado para o EERIMAC, conforme pode ser verificado na linha 8, agenda um evento que determina a expiração do tempo de espera por um beacon. O protocolo SWPTMAC faz o mesmo tratamento do protocolo EERIMAC com os pacotes gerados pelo dispositivo e que chegam pela camada de rede até a camada de enlace.

5.4 Códigos usados pelo SWPTMAC

Nesta seção, são apresentados os métodos da implementação do SWPTMAC. Os algoritmos apresentados são 10 e 11. Os algoritmos implementados para o SWPTMAC são baseados nos algoritmos implementados para o EERIMAC. As mudanças realizadas no SWPTMAC em relação ao EERIMAC são para possibilitar que TSFE atue de maneira conjunta com a programação sono-vigília.

Algoritmo 10 Os estados do protocolo SWPTMAC e o estado de rádio correspondente.

```
1: newState ← GETNEWSTATE()
2: SWITCH newState DO
3:   CASE SLEEP
4:     SETRADIOToSLEEP
5:   CASE SLEEP_WAIT_BEACON
6:     SETRADIOToSLEEP
7:   CASE RECEIVING_ENERGY
8:     SETRADIOToSLEEP
9:   CASE TEST_ENERGY_LEVEL
10:    SETRADIOToSLEEP
11:  CASE WAIT_BEACON_OVER
12:    SETRADIOToTX
13:  CASE WAIT_DATA_OVER
14:    SETRADIOToTX
15:  CASE WAIT_DATA_OVER
16:    SETRADIOToTX
17:  CASE BEACON_CCA
18:    SETRADIOToRX
19:  CASE DATA_CCA
20:    SETRADIOToRX
21:  CASE WAIT_DATA
22:    SETRADIOToRX
23:  CASE WAIT_DATA
24:    SETRADIOToRX
25:  CASE WAIT_ACK
26:    SETRADIOToRX
```

O Algoritmo 10 apresenta o método implementado para definir os estados do protocolo SWPTMAC. O protocolo SWPTMAC possui todos os estados dos protocolos RIMAC e EERIMAC. Além dos estados do RIMAC e do EERIMAC, o protocolo SWPTMAC possui os estados necessários para utilização da TSFE que são: RECEIVING_ENERGY e TEST_ENERGY_LEVEL. Tanto no estado RECEIVING_ENERGY quanto no TEST_ENERGY_LEVEL o dispositivo entra no estado de adormecimento.

Algoritmo 11 Código dos comportamentos do protocolo SWPTMAC mediante cada evento recebido.

```

1: eventType ← GETEVENTTYPE()
2: if eventType is RECEIVED_ENERGY then
3:   energyArrived ← false
4:   timeNextEvent ← simTime()
5:   SCHEDULEAT(timeNextEvent, WAKEUP)
6: else if eventType is WAKEUP then
7:   timeNextEvent ← simTime() + sleepInterval * (0.5 + randomNum())
8:   SCHEDULEAT(timeNextEvent, WAKEUP)
9: end if
10: SWITCH macState DO
11:   CASE SLEEP
12:     if eventType is WAKEUP then
13:       SETSTATE(BEACON_CCA)
14:       timeNextEvent ← simTime() + ccaInterval * (1 + randomNum())
15:       SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
16:     else if eventType is RECEIVED_ENERGY then
17:       SETSTATE(RECEIVING_ENERGY)
18:       timeNextEvent ← simTime()
19:       SCHEDULEAT(timeNextEvent, RECEIVE_ENERGY)
20:     end if
21:   CASE BEACON_CCA
22:     if eventType is CCA_TIMEOUT then
23:       if !isChannelClear() then
24:         timeNextEvent ← simTime() + ccaInterval * (1 + randomNum())
25:         SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
26:       else
27:         SENDBEACON(0, BROADCAST_CAST)
28:       end if
29:     end if

```

O método que trata os eventos do protocolo SWPTMAC é apresentado no Algoritmo 11. Os eventos do SWPTMAC são os necessários para um perfeito funcionamento do fluxo de estados apresentados nas máquinas de estados 4.2(a) e 4.2(b). Os eventos do tipo RECEIVE_ENERGY marcam o início do recebimento de energia e os eventos RECEIVED_ENERGY marcam o fim do recebimento de energia. Além disso, três situações

```

30:  CASE DATA_CCA
31:      if eventType is BEACON_RECEIVED then
32:          if !isChannelClear() then
33:              timeNextEvent  $\leftarrow$  simTime() + ccaInterval * (1 + randomNum())
34:              SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
35:          else
36:              SETSTATE(WAIT_DATA_OVER)
37:              timeNextEvent  $\leftarrow$  simTime()
38:              SCHEDULEAT(timeNextEvent, CCA_TIMEOUT)
39:          end if
40:      end if
41:  CASE WAIT_BEACON_OVER
42:      if eventType is TRANSMISSION_ENDED then
43:          SETSTATE(WAIT_DATA)
44:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
45:          SCHEDULEAT(timeNextEvent, BEACON_TIMEOUT)
46:      end if
47:  CASE WAIT_DATA_OVER
48:      if eventType is TRANSMISSION_ENDED then
49:          SETSTATE(WAIT_ACK)
50:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
51:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
52:      else if eventType is CCA_TIMEOUT then
53:          if !txQueue.empty() then
54:              SENDFIRSTPACKAGEINQUEUE()
55:              timeNextEvent  $\leftarrow$  simTime() + timeToTX
56:              SCHEDULEAT(timeNextEvent, TRANSMISSION_ENDED)
57:          end if
58:      else
59:          SETSTATE(WAIT_ACK)
60:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
61:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
62:      end if

```

```

63:  CASE WAIT_ACK_OVER
64:      if eventType is TRANSMISSION_ENDED then
65:          SETSTATE(WAIT_DATA)
66:          timeNextEvent  $\leftarrow$  simTime() + dwellInterval
67:          SCHEDULEAT(timeNextEvent, DWELL_TIMEOUT)
68:      end if
69:  CASE WAIT_DATA
70:      if eventType is BEACON_TIMEOUT then
71:          if txQueue.empty() then
72:              SETSTATE(SLEEP)
73:          else
74:              SETSTATE(WAIT_BEACON)
75:          end if
76:      else if eventType is DWELL_TIMEOUT then
77:          if txQueue.empty() then
78:              SETSTATE(SLEEP)
79:          else
80:              SETSTATE(WAIT_BEACON)
81:          end if
82:      end if
83:  CASE WAIT_ACK
84:      if eventType is ACK_RECEIVED then
85:          if !txQueue.empty() then
86:              SETSTATE(WAIT_DATA_OVER)
87:              SCHEDULEAT(0, CCA_TIMEOUT)
88:          else
89:              SETSTATE(SLEEP)
90:          end if
91:      end if

```

```

92:  CASE WAIT_BEACON
93:    if eventType is BEACON_RECEIVED then
94:      SETSTATE(TEST_ENERGY_LEVEL)
95:      SCHEDULEAT(0, BEACON_RECEIVED)
96:    else if eventType is WAIT_BEACON_TIMEOUT then
97:      SETSTATE(SLEEP_WAIT_BEACON)
98:      timeNextEvent  $\leftarrow$  simTime() + timeToWakeUp
99:      SCHEDULEAT(timeNextEvent, WAKEUP_WAIT_BEACON)
100:   else if eventType is RECEIVE_ENERGY then
101:     SETSTATE(RECEIVING_ENERGY)
102:     timeNextEvent  $\leftarrow$  simTime() + timeToReceiveEnergy    ▷ Calcula o
tempo de recebimento de energia
103:     SCHEDULEAT(timeNextEvent, RECEIVED_ENERGY)
104:   end if
105:  CASE SLEEP_WAIT_BEACON
106:    if eventType is WAKEUP_WAIT_BEACON then
107:      SETSTATE(WAIT_BEACON)
108:    else if eventType is RECEIVE_ENERGY then
109:      SETSTATE(RECEIVING_ENERGY)
110:      timeNextEvent  $\leftarrow$  simTime() + timeToReceiveEnergy
111:      SCHEDULEAT(timeNextEvent, RECEIVED_ENERGY)
112:    end if
113:  CASE TEST_ENERGY_LEVEL
114:    if eventType is BEACON_RECEIVED then
115:      energyLevel  $\leftarrow$  getEnergyLevel()
116:      SETENERGYLEVELTOPACKAGES(energyLevel)
117:      SETSTATE(DATA_CCA)
118:      distance  $\leftarrow$  CALCULATEDISTANCETONODE()
119:      SCHEDULEAT(timeNextEvent, BEACON_RECEIVED)
120:    end if
121:  CASE RECEIVING_ENERGY
122:    if RECEIVED_ENERGY then
123:      SETSTATE(SLEEP)
124:    end if

```

caracterizam o SWPTMAC. A primeira situação é apresentada entre as linhas 2 e 5 onde ocorre a confirmação do recebimento de energia. A linha 2 verifica se teve um recebimento de energia. Na linha 3, define a variável que confirma que o dispositivo não está mais recebendo energia. Nas linhas 4 e 5, ocorre o agendamento de um evento de despertar.

