

WALTER SILVESTRE COAN

Roteamento em redes de sensores sem fio baseado no caminho mais forte em energia

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Curitiba
2012

WALTER SILVESTRE COAN

Roteamento em redes de sensores sem fio baseado no caminho mais forte em energia

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: Ciência da Computação

Orientador: Alcides Calsavara
Co-orientador: Luiz A. P. Lima Jr.

Curitiba
2012

COAN, Walter Silvestre

Roteamento em redes de sensores sem fio baseado no caminho mais forte em energia. Curitiba, 2012.

Dissertação - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Redes de sensores sem fio 2. Algoritmo de roteamento 3. Eficiência energética I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e Tecnologia. Programa de Pós-Graduação em Informática II - t

Dedico este trabalho a minha família, por
todo o apoio.

Agradecimentos

Agradeço por todo o incondicional apoio de meu orientador Prof. Dr. Alcides Cal-savara e de meu co-orientador Prof. Dr. Luiz A. P. Lima Jr.

Agradeço ao apoio da CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior e do Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná (PUCPR) por possibilitar o desenvolvimento desta pesquisa.

Sumário

Agradecimentos	ii
Sumário	iii
Lista de Figuras	vi
Lista de Tabelas	viii
Resumo	ix
Abstract	x
Capítulo 1	
Introdução	1
1.1 Problema	2
1.2 Objetivo	2
1.3 Motivação	3
1.4 Hipótese	3
1.5 Organização	4
Capítulo 2	
Redes de sensores sem fio	5
2.1 Introdução	5
2.2 Redes ad hoc	5
2.3 Redes de sensores sem fio	8

2.3.1	Arquitetura dos dispositivos	10
2.3.2	Tecnologias de comunicação sem fio para redes ad hoc	12
2.4	Roteamento de mensagens em redes de sensores sem fio	14
2.4.1	DSR (<i>Dynamic Source Routing</i>)	16
2.4.2	Roteamento e consumo eficiente de energia em WSN	17
2.5	Trabalhos Relacionados	19
2.5.1	UBPCR	20
2.5.2	<i>Joint Scheduling-power Control Algorithm</i>	21
2.5.3	<i>RAW-E</i>	22
2.5.4	<i>PDSR</i>	22
2.5.5	<i>Roteamento Baseado em Atração de Mensagens</i>	26
2.6	Considerações Finais	28

Capítulo 3

Algoritmo de roteamento SP/WSN	30	
3.1	Introdução	30
3.2	Funcionamento	30
3.3	Mensagens	32
3.3.1	Reluctance Change	32
3.3.2	Send Message to Sink	32
3.3.3	Descoberta das Rotas	34
3.3.4	Propagação da alteração da força de relutância	35
3.3.5	Manutenção das Rotas	36
3.3.6	Diagrama de estados do nó durante a execução do algoritmo	37
3.4	Considerações Finais	38

Capítulo 4

Validação	39
4.1 Introdução	39
4.2 Cenário	39
4.2.1 Aspectos de Implementação	41
4.2.2 Segurança em redes de sensores sem fio	42
4.3 Simulador	43
4.4 Implementação do SP/WSN no simulador	44
4.5 Implementação do PDSR no simulador	48
4.6 Parametrização da Simulação	49
4.7 Resultados	51
4.7.1 Taxa de Entrega de Mensagens	51
4.7.2 Atraso na Entrega de Mensagens	54
4.7.3 Variação da Energia Restante	58
4.7.4 Número de Nós Ativos	60
4.8 Considerações Finais	73
 Capítulo 5	
Conclusões	76
Referências Bibliográficas	78

Lista de Figuras

Figura 2.1	Intervalo de transmissão x zona de captação de sinal do protocolo IEEE 802.11 - Adaptado (JAGANNATHAN, 2007)	8
Figura 2.2	Arquitetura de um nó de uma rede de sensores sem fio - Adaptado (AKYILDIZ et al., 2002)	11
Figura 2.3	Topologias do IEEE 802.15.4 - Adaptado (SCHMIDT-EISENLOHR et al., 2010)	14
Figura 2.4	Difusão direcionada - Adaptado (GEORGE DOLLIMORE JEAN, 2007)	16
Figura 2.5	Descoberta de rota pelo PDSR - Adaptado (WEI et al., 2004)	24
Figura 2.6	Rede magnética virtual - Adaptado (CALSAVARA; LIMA, 2010)	27
Figura 3.1	Força de relutância agindo sobre uma rede magnética virtual	31
Figura 3.2	Construção da rede	35
Figura 3.3	Propagação da alteração da força de relutância	36
Figura 3.4	Diagrama de estados de um nó da rede	38
Figura 4.1	Topologia da rede de sensores	40
Figura 4.2	Área de transmissão e recepção entre nós	41
Figura 4.3	Pilha de protocolos de um sensor - Adaptado (AKYILDIZ et al., 2002)	42
Figura 4.4	Arquivo NED de definição do nó da rede de sensores.	45
Figura 4.5	Arquivo NED de definição da rede	46

Figura 4.6	Diagrama de classes BaseNetwLayer	47
Figura 4.7	Algoritmo SP/WSN em execução, com 100 nós, no simulador OM- Net++ com o MiXim	51
Figura 4.8	Taxa de entrega x tempo - SP/WSN	53
Figura 4.9	Taxa de entrega x tempo - PDSR	55
Figura 4.10	Agrupamento dos nós por estados - SP/WSN	56
Figura 4.11	Atraso na entrega de mensagens - SP/WSN	57
Figura 4.12	Atraso na entrega de mensagens - PDSR	59
Figura 4.13	Variação do percentual de energia restante no cenário I - SP/WSN .	61
Figura 4.14	Variação do percentual de energia restante no cenário I - SP/WSN .	62
Figura 4.15	Variação do percentual de energia restante no cenário I - SP/WSN .	63
Figura 4.16	Variação do percentual de energia restante no cenário II - SP/WSN	64
Figura 4.17	Variação do percentual de energia restante no cenário II - SP/WSN	65
Figura 4.18	Variação do percentual de energia Restante no cenário II - SP/WSN	66
Figura 4.19	Variação do percentual de energia restante no cenário I - PDSR . .	67
Figura 4.20	Variação do percentual de energia restante no cenário I - PDSR . .	68
Figura 4.21	Variação do percentual de energia restante no cenário I - PDSR . .	69
Figura 4.22	Variação do percentual de energia restante no cenário II - PDSR . .	70
Figura 4.23	Variação do percentual de energia restante no cenário II - PDSR . .	71
Figura 4.24	Variação do percentual de energia restante no cenário II - PDSR . .	72
Figura 4.25	Número de nós ativos - SP/WSN	74
Figura 4.26	Número de nós ativos - PDSR	75

Lista de Tabelas

Tabela 4.1	Parametrização da simulação	49
Tabela 4.2	Parametrização do Consumo de Energia do Nó	49
Tabela 4.3	Parametrização área do cenário de simulação com densidade constante	50

Resumo

Esta dissertação apresenta o algoritmo de roteamento SP/WSN (*Strongest Path for Wireless Sensor Networks*) para redes de sensores sem fio com um único nó *sink*. O seu objetivo é manter o balanceamento do nível de energia entre todos os nós, tal que a área e o tempo de cobertura da rede sejam maximizados. Isso é obtido com o emprego do conceito de relutância magnética para, pró-ativamente, estabelecer caminhos com maior quantidade de energia. O SP/WSN foi comparado com o algoritmo PDSR (*Priority Dynamic Source Routing*) através de simulação, considerando-se somente redes estáticas. Os resultados obtidos para taxa de entrega de mensagens e número de nós ativos na rede indicam que o SP/WSN obtem-se uma cobertura de rede maior que a obtida com o PDSR. Além disso, o atraso na entrega de mensagens indica que o SP/WSN tem um desempenho adequado. Finalmente, a análise de variação de energia restante com o SP/WSN permite concluir que o uso de energia pelos nós da rede é balanceado. Com base nos resultados obtidos com a comparação com o PDSR, é possível concluir que o SP/WSN trata-se de uma solução muito mais eficiente e que se adapta melhor as condições existentes nas redes de sensores sem fio.