A segunda situação é marcada pelos processamentos realizados pelos dispositivos quando estão no estado `TEST_ENERGY_LEVEL`, conforme pode ser visto entre as linhas 113 e 120. Na linha 114, verifica se o dispositivo recebeu um beacon. Na linha 115, é recuperado o nível de energia do dispositivo e, na linha 116, este nível é adicionado aos dados que serão transmitidos. Na linha 117, o dispositivo é colocado no estado de verificação de canal livre (`DATA_CCA`). Então, o processo de agendamento de um evento é realizado nas linhas 118 e 119.

A terceira situação é quando o dispositivo está recebendo energia, ou seja, no estado `RECEIVING_ENERGY`, conforme pode ser verificado da linha 121 até a linha 124. É verificado se ocorreu o fim do recebimento de energia na linha 122. Caso seja confirmado o fim do recebimento, o dispositivo vai para o estado de adormecimento na linha 123.

5.5 Códigos da implementação do drone

Nesta seção, é apresentado o método que implementa o comportamento do drone. O Algoritmo 12 apresenta o código deste método. Na linha 1, é recuperado o evento recebido pelo drone. A partir da linha 2 inicia-se uma seção de escolha (*SWITCH*) que se estende pelo resto do algoritmo. Entre as linhas 3 e 19, é processado um evento de requisição de energia. Já entre as linhas 20 e 23 é processada uma desocupação do drone.

O processamento da requisição de energia inicia-se na linha 4 verificando se o drone está ocupado levando energia para algum outro dispositivo. Caso o drone esteja ocupado levando energia para algum dispositivo, é recuperada a identificação do dispositivo que solicitou energia. Então, é calculado o tempo que ainda falta para o drone se desocupar na linha 6 e essa informação é enviada, na linha 7, para o dispositivo que solicitou energia.

Caso o drone esteja desocupado, calcula-se nas linhas 9, 10, 11, 12, 13 e 14 a distância, o tempo de viagem, a energia necessária para a viagem, a energia que está armazenada no drone, a energia que será transferida e o tempo de transferência de energia, respectivamente. Então, na linha 15, é calculado o tempo em que o drone estará indisponível, viajando e transferindo energia. Na linha 16, agenda o evento de desocupação do drone. Na linha 17, é enviada uma mensagem para o dispositivo que solicitou energia informando a quantidade de energia que está sendo enviada. Por fim, a variável que informa que o drone está ocupado é modificada para verdadeiro na linha 18.

O outro evento processado pelo drone é a desocupação do mesmo. Quando este evento ocorre, as estatísticas de viagens do drone são atualizadas. Na linha 21, é atualizada a estatística do número total de viagens realizadas pelo drone. Na linha 22, o tempo total que o drone passou viajando também é atualizado. Por fim, na linha 23, a variável que informa que o drone está ocupado é modificada para falsa e o drone fica disponível para transferir energia para algum outro dispositivo que solicitar.

Algoritmo 12 Código do comportamento do drone.

```

1: eventType ← GETEVENTTYPE()
2: SWITCH eventType DO
3:   CASE REQUEST_ENERGY
4:     if occupied then
5:       nodeNumber ← GETNODENUMBER()
6:       timeToDisoccupie ← GETTIMETODESOCUPIE()
7:       SEND(Message(timeToDisoccupie, DRONE_IS_OCCUPIED), "to-
Node", nodeNumber)
8:     else
9:       distance ← CALCULATEDISTANCETONODE()
10:      travelTime ← CALCULATETRAVELTIMETONODE(distance)
11:      energyToTravel ← CALCULATEENERGYTOTRAVEL(travelTime)
12:      energyLevel ← GETDRONEENERGYLEVEL()
13:      energyToTransfer ← energyLevel – energyToTravel
14:      transferTime ← CALCULATETIMETOTRANSFER(energyToTransfer)
15:      timeNextEvent ← simTime() + travelTime + transferTime
16:      SCHEDULEAT(timeNextEvent, DRONE_OCCUPIED_TIMEOUT)
17:      SEND(Message(energyToTransfer, RECEIVE_ENERGY), "toNode", no-
deNumber)
18:      occupied ← true
19:    end if
20:   CASE DRONE_OCCUPIED_TIMEOUT
21:     totalTravels ← totalTravels + 1
22:     totalTravelsTime ← totalTravelsTime + travelTime
23:     occupied ← false

```

5.6 Códigos da implementação do pivô central

Nesta seção, são apresentados os métodos que implementam o comportamento do pivô central. Foi implementado um módulo chamado CircleMobilityManager que caracteriza o movimento em círculo de um dispositivo. O módulo é uma implementação do MobilityManager que faz parte do framework do simulador adotado. Os movimentos circulares são feitos pelo método apresentado no Algoritmo 13. Os eventos periódicos de movimentação e feita através do método apresentado no Algoritmo 14.

Na linha 1 do Algoritmo 13, é calculada a circunferência que o dispositivo deve percorrer baseado no raio fornecido. Na linha 2, é calculado tempo necessário para percorrer todo caminho dividindo a circunferência calculada pela velocidade de locomoção. Na linha 3, é calculado o ângulo percorrido pelo dispositivo baseado no tempo que o percurso está sendo seguido. Então, na linha 4 é encontrado o ângulo atual do dispositivo somando o ângulo inicial com o ângulo já percorrido. Então, nas linhas 5 e 6 é atualizada a posição do dispositivo baseando-se no ângulo calculado anteriormente.

A movimentação do pivô ocorre de maneira semelhante a movimentação de outros tipos de dispositivos no simulador utilizado. A movimentação é feita através de um evento que foi agendado anteriormente e, após o movimento, um novo evento de movimentação é agendado. Na linha 1 do Algoritmo 14, é adicionado o evento recebido pelo pivô e, na linha 2, é verificado se este evento é de movimentação (*MOBILITY_PERIODIC*). Caso seja um evento de movimentação, na linha 3, é calculado o momento que deve ocorrer outro evento semelhante e, na linha 4, este outro evento é agendado. Na linha 5, é acionado o método apresentado no Algoritmo 13. Por último, o canal de comunicação sem fio é informado sobre a nova posição do dispositivo na linha 6.

Algoritmo 13 Código da movimentação circular dos dispositivos.

```

1:  $circ \leftarrow 2 * 3.14159 * r$ 
2:  $timeCirc \leftarrow circ/speed$ 
3:  $tempAngle \leftarrow 360 * simTime()/timeCirc$ 
4:  $angle \leftarrow startAngle + tempAngle$ 
5:  $nodeLocationX \leftarrow cx + r * cos(angle)$ 
6:  $nodeLocationY \leftarrow cy + r * sin(angle)$ 

```

Algoritmo 14 Código de agendamento e tratamento de eventos de movimentos.

```

1:  $eventType \leftarrow GETEVENTTYPE()$ 
2: if  $eventType$  is MOBILITY_PERIODIC then
3:    $timeNextEvent \leftarrow simTime() + updateInterval$ 
4:    $SCHEDULEAT(timeNextEvent, MOBILITY\_PERIODIC)$ 
5:    $MOVE()$ 
6:    $NOTIFYWIRELESSCHANNEL()$ 
7: end if

```

Capítulo 6

Simulação e Resultados

Neste capítulo será explicada como foi feita a avaliação da arquitetura proposta e dos protocolos em questão. As especificações da máquina utilizada para fazer a simulação é apresentada na Tabela 6.1. Foram feitas de 10 até 30 repetições de cada configuração da simulação. Os resultados apresentados são a média e a barra de erro com intervalo de confiança de 95%.

Tabela 6.1: Especificações da Máquina.

Dispositivo	Notebook
Sistema Operacional	Windows 10
Memória RAM	8
Processador	Intel Core i5
Armazenamento	240 GB SSD
Virtualização	VirtualBox
Sistema Virtualizado	Ubuntu 20.4
Simulador	Omnet 4.6
Framework	Castália

Cada configuração da simulação terá um número atribuído e serão um total de 5 Simulações. As camadas de comunicação utilizadas nas Simulações 1 até 5 são apresentadas na Tabela 6.2. Já as camadas de comunicação utilizadas na Simulação 4 são apresentadas na Tabela 6.3 e as camadas de comunicação da Simulação 5 são apresentadas na Tabela 6.4.

Tabela 6.2: Pilha de Comunicação das Simulações 1 ao 3.

Aplicação que Testa Taxa de Transferência
Apenas Encapsulamento dos Dados na Camada de Rede
Apenas Encapsulamento dos Dados na Camada de Enlace
Rádio CC2420

Tabela 6.3: Pilha de Comunicação da Simulação 4.

Aplicação que Testa Taxa de Transferência
Apenas Encapsulamento dos Dados na Camada de Rede
SWPTMAC
Rádio CC2240

Tabela 6.4: Pilha de Comunicação da Simulação 5.