Palavras-chave: Redes de sensores sem fio, Algoritmo de roteamento, Eficiência energética

Abstract

This Masters dissertation presents the routing algorithm SP/WSN (Strongest Path for Wireless Sensor Networks) for wireless sensor networks with a single node sink. Its goal is to maintain a balanced energy level between all nodes, such that the area and time coverage of the network are maximized. This is achieved by employing the concept of reluctance to establish paths with more energy pro-actively. The SP/WSN was compared with the algorithm PDSR (Priority Dynamic Source Routing) through simulation, considering only static networks. The results obtained for message delivery rate and number of active nodes in the network indicate that the SP/WSN provides a network coverage greater than that obtained with the PDSR. Moreover, the message delay indicates that the SP/WSN has adequate performance. Finally, the analysis of variation of power remaining with the SP/WSN shows that energy use by the network nodes is balanced. Based on the results obtained from the comparison with the PDSR, we conclude that the SP/WSN it is a much more efficient and better suited to conditions in wireless sensor networks.

Keywords: Wireless sensor networks, Routing protocol, Energy efficiency

Capítulo 1

Introdução

Os nós das redes de sensores sem fio (*Wireless Sensor Networks* – WSN) são, tipicamente, pequenos dispositivos eletrônicos de baixo custo, equipados com sensores, microcontroladores, rádio transmissores e fontes de energia não renováveis ou cuja renovação se dá em ciclos, dispostos em uma área de interesse com o objetivo de monitorar um determinado aspecto físico (JUN, 2009). Segundo (CHONG; KUMAR, 2003), a agência DARPA (*Defense Advanced Research Projects Agency*) aponta que WSN é uma tecnologia com grande relevância para o desenvolvimento de pesquisas devido as suas diversas aplicações, incluindo monitoramento militar de regiões em conflito, verificação de condições físicas, tais como a qualidade da água ou do ar, segurança e vigilância de ambientes restritos, criação de casas autônomicas e acompanhamento à distância de sinais vitais de pessoas. Ainda segundo (CHONG; KUMAR, 2003), as primeiras pesquisas realizadas sobre WSN datam da década de 80 no intuito de desenvolver redes com propósitos militares funcionando como uma expansão da ARPANet.

Muitas pesquisas como (AKKAYA; YOUNIS, 2005), contribuíram para a evolução das WSN, desde a área de microcomponentes mais eficientes e baratos, rádio transmissores com maior alcance e mais facilmente gerenciáveis, baterias com maior duração e, por fim, sistemas operacionais, protocolos de comunicação e algoritmos de roteamento que possibilitam um funcionamento mais otimizado dos nós da rede. Observa-se que a pesquisa mais recente em WSN ainda tem foco na evolução desses componentes básicos, indicando que esse tipo de rede ainda não alcançou o estágio de total domínio da tecnologia.

1.1 Problema

O propósito de uma WSN, como afirma (NIKOLAOS; DIMITRIOS, 2008), é garantir que uma área de interesse seja monitorada, tal que os dados coletados sejam transmitidos até um ou mais nós de destino (*sink*), onde possam ser analisados. A eficácia de uma WSN depende, a rigor, das características de sua aplicação. Mas, no caso geral, inclui o tempo de funcionamento total da rede, a cobertura da área monitorada e o roteamento correto de mensagens. O tempo de funcionamento total é dado pelo momento em que nenhum nó consegue entregar mensagens ao *sink*. A cobertura da área monitorada é dada pela parte dessa área onde existem nós em funcionamento e com meios para enviar mensagens ao *sink*. O roteamento correto de mensagens é dado pela entrega com sucesso ao *sink* das mensagens geradas pelos nós em funcionamento. Assim, a eficácia de uma rede depende diretamente da manutenção da carga da fonte de energia em todos os seus nós. Uma vez que o rádio transmissor é o principal responsável pelo uso da fonte de energia de um nó, o algoritmo de roteamento utilizado é crítico para determinar a eficácia da rede.

O problema que este trabalho pretende abordar consiste em aumentar a cobertura realizada pela rede, através de um balanceamento do consumo de toda a energia existente nos nós da rede. Diversas dificuldades estão relacionadas a este cenário:

- Criar rotas entre os nós até o *sink* e que não sobrecarregue uma determinada região.
- Utilizar a menor quantidade possível de mensagens de controle para reduzir o consumo de energia realizado pelo rádio transmissor.
- Evitar que as mensagens, durante o processo de roteamento, entrem em *loop* consumindo mais energia de uma determinada área da rede.

1.2 Objetivo

Este trabalho apresenta um novo algoritmo de roteamento de mensagens para WSN, denominado SP/WSN (*Strongest Path for Wireless Sensor Networks*), cujo princípio é encaminhar as mensagens de um nó ao *sink* através do caminho que apresente o maior nível de energia. Com o propósito de atingir o equilíbrio entre todos os nós da rede no uso de energia, aumentando a eficácia da rede.

Há diversas propostas de algoritmos que otimizam o roteamento visando à eficiência no consumo de energia, tais como: (*i*) o PDSR (WEI et al., 2004), que reativamente realiza

o processo de descoberta de rotas, o que provoca uma inundação de mensagens de controle na rede; (ii) o UBPCR (MIN; KIM, 2007), que modifica a potência do rádio transmissor a fim de encurtar o número de nós necessários para transmitir uma mensagem, abordagem essa que acaba aumentando significativamente o ruído de transmissão entre os demais nós; (iii) o *Joint Scheduling-power Control Algorithm* de (ELBATT; EPHREMIDES, 2002), utiliza uma abordagem parecida com o UBPCR porém baseado na tecnologia TDMA (*Time Division Multiple Access*). Observa-se com isso que não existe uma solução definitiva para o problema de roteamento visando a eficiência energética, e as soluções propostas possuem particularidades que impedem de serem adotadas em qualquer tipo de WSN.

1.3 Motivação

Algoritmos de roteamento com o propósito de reduzir o consumo de energia utilizam diversos artifícios para reduzir o consumo de energia, contudo, boa parte das propostas acaba por penalizar o desempenho da rede em detrimento a conservação da fonte de energia. A motivação para este trabalho consiste na possibilidade de realizar um uso mais eficiente das fontes de energia dos nós, sem restringir o funcionamento, através de um algoritmo de roteamento que não utilize um número excessivo de mensagens de controle.

1.4 Hipótese

A hipótese na qual esse trabalho se baseia é que a aplicação do conceito de relutância magnética, que consiste na força pela qual dois objetos magnéticos se repelem, pode ser utilizado para definição da melhor rota a ser utilizada dentro da rede. A relutância que um nó conhece de seus vizinhos indica a quantidade de energia existente pela rota formada a partir do vizinho até o nó *sink*. Desta forma, as rotas com menor quantidade de energia se tornam mais relutantes em comparação às outras. Esta abordagem nos permite criar um algoritmo de roteamento que, nó a nó, decide qual a melhor rota a ser utilizada considerando todos os nós pertencentes à rota, mas de forma totalmente distribuída, sem a necessidade de nós centralizadores.

O SP/WSN (*Strongest Path for Wireless Sensor Networks*) foi validado através da simulação de redes estáticas, no ambiente OMNet++, configurado com o módulo para WSN denominado MiXiM. A sua avaliação foi feita por meio de vários indicadores, in-

cluindo o tempo de vida da rede, a taxa de entrega de mensagens, o número de nós ativos na rede e o atraso na entrega de mensagens. Esses mesmos indicadores foram também gerados para o algoritmo PDSR, para fins de comparação.

1.5 Organização

O restante deste trabalho está assim organizado: no Capítulo 2 apresenta-se um apanhado dos principais conceitos relacionados a Redes de Sensores sem Fio; no Capítulo 3 apresenta-se o algoritmo SP/WSN; no Capítulo 4 são apresentados detalhes sobre o ambiente de simulação a implementação do algoritmo SP/WSN e sua validação através de diversos indicadores. E, por fim, no Capítulo 5 é apresentada a conclusão e são apresentadas as propostas de trabalhos futuros.

Capítulo 2

Redes de sensores sem fio

2.1 Introdução

Redes de sensores sem fio é um tópico bastante estudado devido suas diversas aplicações. Neste capítulo, discutimos os principais conceitos relacionados a WSN, sua concepção possibilitada pelas redes ad hoc, como é a arquitetura dos componentes desta rede e, por fim, os métodos de roteamento utilizados.