Aplicação que Testa Taxa de Transferência
Apenas Encapsulamento dos Dados na Camada de Rede
SWPTMAC e Outros Protocolos de Programação Sono-Vigília
Rádio CC2240

A próxima seção apresenta as métricas que serão levadas em consideração nas simulações. As demais seções deste capítulo apresentam explicações sobre as simulações, os parâmetros utilizados em cada simulação e os gráficos contendo a média dos resultados alcançados, bem como as barras de erros apresentando o Intervalo de Confiança de T Student.

6.1 Métricas de Desempenho

- Taxa de Transferência: é uma métrica essencial para avaliar o desempenho de uma rede de comunicação em termos de eficiência de transmissão de dados. De acordo com [61], a taxa de transferência refere-se à quantidade de dados que podem ser transmitidos através da rede em um determinado período de tempo. Dentre os vários fatores que influenciam a taxa de transferência em uma rede, [62] destaca a largura de banda disponível, a qualidade do canal de comunicação, os protocolos de rede, o nível de congestionamento e a capacidade de processamento dos dispositivos como fatores que mais afetam a taxa de transferência. É essencial considerar esses fatores ao projetar e otimizar uma rede de comunicação para garantir uma taxa de

transferência eficiente. Para medir a taxa de transferência em uma rede, são utilizadas diversas abordagens. De acordo com o trabalho [63], testes de desempenho são realizados, nos quais pacotes de teste são enviados entre dois pontos da rede e o tempo necessário para a conclusão da transferência é medido. Além disso, ferramentas de simulação e modelagem, como o simulador Omnet usado nesta pesquisa, são amplamente utilizadas para estimar a taxa de transferência em diferentes cenários de rede.

- Perda de Pacotes: é uma métrica que indica a porcentagem de pacotes de dados perdidos durante a transmissão na rede. Segundo [64], a perda de pacotes ocorre quando dados enviados não são recebidos pelo destino, e essa ocorrência pode ser prejudicial para aplicações sensíveis à latência e à confiabilidade, como videoconferências e transmissão de voz sobre IP. A taxa de perda de pacotes, geralmente expressa como uma porcentagem, é calculada pela divisão dos pacotes perdidos pelo total de pacotes enviados. Segundo [65], a perda de pacotes pode ocorrer devido a vários fatores como colisões e falta de alcance entre os rádios.
- Estimativa de VUR: é uma métrica fundamental para avaliar o desempenho energético dos dispositivos em uma RSSF. De acordo com [66], a estimativa de VUR é calculada por meio de simulação, dividindo a carga de energia remanescente dos dispositivos pelo seu consumo médio destes. Essa métrica fornece uma medida quantitativa da duração esperada de pleno funcionamento da rede. Segundo [67], foi demonstrado que a estimativa de vida útil da rede é afetada por vários fatores, como o consumo de energia dos dispositivos em diferentes estados operacionais (transmitindo e recebendo dados, ocioso, adormecido), as taxas de transmissão de dados e as estratégias de gerenciamento de energia. Portanto, é de suma importância considerar esses fatores ao realizar o cálculo da VUR. Além disso, de acordo com [68] é importante considerar a variabilidade das fontes de energia ao calcular a estimativa de vida útil da rede. No caso do trabalho apresentado nesta dissertação, a fonte de energia era a bateria interna de cada dispositivo e a energia recebida pelos drones.

6.2 Simulação 1

A Simulação 1 busca avaliar o impacto causado pela variação da quantidade de pacotes enviados por segundo de um dispositivo para o outro. As métricas utilizadas na avaliação são a taxa de transferência, a perda de pacotes e a estimativa de VUR. A Tabela 6.5 apresenta os parâmetros utilizados nessa simulação.

Tabela 6.5: Parâmetros da Simulação 1.

Potência de Transmissão (dBm)	0
Sensibilidade (dB)	-95
Número de Dispositivos	2
Distância Entre Dispositivos (m)	15
Cabeçalhos do Pacote (bytes)	29
Corpo do Pacote (bytes)	100
Frequência de Envio (Hz)	100, 200, 300, 400, 500
Tempo de Simulação(s)	100

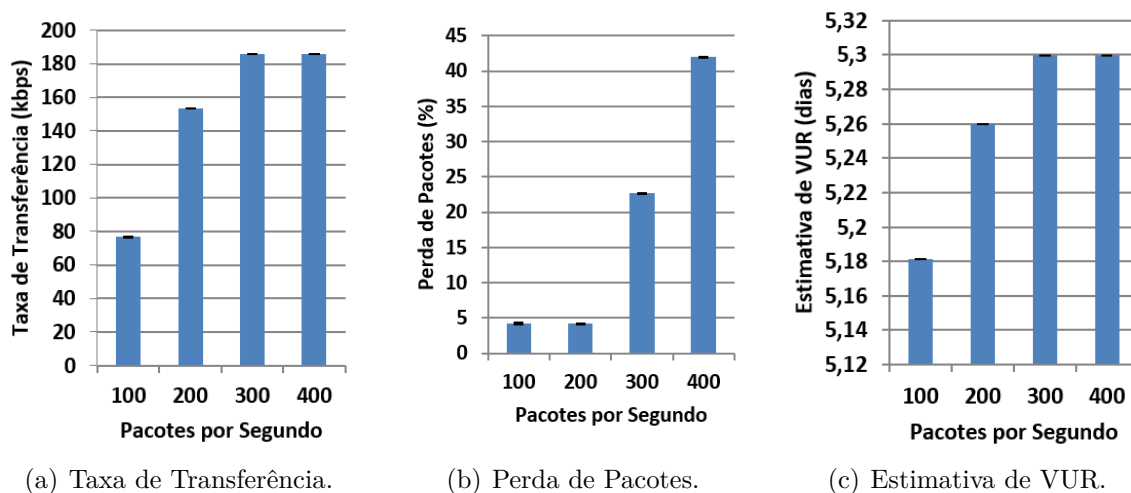


Figura 6.1: Avaliação do Impacto da Quantidade de Pacotes Enviados de Um Dispositivo para Outro em uma RSSFS.

Os resultados alcançados com a primeira simulação são apresentados na Figura 6.1. A Figura 6.1(a) mostra a relação entre a quantidade de pacotes enviados por segundo e a taxa de transferência. A taxa de transferência é calculada multiplicando o tamanho dos pacotes pela quantidade de pacotes transferidos em um determinado período. Sendo assim, à medida que cresce a quantidade de pacotes transferidos, aumenta a taxa de transferência. A taxa de transferência vai crescendo até o limite de capacidade daquela rede. Este limite é determinado pelos softwares, hardwares e meios utilizados na comunicação. Então, os resultados alcançados na simulação demonstram que há um crescimento da taxa de transferência à medida que aumenta a frequência de envio de pacotes até um limite de cerca de 180 kbps.

A Figura 6.1(b) mostra a relação entre a quantidade de pacotes enviados por segundo e a perda de pacotes. A perda de pacotes é a porcentagem de pacotes que foram transmitidos e não chegaram perfeitamente no destino. Algumas das causas para perda de pacotes são:

colisões, falta de alcance entre os rádios ou congestionamento da rede. Tendo em vista que nesta simulação um único dispositivo transfere informações para o outro, descarta-se, portanto, a possibilidade de colisões. Os dispositivos estão em posições fixas, portanto, caso o problema fosse de falta de alcance entre os rádios, a perda de pacotes seria de 100%. Então, pode-se concluir que o motivo para as perdas de pacotes é o congestionamento da rede ao atingir o limite da própria capacidade.

Por fim, a Figura 6.1(c) mostra a relação entre a quantidade de pacotes enviados por segundo e a estimativa de VUR. A estimativa de VUR é dada pela divisão da carga energética dos dispositivos pelo consumo energético destes. A transição entre os estados do rádio (transmitindo, recebendo e adormecido) é um fator de grande consumo energético. Os resultados alcançados, demonstram que em casos de menor frequência de transmissão a estimativa de VUR é menor. O motivo para esta menor estimativa de VUR é o grande número de transições entre estados do rádio do dispositivo, o que aumenta o consumo energético. Quando a frequência de transmissão cresce o dispositivo fica sempre no estado de transmissão e por isso o consumo energético, e conseqüentemente a estimativa de VUR, é constante.

6.3 Simulação 2

Assim como a Simulação 1, a Simulação 2 busca avaliar a taxa de transferência e a perda de pacotes. O fator de variação dessa vez é o tamanho dos pacotes enviados por segundo de um dispositivo para o outro. A Tabela 6.6 apresenta os parâmetros utilizados nessa simulação. Os resultados alcançados com a segunda simulação são apresentados na Figura 6.2.

Tabela 6.6: Parâmetros da Simulação 2.

Potência de Transmissão (dBm)	0
Sensibilidade (dB)	-95
Número de Dispositivos	2
Distância Entre Dispositivos (m)	15
Cabeçalhos do Pacote (bytes)	29
Corpo do Pacote (bytes)	100, 200, 300, 400, 500
Frequência de Envio (Hz)	300
Tempo de Simulação(s)	100

A Figura 6.2(a) mostra que há um crescimento da taxa de transferência à medida que aumenta o tamanho dos pacotes até um limite de cerca de 200 kbps. A taxa de

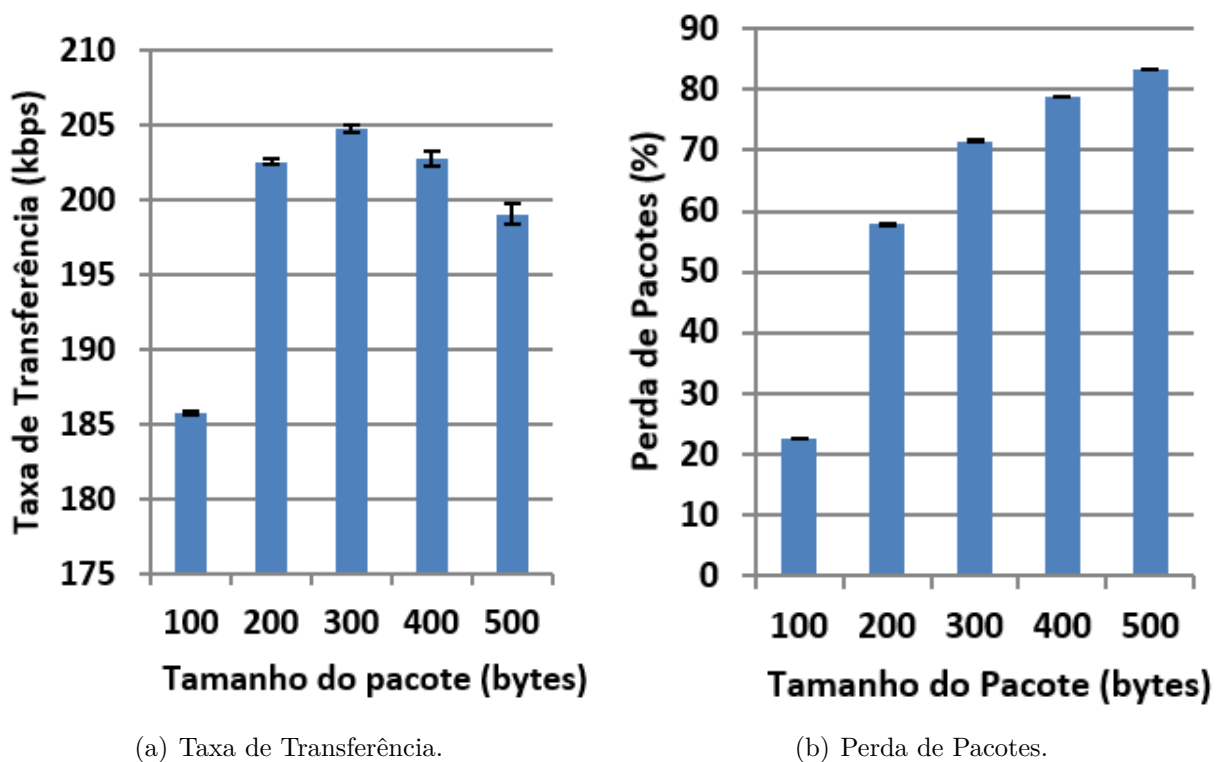


Figura 6.2: Avaliação do Impacto do Tamanho dos Pacotes Enviados de Um Dispositivo para Outro em uma RSSFS.

transferência máxima alcançada aumentando o tamanho do pacote é maior do que a taxa alcançada aumentando a frequência do envio de pacotes. Cada pacote exige um determinado intervalo de tempo para ser processado. O tempo necessário para processar um grande número de pacotes pequenos pode ser maior do que o tempo necessário para processar um número menor de pacotes grandes. No entanto, a soma dos bits de um número pequeno de pacotes grandes podem representar mais bits do que muitos pacotes pequenos. Sendo assim, por isso nesta simulação foi possível atingir uma maior taxa de transferência.

A perda de pacotes da Simulação 2, é apresentado na Figura 6.2(b). Foi relacionada nesta simulação a perda de pacotes ao tamanho dos pacotes transmitidos. Assim como na Simulação 1, nesta simulação a causa da perda de pacotes é o congestionamento da rede. À medida que aumentou o tamanho dos pacotes, a quantidade de pacotes perdidos cresceu.

A estimativa de VUR da rede na Simulação 2 é constante. Um motivo para esta constância é que desde o início a rede estava transmitindo o máximo de dados possíveis. Sendo assim, os estados dos rádios eram constantes e conseqüentemente o consumo energético foi constante. O consumo energético sendo constante, conseqüentemente a estimativa de

VUR também é constante. Outro motivo para a estimativa de VUR ser constante é o fato dos sensores enviarem os dados diretamente para o coletor de dados, ou seja, sem saltos adicionais. O número de saltos é um fator impactante no consumo energético e, portanto, como o número de saltos foi sempre constante, a estimativa de VUR também foi constante.

6.4 Simulação 3

A Simulação 3 busca avaliar o impacto causado pelo número de sensores na RSSFS. Nesta simulação os sensores enterrados são dispostos em grades e o coletor de dados é instalado em um braço de um pivô central. A Tabela 6.7 apresenta os parâmetros utilizados nessa simulação. Os resultados alcançados com a terceira simulação são apresentados na Figura 6.3. Os resultados apresentados são as médias da taxa de transferência e a perda de pacotes.

Tabela 6.7: Parâmetros da Simulação 3.

Potência de Transmissão (dBm)	0
Sensibilidade (dB)	-95
Cabeçalhos do Pacote (bytes)	29
Corpo do Pacote (bytes)	100
Frequência de Envio (Hz)	300
Quantidade de Sensores	9, 16, 25, 36, 49
Área da Simulação (m^2)	900
Tempo de Simulação(s)	100

A Figura 6.3(a) mostra a avaliação das taxas de transferências. O resultado apresentado é a média da taxa de transferência dos dispositivos. À medida que cresce o número de sensores a taxa de transferência média vai diminuindo. O motivo desta diminuição da taxa de transferência é que os dispositivos acabam concorrendo pelo mesmo canal. Consequentemente, à medida que aumenta o número de sensores o tempo de uso do canal é dividido entre eles.

Os resultados apresentados na Figura 6.3(b) demonstram um crescimento médio na perda de pacotes à medida que aumenta o número de sensores dispostos no campo. Quando 2 ou mais dispositivos próximos tentam transmitir informações ao mesmo tempo, usando o mesmo canal, pode ocorrer uma interferência dos sinais. Essas interferências são colisões que provocam perda de informações. Sendo então, as colisões a principal causa para as perdas de pacotes neste caso.

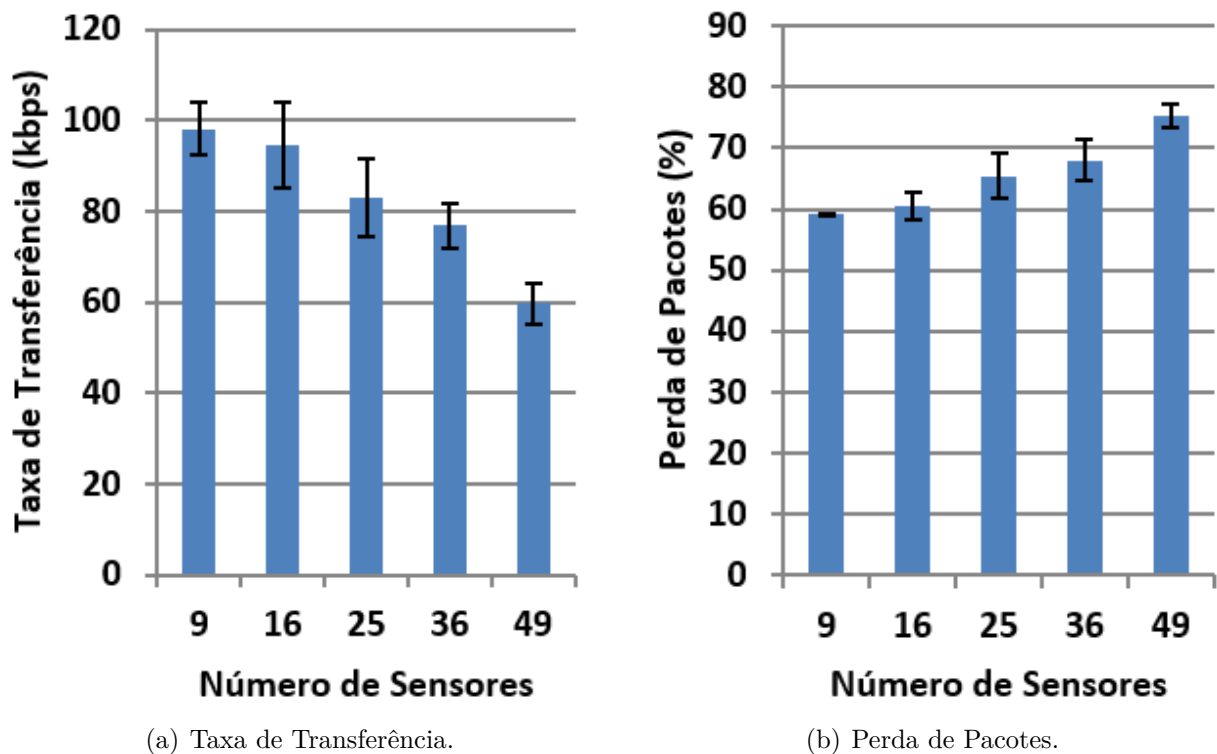


Figura 6.3: Avaliação do Impacto da Quantidade de Sensores em uma RSSFS.