2.2 Redes ad hoc

Uma rede ad hoc, também conhecida pelo termo MANET (*Mobile ad hoc network*) é definida por (JAGANNATHAN, 2007) como uma coleção autônoma de usuários móveis que se comunicam sobre um canal de comunicação que possui: (i) restrições de largura de banda; (ii) recursos de fonte de energia limitados; (iii) utilizados em um ambiente altamente dinâmico. O conceito de usuários autônomos é considerado o principal diferencial desta rede para as demais tecnologias, por utilizar uma topologia onde não existe um componente centralizador, como um switch ou roteador. Cada ponto da rede deve realizar as tarefas básicas de iniciar seu funcionamento, executar o processo de descoberta de outros indivíduos com os quais possa se comunicar e, por fim, estabelecer a conexão para a transferência de dados. O canal de comunicação utilizado por esse tipo de aplicação é o segundo ponto importante do conceito. Como não existe um posicionamento fixo dos componentes da rede, devido à característica da mobilidade e dinamicidade, a conexão precisa ser realizada através de comunicação sem fio. A adoção da comunicação através de

rádio transmissão traz a vantagem da mobilidade, porém possui diversas restrições como:

- Limitação do raio de transmissão;
- Limitações severas de largura de banda;
- Maior vulnerabilidade a interferências de outros dispositivos que utilizam rádio transmissão;
- Consumo mais elevado de energia para seu funcionamento.

Em alguns casos, a limitação de recursos de fonte de energia se dá devido à arquitetura adotada para os dispositivos deste tipo de rede. De acordo com a aplicação desejada, os componentes precisam ser miniaturizados ao extremo, reduzindo as possibilidades de acoplamento de grandes fontes de energia. Durante o funcionamento da rede, os recursos de fonte de energia são consumidos até o momento em que cessam e o dispositivo é desligado por completo. Portanto, a rede precisa ser capaz de reconstruir rotas utilizando os dispositivos que ainda estão em funcionamento para que as informações coletadas continuem sendo transmitidas aos pontos de processamento.

Um conjunto de dispositivos dentro de uma MANET formam uma rede capaz de operar em dois tipos de transmissão segundo (JAGANNATHAN, 2007): *(i)* dois nós podem se comunicar entre si diretamente se estiverem próximos o suficiente (ponto-a-ponto); *(ii)* ou (*multihop*) indiretamente através de nós intermediários que retransmitem os pacotes. O processo de transmissão indireta de dados entre *host* obriga a existência de um algoritmo de roteamento capaz de coordenar as trocas de pacotes considerando a inexistência de uma arquitetura centralizada. Essa questão remete a duas características importantes das MANETs: a limitação temporal e de extensão espacial. Conforme explica (JAGANNATHAN, 2007), estas limitações garantem que o ato de criação e dissolução da rede ad hoc deve ser suficientemente simples e conveniente de modo a ser realizado por usuários não-técnicos. Nenhuma habilidade técnica especializada é necessária e pouco ou nenhum investimento de tempo e de recursos adicionais deve ser necessário, além do simples posicionamento dos nós; o que possibilita às aplicações que utilizam este tipo de rede uma grande flexibilidade.

Ao contrário de redes baseadas em meios físicos de transmissão como o IEEE 802.3, redes sem fio possuem limitações de banda e sofrem com interferências. Essas interferências podem ocorrer devido ao número de usuários da rede ou ainda devido a fatores externos que influenciam o processo de rádio transmissão. Segundo (JAGANNATHAN,

2007), os meios mais eficientes para combater a interferência é aumentar a capacidade e reutilização espacial da rede. Ocorre que normalmente os protocolos utilizados em redes sem fio suportam apenas um único nó realizando a transmissão, enquanto os demais interessados realizam o processo de recepção do sinal e sua decodificação, dentro de uma mesma área de abrangência de transmissão e recepção. Quando um nó possui uma potência de transmissão muito alta, ele tem alcance a mais nós da rede em comparação a outros nós. Desta forma, ao entrar em modo de transmissão, vários nós recebem seu sinal devido sua maior área de abrangência. Porém, nem sempre todos esses nós têm interesse nessa informação e, devido o grande alcance, o canal de transmissão desta região acaba ficando congestionado, além de gerar um custo muito alto para o nó transmissor. Portanto, a correta adequação da potência de transmissão é importante para reduzir a possibilidade de congestionamentos na rede, além de aumentar o tempo de vida das fontes de energia utilizadas pelos nós.