Assim como na Simulação 2, na Simulação 3 o fato dos sensores enviarem os dados diretamente para o coletor de dados, sem saltos adicionais, fez a estimativa de VUR ser constante. O número de saltos é um fator impactante no consumo energético e, portanto, como o número de saltos foi sempre constante, a estimativa de VUR também foi constante. Outra questão determinante para a constância da estimativa de VUR é o fato da quantidade de pacotes transmitida ser constante.

6.5 Simulação 4

A Simulação 4 busca avaliar qual a quantidade mínima de drones que uma rede utilizando o SWPTMAC nas condições avaliadas necessita. O fator de variação dessa vez é o número de drones. A Tabela 6.8 apresenta os parâmetros utilizados nessa simulação. Os resultados alcançados com a quarta simulação são apresentados na Figura 6.4.

Tabela 6.8: Parâmetros da Simulação 4.

Potência de Transmissão (dBm)	0
Sensibilidade (dB)	-95
Cabeçalhos do Pacote (bytes)	29
Corpo do Pacote (bytes)	100
Intervalo Entre o Envio de Pacotes (min)	60
Energia Inicial (J)	30.000
Quantidade de Sensores	9
Quantidade de Drones	1, 2, 3, 4
Área da Simulação (m^2)	900
Tempo de Simulação(horas)	2

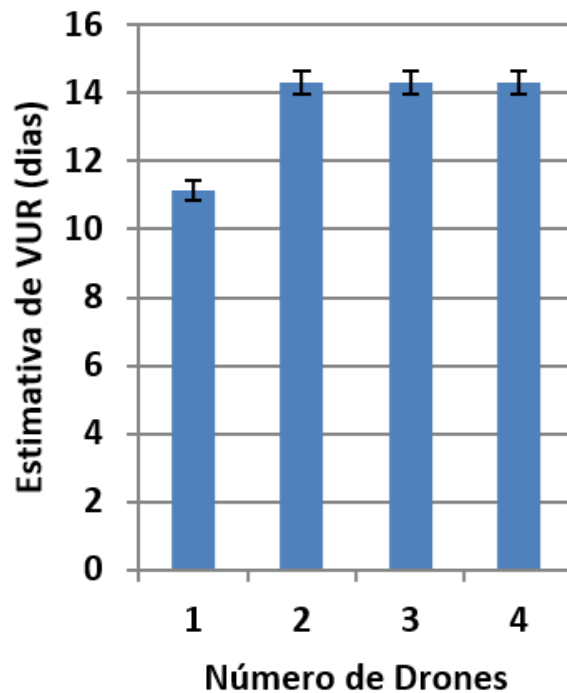


Figura 6.4: Avaliação do Impacto da Quantidade de Drones em uma RSSFS Usando o Protocolo SWPTMAC.

No cenário proposto, os resultados alcançados demonstram que aumentando de um drone para dois, há um aumento na estimativa de VUR. No entanto, mais de dois drones não ofereceram ganho na estimativa de VUR. O número de sensores, o tempo de simulação, a distância dos sensores para a base dos drones, o caminho escolhido para o drone percorrer e o nível energético escolhido para o dispositivo começar a receber energia impactam no

número de drones que possibilitam a melhor estimativa de VUR. O número de drones deve ser ajustado para cada cenário que desejar avaliar.

6.6 Simulação 5

A Simulação 5 busca avaliar a estimativa de VUR da RSSFS simulada. Serão avaliados os protocolos BypassMAC (que faz apenas o encaminhamento de pacotes de uma camada para outra), RIMAC, EERIMAC e SWPTMAC. Os parâmetros são iguais aos utilizados nas simulações de validação da arquitetura de RSSFS na agricultura. O número de drones utilizados é baseado no resultado que foi alcançado na Simulação 4. O intervalo de geração de pacote de hora em hora é o mesmo da estação agrometeorológica de Cadriano [69]. A Tabela 6.9 apresenta os parâmetros utilizados nessa simulação.

Tabela 6.9: Parâmetros da Simulação 5.

Potência de Transmissão (dBm)	0
Sensibilidade (dB)	-95
Cabeçalhos do Pacote (bytes)	29
Corpo do Pacote (bytes)	100
Intervalo Entre o Envio de Pacotes (min)	60
Energia Inicial (J)	30.000
Quantidade de Sensores	9
Quantidade de Drones	2
Área da Simulação (m^2)	900
Tempo de Simulação(horas)	2

Os resultados alcançados com a quinta simulação são apresentados na Figura 6.5. De acordo com [28], os protocolos RIMAC e EERIMAC apresentam uma taxa de 100% de entrega de pacotes, o que foi confirmado pela pesquisa apresentada nesta dissertação. Além disso, o protocolo SWPTMAC também teve uma taxa de 100% de entrega de pacotes. Sendo assim, a métrica utilizada na avaliação é a estimativa de VUR.

Os resultados alcançados demonstram que o protocolo BypassMAC, que não utiliza nenhuma técnica de maximização de VUR, apresenta a menor estimativa de VUR. Os resultados alcançados demonstram que o protocolo RIMAC, que utiliza a técnica de programação sono-vigília, apresenta um ganho na estimativa de VUR. No entanto, o protocolo EERIMAC, utiliza a programação sono-vigília com um mecanismo de colocar o dispositivo em adormecimento após um período esperando o beacon, apresenta uma maior estimativa de VUR do que o RIMAC. Já o protocolo SWPTMAC proposto nessa dissertação

apresenta uma estimativa de VUR maior do que os demais protocolos testados. Essa maior estimativa de VUR acontece porque, além de utilizar a programação sono-vigília e o mesmo mecanismo do EERIMAC, ele integra a técnica TSFE.

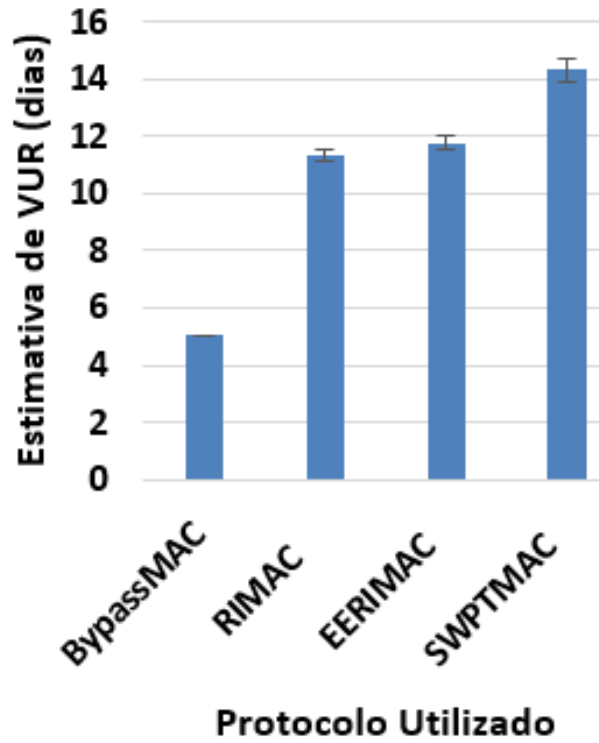


Figura 6.5: Comparação da Estimativa de VUR Alcançada com Cada Protocolo.

Capítulo 7

Considerações Finais

O monitoramento subterrâneo é importante para verificar a água e conteúdo mineral do solo, integridade das infraestruturas, deslizamentos de terra e ocorrências de terremotos [8], sendo útil para áreas que trabalham com tubulações, infraestrutura e agricultura [5][6][7]. As Redes de Sensores Sem Fio Subterrâneos (RSSFS) podem ser utilizadas para realizarem esse monitoramento. As RSSFS são redes formadas por sensores que ficam enterrados e utilizam conexão sem fio para se comunicarem. Ainda que as RSSFS sejam fundamentais para o monitoramento subterrâneo, estas redes possuem os recursos energéticos limitados, o que acaba diminuindo a Vida Útil da Rede (VUR).

Foi feita uma pesquisa exploratória sobre as RSSFS. Foram pesquisadas as áreas de aplicação dessas redes e os desafios a serem solucionados. Além disso, foi feita uma busca em relação aos trabalhos que buscam solucionar o problema de maximização da VUR das RSSFS. Na busca bibliográfica, não foram encontradas soluções que utilizam em conjunto as técnicas de programação sono-vigília e de TSFE. Sendo assim, a solução apresentada neste documento é um protocolo, chamado SWPTMAC, que utiliza programação sono-vigília e é adaptado para utilização de TSFE para a maximização da VUR de RSSFS.

Na sequência, foi feita via simulação a caracterização de uma RSSFS aplicada à agricultura. Na RSSFS caracterizada, os sensores enterrados coletam informações do solo e enviam através de rádios de ondas eletromagnéticas para o coletor de dados que fica instalado no braço de um pivô central. Já a TSFE foi caracterizada para ser feita através de um drone que vai até próximo aos sensores enterrados e transfere a energia. Então, foi implementado o protocolo SWPTMAC baseando-o no protocolo EERIMAC. Comparando o cenário do protocolo apresentado com o estado da arte, verificou-se um ganho de 24% na estimativa de VUR.

Portanto, tendo em vista os resultados alcançados, conclui-se que a solução alcançou ganho em relação ao estado da arte. Como trabalhos futuros, devem ser feitas avaliações de outros cenários da solução proposta. A proposta deve ser avaliada em outras áreas além

da agricultura como, por exemplo, caracterizando uma região onde ocorreram desastres geológicos. Outro caminho de pesquisa futura é de otimização das viagens dos drones, bem como trabalhar a conexão do drone na mesma rede dos sensores subterrâneos para receber informações sobre o nível energético destes sensores. Além disso, para uma completa validação, deve ser feita uma implementação física ou um modelo matemático da solução apresentada para serem comparados com os resultados alcançados com a simulação.

Referências

- [1] Unnikrishna Menon, K. A., Vippalapalli Vikas e Balaji Hariharan: *Wireless power transfer to underground sensors using resonant magnetic induction*. Em *2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN)*, páginas 1–5, 2013. x, 10, 11
- [2] Sharma, Priyanka, Rishi Pal Singh, Mazin Abed Mohammed, Rachna Shah e Jan Nedoma: *A survey on holes problem in wireless underground sensor networks*. IEEE Access, 10:7852–7880, 2022. 1, 6
- [3] Sun, Zhi e Ian F. Akyildiz: *Connectivity in wireless underground sensor networks*. Em *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, páginas 1–9, 2010. 1, 11
- [4] Sun, Zhi, Ian F. Akyildiz e Gerhard P. Hancke: *Dynamic connectivity in wireless underground sensor networks*. IEEE Transactions on Wireless Communications, 10(12):4334–4344, 2011. 1, 11
- [5] Lin, Kaiqiang e Tong Hao: *Link quality analysis of wireless sensor networks for underground infrastructure monitoring: A non-backfilled scenario*. IEEE Sensors Journal, 21(5):7006–7014, 2021. 1, 11, 12, 64
- [6] Lv, Haibin, Dongzhi Wang e Hui Ming: *Simulation test of wireless underground sensor network in stadiums*. Journal of Sensors, 2021, ISSN 1687-725X. <https://doi.org/10.1155/2021/6827961>. 1, 11, 12, 64
- [7] Dong, Xin, Mehmet C. Vuran e Suat Irmak: *Autonomous precision agriculture through integration of wireless underground sensor networks with center pivot irrigation systems*. Ad Hoc Networks, 11(7):1975–1987, 2013, ISSN 1570-8705. <https://www.sciencedirect.com/science/article/pii/S1570870512001291>, Theory, Algorithms and Applications of Wireless Networked Robotics Recent Advances in Vehicular Communications and Networking. 1, 11, 12, 28, 64
- [8] Akyildiz, Ian F. e Erich P. Stuntebeck: *Wireless underground sensor networks: Research challenges*. Ad Hoc Networks, 4(6):669–686, 2006, ISSN 1570-8705. <https://www.sciencedirect.com/science/article/pii/S1570870506000230>. 1, 2, 11, 15, 64
- [9] Lin, Kaiqiang e Tong Hao: *Experimental link quality analysis for lora-based wireless underground sensor networks*. IEEE Internet of Things Journal, 8(8):6565–6577, 2021. 1

- [10] Bhardwaj, M., T. Garnett e A.P. Chandrakasan: *Upper bounds on the lifetime of sensor networks*. Em *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, volume 3, páginas 785–790 vol.3, 2001. 1
- [11] Kisseleff, Steven, Xiaoyang Chen, Ian F. Akyildiz e Wolfgang H. Gerstacker: *Efficient charging of access limited wireless underground sensor networks*. *IEEE Transactions on Communications*, 64(5):2130–2142, 2016. 1, 15, 16
- [12] Kisseleff, S., X. Chen, I. F. Akyildiz e W. Gerstacker: *Wireless power transfer for access limited wireless underground sensor networks*. Em *2016 IEEE International Conference on Communications (ICC)*, páginas 1–7, 2016. 1, 15, 16
- [13] Deekshith, P K, Trupthi Chougule, Shreya Turmari, Ramya Raju, Rakshitha Ram e Vinod Sharma: *Transmitting information and harvested energy over a fading mac with minimum rate constraints*. Em *2017 International Symposium on Wireless Communication Systems (ISWCS)*, páginas 78–83, 2017. 1, 16
- [14] Silva, Agnelo R. e Mahta Moghaddam: *Design and implementation of low-power and mid-range magnetic-induction-based wireless underground sensor networks*. *IEEE Transactions on Instrumentation and Measurement*, 65(4):821–835, 2016. 2, 15, 16
- [15] Liu, Guanghua, Zehua Wang e Tao Jiang: *Qos-aware throughput maximization in wireless powered underground sensor networks*. *IEEE Transactions on Communications*, 64(11):4776–4789, 2016. 2, 16
- [16] Liu, Guanghua, Zhi Sun e Tao Jiang: *Joint time and energy allocation for qos-aware throughput maximization in mimo-based wireless powered underground sensor networks*. *IEEE Transactions on Communications*, 67(2):1400–1412, 2019. 2, 16
- [17] Sun, Yanjun, Omer Gurewitz e David B. Johnson: *Ri-mac: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks*. *SenSys '08*, página 1–14, New York, NY, USA, 2008. Association for Computing Machinery, ISBN 9781595939906. <https://doi.org/10.1145/1460412.1460414>. 2, 8, 26
- [18] Fourati, Lamia, Sarrah El-Kaffel, Adel Ben Mnaouer e Farid Touati: *Study of nature inspired power-aware wake-up scheduling mechanisms in wsn*. Em *2020 International Wireless Communications and Mobile Computing (IWCMC)*, páginas 2154–2159, 2020. 2, 7, 17, 26
- [19] Kour, Vippon Preet e Sakshi Arora: *Recent developments of the internet of things in agriculture: A survey*. *IEEE Access*, 8:129924–129957, 2020. 2, 5, 28
- [20] Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari e Moussa Ayyash: *Internet of things: A survey on enabling technologies, protocols, and applications*. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, 2015. 4

- [21] Panatik, Kamarul Zaman, Kamilia Kamardin, Sya Azmeela Shariff, Siti Sophiayati Yuhaniz, Noor Azurati Ahmad, Othman Mohd Yusop e SaifulAdli Ismail: *Energy harvesting in wireless sensor networks: A survey*. Em *2016 IEEE 3rd International Symposium on Telecommunication Technologies (ISTT)*, páginas 53–58, 2016. 6
- [22] Borges, Luís M., Fernando J. Velez e António S. Lebres: *Survey on the characterization and classification of wireless sensor network applications*. *IEEE Communications Surveys Tutorials*, 16(4):1860–1890, 2014. 6
- [23] Dubey, Kumkum, Prashant Kumar, Pragati Pathak, Shobhit Kumar, Prince Rajpoot e Shivendra Kumar Pandey: *A survey on fundamental of wireless sensor network with various issues in optimization*. Em *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, páginas 288–293, 2019. 6
- [24] Samara, Ghassan: *Wireless sensor network mac energy - efficiency protocols: A survey*. Em *2020 21st International Arab Conference on Information Technology (ACIT)*, páginas 1–5, 2020. 6
- [25] Akyildiz, I.F. e M.C. Vuran: *Wireless Sensor Networks*. *Advanced Texts in Communications and Networking*. Wiley, 2010, ISBN 9780470515198. <https://books.google.com.br/books?id=7YBHYJsSmS8C>. 7
- [26] Shi, Lei, Jianghong Han, Dong Han, Xu Ding e Zhenchun Wei: *The dynamic routing algorithm for renewable wireless sensor networks with wireless power transfer*. *Computer Networks*, 74:34–52, 2014, ISSN 1389-1286. <https://www.sciencedirect.com/science/article/pii/S1389128614003089>. 8
- [27] Huang, Jun, Yide Zhou, Zhaolong Ning e Hamid Gharavi: *Wireless power transfer and energy harvesting: Current status and future prospects*. *IEEE Wireless Communications*, 26(4):163–169, 2019. 8, 10, 26
- [28] Yong, Yueh Tiam, Chee Onn Chow, Jeevan Kanesan e Hiroshi Ishii: *Ee-ri-mac: An energy-efficient receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks*. *International journal of physical sciences*, 6:2633–2643, junho 2011. 8, 9, 62
- [29] Goudjil, Lakhdar e Fouzi Semchedine: *Cosfi-rimac: A cooperative short frame identifier receiver initiated mac protocol for wireless sensor network*. *Wireless Personal Communications*, páginas 1–17, setembro 2022. 8
- [30] Bayrakdar, Muhammed Enes: *A smart insect pest detection technique with qualified underground wireless sensor nodes for precision agriculture*. *IEEE Sensors Journal*, 19(22):10892–10897, 2019. 12
- [31] Wohwe Sambo, Damien, Anna Forster, Blaise Omer Yenke, Idrissa Sarr, Bamba Gueye e Paul Dayang: *Wireless underground sensor networks path loss model for precision agriculture (wusn-plm)*. *IEEE Sensors Journal*, 20(10):5298–5313, 2020. 13
- [32] Semnani, Abbas, Badri Baskaran e Dimitrios Peroulis: *Microwave wireless powering of sensed agricultural tile drainages*. *IEEE Transactions on Antennas and Propagation*, 69(5):2913–2920, 2021. 13