O protocolo IEEE 802.11 da camada de enlace foi especialmente desenvolvido para aplicação em ad hoc networks. A especificação do protocolo descreve: serviços, métodos de identificação dos nós da rede, técnicas de coexistência de redes similares além de questões de privacidade e segurança que tornaram possível a criação das redes ad hoc. Portanto, a sua especificação abrange questões da camada de enlace do modelo OSI, definindo padrões de banda e frequências que são utilizadas. Segundo descrito por (JAGANNATHAN, 2007), na especificação do protocolo IEEE 802.11 MAC, cada transmissor quando utilizado em potência máxima atinge uma área de transmissão de raio de 250 metros, quando não existem obstáculos que interfiram no sinal, sendo essa área conhecida por alcance de transmissão. Conforme a Figura 2.1, considerando a mesma potência máxima de transmissão, a área de raio de 500m a partir do transmissor é chamada de intervalo de captação de sinal, o que significa que em toda esta área é possível que um receptor receba algum sinal transmitido. A diferença do intervalo de transmissão para o intervalo de captação de sinal nos dá a zona de captação de sinal, nessa área o receptor consegue receber algum sinal transmitido, porém não é capaz de decodificar corretamente o que foi transmitido. Esse fenômeno ocorre devido a degradação do sinal, além de possíveis interferências que o sinal sofre durante a distância percorrida dentro da zona de captação de sinal.

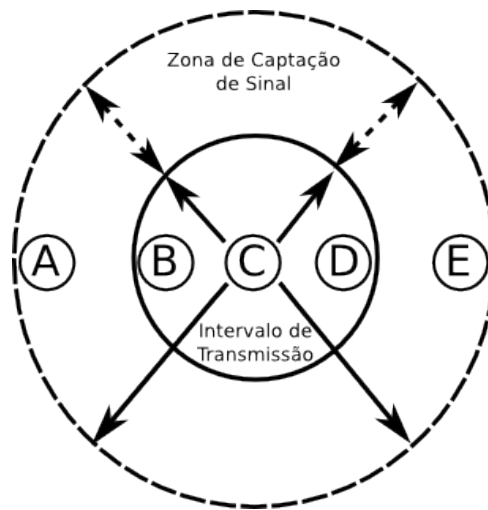


Figura 2.1: Intervalo de transmissão x zona de captação de sinal do protocolo IEEE 802.11 - Adaptado (JAGANNATHAN, 2007)

2.3 Redes de sensores sem fio

As redes de sensores sem fio são uma das principais aplicações das redes ad hoc, sendo que os dois termos estão tão fortemente ligados que muitas vezes são confundidos como um mesmo conceito. Porém a presença do sensor é que as diferencia, além disso, nas WSN temos inicialmente uma região bem definida chamada de espaço inteligente, que consiste no local onde os sensores serão instalados. Segundo (GEORGE DOLLIMORE JEAN, 2007), um espaço inteligente é qualquer local físico com serviços incorporados, isto é, serviços fornecidos apenas ou principalmente dentro desse espaço físico. É possível introduzir dispositivos de computação ao acaso, onde não existe nenhuma infra-estrutura, para executar uma aplicação, como o monitoramento ambiental.

Os autores (JAGANNATHAN, 2007) e (GEORGE DOLLIMORE JEAN, 2007) chamam esta região de espaços inteligentes, devido o fato de que os sensores ali distribuídos possibilitam que uma arquitetura baseada em componentes eletrônicos registrem modificações no ambiente e, através da análise desse conjunto de informações, tomem decisões, disparem alertas ou ainda realizem operações pré-definidas. Basicamente, os sensores consistem em componentes que têm a capacidade de monitorar fatores físicos, como a temperatura, a umidade, a velocidade de deslocamento ou o posicionamento em relação aos pólos magnéticos da terra, para transformar essas informações em sinais digitais capazes de serem interpretados e transmitidos através de um canal de comunicação.

Outra questão importante é a mobilidade. As redes ad hoc têm a característica de mobilidade, porém as WSN não necessariamente têm seus nós móveis. Em alguns ca-

sos, os sensores são aleatoriamente distribuídos dentro da área de interesse, então passam a estabelecer conexões e a transmitir seus dados sem modificar sua posição. Segundo (JAGANNATHAN, 2007), uma rede de sensores sem fio pode não ter um endereço IP, enquanto um nó da rede ad hoc pode tê-lo. Essa diferença ocorre principalmente pela necessidade de simplificar o funcionamento da rede e a inexistência de um componente centralizador capaz de determinar endereços de IP distintos para cada dispositivo. A mobilidade dos nós agrega a característica da dinamicidade do ambiente que está fortemente presente nas WSN. Constantemente, os nós da rede podem encontrar novos nós para realizar conexões (associações). No caso de redes sem mobilidade, a dinamicidade ocorre devido ao desligamento de um nó vizinho, o que ocasiona a perda da associação. Como afirma (GEORGE DOLLIMORE JEAN, 2007), usamos o termo associação para o relacionamento lógico formado por pelo menos um par de componentes que se comunicam, durante um período de tempo bem definido, sendo que ocorrem interações entre os componentes durante o tempo de associação.