- [33] Salam, Abdul: *Design of subsurface phased array antennas for digital agriculture applications*. Em *2019 IEEE International Symposium on Phased Array System Technology (PAST)*, páginas 1–5, 2019. 13
- [34] Ghosh, Debjani, Akash Anand, Satya Sankalp Gautam e Ankit Vidyarthi: *Soil fertility monitoring with internet of underground things: A survey*. *IEEE Micro*, páginas 1–1, 2021. 13
- [35] Lin, Kaiqiang e Tong Hao: *Adaptive selection of transmission configuration for lora-based wireless underground sensor networks*. Em *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, páginas 1–6, 2021. 14
- [36] Sahota, Herman e Ratnesh Kumar: *Maximum-likelihood sensor node localization using received signal strength in multimedia with multipath characteristics*. *IEEE Systems Journal*, 12(1):506–515, 2018. 14
- [37] Salam, Abdul, Mehmet C. Vuran e Suat Irmak: *A statistical impulse response model based on empirical characterization of wireless underground channels*. *IEEE Transactions on Wireless Communications*, 19(9):5966–5981, 2020. 14
- [38] Bayrakdar, Muhammed Enes: *A smart insect pest detection technique with qualified underground wireless sensor nodes for precision agriculture*. *IEEE Sensors Journal*, 19, 2019, ISSN 15581748. 14
- [39] Sambo, Damien Wohwe, Anna Forster, Blaise Omer Yenke, Idrissa Sarr, Bamba Gueye e Paul Dayang: *Wireless underground sensor networks path loss model for precision agriculture (wusn-plm)*. *IEEE Sensors Journal*, 20, 2020, ISSN 15581748. 14
- [40] Yang, Sijung, Omar Baltaji, Andrew C. Singer e Youssef M. A. Hashash: *Development of an underground through-soil wireless acoustic communication system*. *IEEE Wireless Communications*, 27(1):154–161, 2020. 14
- [41] Di Renzone, Gabriele, Stefano Parrino, Giacomo Peruzzi, Alessandro Pozzebon e Duccio Bertoni: *Lorawan underground to aboveground data transmission performances for different soil compositions*. *IEEE Transactions on Instrumentation and Measurement*, 70:1–13, 2021. 14
- [42] Kisseleff, Steven, Ian F. Akyildiz e Wolfgang H. Gerstacker: *Survey on advances in magnetic induction-based wireless underground sensor networks*. *IEEE Internet of Things Journal*, 5(6):4843–4856, 2018. 15, 16
- [43] Yadav, Sadanand, Vinay Kumar, Sanjay B. Dhok e Dushantha N. K. Jayakody: *Energy-efficient design of mi communication-based 3-d non-conventional wsns*. *IEEE Systems Journal*, 14(2):2585–2588, 2020. 16
- [44] Liu, Guanghua: *Data collection in mi-assisted wireless powered underground sensor networks: Directions, recent advances, and challenges*. *IEEE Communications Magazine*, 59(4):132–138, 2021. 16

- [45] Wang, Kevin I Kai, Shiyang Wu, Ameer Ivoghlian, Zoran Salcic, Andrew Austin e Xiaokang Zhou: *Lws: A lorawan wireless underground sensor network simulator for agriculture applications*. Em *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/S-CALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, páginas 475–482, 2019. 16
- [46] Lin, K., O.L.A. López, H. Alves, D. Chapman, N. Metje, G. Zhao e T. Hao: *Throughput optimization in backscatter-assisted wireless-powered underground sensor networks for smart agriculture*. *Internet of Things (Netherlands)*, 20, 2022. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85142326535&doi=10.1016%2Fj.iot.2022.100637&partnerID=40&md5=dd928c48a4d85408eab9111ba084c4ea>, cited By 0. 18, 19, 20
- [47] Sanchez, J., J.M. Arteaga, C. Zeisiger, D.J. Young, R. Goel, P.D. Mitcheson, E.M. Yeatman e S. Roundy: *Integration of a high frequency inductive power transfer system to energize agricultural sensors through soil*. páginas 366–371, 2022. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85137370094&doi=10.1109%2FWPW54272.2022.9853990&partnerID=40&md5=39209672cd6886fcd20783c7f03afd30>, cited By 0. 18, 19, 20
- [48] Semnani, A., B. Baskaran e D. Peroulis: *Microwave wireless powering of sensed agricultural tile drainages*. *IEEE Transactions on Antennas and Propagation*, 69(5):2913–2920, 2021. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85105563685&doi=10.1109%2FTAP.2020.3030901&partnerID=40&md5=41354cfa827ae302ff8cdb47e82027a2>, cited By 1. 18, 19, 20, 21
- [49] Alshehri, A.A., C.H. Martins, S. C. Lin, I.F. Akyildiz e H.K. Schmidt: *Fractot technology for mapping hydraulic fractures*. *SPE Journal*, 26(2):610–626, 2021. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85108831163&doi=10.2118%2F187196-PA&partnerID=40&md5=c786ab20023c4a3fee860db3acd5305c>, cited By 4. 19, 21
- [50] Liu, G.: *Data collection in mi-assisted wireless powered underground sensor networks: Directions, recent advances, and challenges*. *IEEE Communications Magazine*, 59(4):132–138, 2021. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85106459578&doi=10.1109%2FMCOM.001.2000921&partnerID=40&md5=96b82a473cc40618651a3ee33e5b727e>, cited By 13. 19, 22
- [51] Akafua, J., R. Chapman e H. Guo: *A design of wireless communication and wireless energy transfer system for in-pipe robots*. páginas 84–89, 2021. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85123215891&doi=10.1109%2FWiSEE50203.2021.9613833&partnerID=40&md5=2c32426a8093631f1e26f1400c173f95>, cited By 1. 19, 22, 23
- [52] Ikeda, N., R. Shigeta, J. Shiomi e Y. Kawahara: *Soil-monitoring sensor powered by temperature difference between air and shallow underground soil*. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1), 2020. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85089763285&doi=>

- 10.1145%2f3380995&partnerID=40&md5=6103e3761a1c64fbf8063f0c3b0c6268, cited By 11. 19, 23, 24
- [53] Alshehri, A.A. e A.H. Shewoil: *Connectivity analysis of wireless fractbots network in hydraulic fractures environment*. 2020. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85097634253&partnerID=40&md5=b5e1a83e12839ceb00f77d9f1ac383c7>, cited By 3. 19, 21
- [54] Ikeda, N., R. Shigeta e Y. Kawahara: *Energy prediction for energy management method of sensor node powered by temperature difference between air and shallow underground soil*. Volume 2019-October, 2019. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85078700913&doi=10.1109%2fSENSORS43011.2019.8956881&partnerID=40&md5=552421a91e2eb02953585ebbbe6a4e78>, cited By 1. 19, 23, 24
- [55] Kisseleff, S., I.F. Akyildiz e W.H. Gerstacker: *Survey on advances in magnetic induction-based wireless underground sensor networks*. IEEE Internet of Things Journal, 5(6):4843–4856, 2018. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85053327701&doi=10.1109%2fJIOT.2018.2870289&partnerID=40&md5=25dcbec5f67c61734441e6aaf3497090>, cited By 42. 19, 24, 25
- [56] Zaman, Idrees, Martin Gellhaar, Jens Dede, Hartmut Koehler e Anna Foerster: *Demo: Design and evaluation of molenet for wireless underground sensor networks*. Em *2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops)*, páginas 145–147, 2016. 26
- [57] Zaman, Idrees e Anna Förster: *Self adaptive communication based on soil moisture for wireless underground sensor networks*. Relatório Técnico, Universität Bremen, 2020. <https://media.suub.uni-bremen.de/handle/elib/4292>. 26
- [58] Queiroz, Diego Vêras de: *Simulação realista de redes de sensores sem fio industriais*. Tese de Mestrado, Universidade Federal da Paraíba, João Pessoa, 2016. 29
- [59] Boulis, Athanassios: *Castalia: A simulator for wireless sensor networks and body area networks (user’s manual)*. <https://github.com/boulis/Castalia>, 2013. 30, 31
- [60] Cotrim, Rafael, João Caldeira e Vasco Soares: *Desenvolvimento, simulação e validação de protocolos mac para redes de sensores sem fios*. Revista Brasileira de Computação Aplicada, 14(1):94–105, mar. 2022. <http://seer.upf.br/index.php/rbca/article/view/12679>. 32
- [61] Zhang, Yan, Jie Wang, Wei Chen e Lina Wang: *Comparative study of wireless communication technologies in iot-based applications*. Wireless Communications and Mobile Computing, 2017, 2017. 54
- [62] Liang, Lin, Wei Jiang, Jingna Bi, Xuemin Sherman Shen, Hui Li e Xiaoxiao Chen: *Performance evaluation of network coding in wireless sensor networks: A survey*. IEEE Internet of Things Journal, 6(5):8893–8912, 2019. 54

- [63] Choi, Jeongyeup, Changyoon Kim, Daeyoung Jeon, Jemin Kim, Hanbyul Seo, Joonhyuk Choi e Younggoo Kim: *Performance analysis of lorawan for indoor iot applications*. Sensors, 21(4):1392, 2021. 55
- [64] Tanenbaum, A.S. e D.J. Wetherall: *Redes de computadores*. Pearson Prentice Hall, 2011, ISBN 9788576059240. <https://books.google.com.br/books?id=fnfwkQEACAAJ>. 55
- [65] Kim, Jaeku, Jaeyeong Jung, Sehyun Park, Sungwon Lee, Hyunwoong Choi e Jonghwa Ahn: *Evaluation of lorawan for industrial iot applications: Performance analysis and field test results*. IEEE Access, 8:136241–136249, 2020. 55
- [66] Kumar, Anil e Danda B Rawat: *Estimating network lifetime in wireless sensor networks: A comprehensive review*. Wireless Networks, 25(6):3513–3538, 2019. 55
- [67] Sharma, Pooja, Anil Kumar e Madhusmita Singh: *Energy efficient routing protocols in wireless sensor networks: A survey*. Wireless Personal Communications, 110(3):1373–1416, 2020. 55
- [68] Li, Hua, Yuhao Zhang, Yijie Xu, Yonghui Li e Qian Wu: *Energy-aware network selection for heterogeneous wireless networks: A survey*. IEEE Communications Surveys & Tutorials, 20(4):2737–2774, 2018. 55
- [69] Ventura, Francesca, Osvaldo Facini, Piana Stefano e Rossi Paola: *Soil moisture measurements: Comparison of instrumentation performances*. Journal of Irrigation and Drainage Engineering, 136:81–89, fevereiro 2010. 62

Apêndice A

Fichamento de Artigo Científico

1. **Identificação do aluno:** Luan Borges dos Santos, 20/0104161
2. **Identificação do texto:** Dos Santos, L. B., Rocha Filho, G. P., Bondan, L., Caetano, M. F., Araujo, A., & Marotta, M. A. (2024). SWPTMAC: Sleep Wake-up Power Transfer MAC Protocol. IEEE Network Operations and Management Symposium (NOMS 2024)
3. **Pontos-chave:**

Proposta: Um protocolo que utiliza a técnica de programação sono-vigília em conjunto com a transferência sem fio de energia para maximização da vida útil de uma rede de sensores sem fio subterrâneos.

Contribuição: Desenvolvimento de um novo protocolo de programação de sono-vigília em redes que recebem transferência de energia chamado SWPTMAC (Sleep-Wakeup Power-Transfer Media Access Control). Uma arquitetura de RSSFS com protocolos otimizados para maximização da VUR e aplicada a agricultura.

Validação: A pesquisa apresentada foi realizada através de simulações com o simulador Omnet, utilizando o framework Castália. Comparação com o estado da arte de protocolos típicos de sono-vigília, como o RIMAC e EERIMAC.

Perspectivas: Avaliar em outros cenários além da agricultura. Otimizar a viagem dos drones. Realizar uma implementação física ou um modelo matemático da solução para uma completa validação.
4. **Palavras-chave:** Wireless Underground Sensor Network, Network Lifetime, Sleep-wakeup Protocols, Wireless Power Transfer
5. **Sinopse do texto:** As Redes de Sensores Sem Fio Subterrâneos (RSSFS) são um tipo de Redes de Sensores Sem Fio (RSSF) formadas por sensores enterrados, que

são capazes de realizar o monitoramento do ambiente subterrâneo e comunicam-se sem utilização de fios. As RSSFS podem ter grande utilidade para áreas como, por exemplo, a agricultura, que depende de informações sobre o solo para um bom desempenho e por isso esta foi a área de aplicação utilizada na pesquisa apresentada. Embora sejam de grande relevância, este tipo de rede possui uma grande limitação energética. Devido ao ambiente em que estão inseridas, os recursos energéticos destas redes são difíceis de serem renovados. Sendo assim, com a finalidade de manter a cobertura de rede pelo maior período possível torna-se necessário buscar maneiras de maximizar a Vida Útil da Rede (VUR).

Na literatura revisada, foram encontrados alguns estudos que buscam maximizar a VUR de RSSFS. As técnicas encontradas estão relacionadas com a proposição de melhorias no uso dos recursos energéticos, bem como a utilização de Transferência Sem Fio de Energia (TSFE). Na busca realizada, pode-se concluir que TSFE é feita predominantemente através da indução magnética. Além disso, foi feita uma busca na literatura que foi além das RSSFS, possibilitando encontrar a técnica de programação sono-vigília, que consiste em permitir que algum dispositivo que não está transmitindo ou recebendo informações possa entrar em estado de adormecimento e economizar energia.

Ao analisar trabalhos relacionados com a pesquisa foi possível verificar que nenhum utilizava a programação sono-vigília em RSSFS. Além disso, foi observado que dentre os protocolos de programação sono-vigília analisados nenhum trabalhava em conjunto com a TSFE. Sendo assim, foi proposto o Sleep Wake-Up Power Transfer MAC (SWPTMAC), um protocolo de programação sono-vigília que ao mesmo tempo possibilitasse a utilização da técnica de TSFE. O protocolo foi baseado nos já reconhecidos protocolos de programação sono-vigília RIMAC e EERIMAC.

A avaliação da solução proposta foi realizada utilizando simulação. Foram conduzidas cinco simulações. As três primeiras simulações serviram para avaliar os parâmetros que caracterizam a RSSFS aplicada na agricultura. A quarta simulação avaliou o parâmetro da quantidade de drones que seriam adequados para o cenário de utilização do SWPTMAC na RSSFS. Já a quinta e última simulação serviu para comparação do SWPTMAC com o estado da arte de protocolos típicos de programação sono-vigília, como o RIMAC e EERIMAC.

Ainda existem avanços futuros no estudo do SWPTMAC. São perspectivas de avanços futuros: a avaliação do protocolo proposto em outros cenários além da agricultura, a otimização da viagem dos drones e realização de uma implementação física ou um modelo matemático da solução para uma completa validação. Ainda que

existam avanços a serem feitos, analisando os resultados alcançados pode-se concluir que a pesquisa alcançou êxito tendo em vista que se verificou um ganho de 24 % na estimativa de VUR com o cenário do protocolo proposto.

6. **Análise crítica:** O artigo apresentado aborda um tema relevante e atual no contexto das Redes de Sensores Subterrâneos Sem Fio (RSSFS), que é a maximização da Vida Útil da Rede (VUR) por meio de um protocolo de controle de acesso ao meio eficiente. A escolha da agricultura como área de aplicação da RSSFS avaliada é coerente tendo em vista que para uma produção maior e mais qualificada dos alimentos é de suma importância a coleta de informações sobre o solo.

Buscou-se abordar o assunto no artigo de maneira bem estruturada, trazendo uma base de fundamentação teórica e de trabalhos relacionados que pudesse esclarecer qual o contexto de pesquisa que o trabalho apresentado no artigo está inserido. Além disso, ao apresentar os trabalhos relacionados foi esclarecida a lacuna de pesquisa que a solução proposta no artigo buscou preencher.

Outros artigos conhecidos focam em tratar de Transferência Sem Fio de Energia (TSFE) ou programação sono-vigília de maneira separada. Sendo assim, em comparação com outras abordagens do mesmo assunto, o estudo apresentado no artigo se destaca ao integrar o conceito programação sono-vigília com a técnica de TSFE visando maximizar a VUR. A explicação sobre o protocolo proposto (SWPTMAC) e a máquina de estados que permitem um maior esclarecimento sobre o protocolo podem ser considerados como pontos fortes do artigo.

No geral, o artigo pode ser visto como contribuindo para o campo das RSSFS, oferecendo uma solução de maximização da VUR que poderá ser aplicada em diferentes áreas. A comparação dos resultados alcançados demonstraram um ganho com a solução proposta no artigo. Além disso, a integração de programação sono-vigília e TSFE pode ser vista como algo inovador. O artigo como um todo e as possíveis direções de pesquisa futura sugeridas no artigo podem servir de contribuição para outros pesquisadores.