Baseado no conceito das MANETs, as WSN utilizam protocolos de roteamento multihop, onde os dados são transmitidos de um nó a outro até chegarem a seu destino final. De acordo com (GEORGE DOLLIMORE JEAN, 2007), um motivo pelo qual os nós não se comunicam em um único salto (hop) com todos os outros nós, mas, em vez disso, se comunicam diretamente apenas com os nós vizinhos, é que a comunicação sem fio tem um alto consumo de energia que aumenta com o quadrado do alcance do rádio. A comunicação direta entre os nós traz uma maior complexidade na disputa entre os nós para ter acesso de transmissão na rede. Desta forma (JAGANNATHAN, 2007) afirma que um nó sensor gasta a maior parte de sua energia em comunicação, que normalmente envolve a transmissão e a recepção de dados. A Fórmula 2.1 de (SHIH et al., 2001) define o consumo de energia como:

$$P_{radio} = N_{tx}[P_{tx}(T_{on-tx} + T_{st}) + P_{out}T_{on-tx}] + N_{rx}[P_{rx}(T_{on-rx} + T_{st})] \quad (2.1)$$

Segundo (SHIH et al., 2001):

- $N_{tx/rx}$ são a média do número de vezes por segundo que a transmissão ou recepção de dados ocorre;
- $P_{tx/rx}$ são a energia consumida para transmissão ou recepção de dados;
- P_{out} é a potência de transmissão de dados;

- $T_{on-tx/rx}$ são o tempo no momento da transmissão ou recepção de dados;
- T_{st} é o tempo inicial da transmissão de dados.

É possível observar com a Fórmula 2.1 que o consumo de energia está diretamente ligado ao número de vezes em que ocorre a troca de dados e principalmente o tempo que cada uma dessas trocas demora. Portanto, é preferível realizar uma maior quantidade de processamento pelo nó a fim de reduzir a quantidade de dados transmitidos. Segundo (JAGANNATHAN, 2007), o consumo de energia pelo processamento é menor que o consumo da transmissão dos dados, principalmente devido à evolução dos microcontroladores utilizados. Segundo (POTTIE; KAISER, 2000), um processador de propósito geral que execute três milhões de instruções consome 3J de quantidade de energia. A mesma quantidade de energia consegue transmitir 1Kbit a 100m de distância. Portanto, a realização de processamento no próprio nó ocasiona uma redução de energia. (POTTIE; KAISER, 2000) ainda cita alguns exemplos de tarefas que podem ser realizadas pelos nós para reduzir a necessidade de transmissão de dados: (i) agregar ou tirar a média de valores recebidos de nós vizinhos, com o objetivo de examinar valores de uma área, em vez de um único sensor; (ii) filtrar dados sem interesse ou repetidos; (iii) examinar dados para detectar alarmes; (iv) ativar ou desativar sensores, de acordo com os valores percebidos. É importante observar que somente considerar o processamento não é a solução definitiva para o consumo de energia dos nós, simplesmente aumentar o poder de processamento desses componentes acarretaria diretamente em um aumento do consumo de energia e no aumento do tamanho físico do nó, o que prejudica a miniaturização do sensor além de aumentar seu custo.

Por fim, os nós de uma WSN possuem componentes sensores que têm por objetivo servir de meio de conexão do dispositivo com o mundo real. Segundo (GEORGE DOLLIMORE JEAN, 2007), os sensores são dispositivos que medem parâmetros físicos e fornecem seus valores para o software. Juntamente com os sensores podem existir os controladores, que são componentes que podem alterar alguma característica no mundo físico.

2.3.1 Arquitetura dos dispositivos

Um nó de uma rede WSN consiste basicamente em um dispositivo eletrônico composto pelos seguintes componentes (processador, memória, transmissor, sensor, conversor analógico para digital, fonte de energia, gerador de energia, módulo de localização e mó-

dulo de movimentação), exemplificados na Figura 2.2. A arquitetura desses dispositivos é simples e restrita para possibilitar que os sensores tenham o tamanho mais reduzido possível, e que seu custo final de produção seja reduzido ao máximo, pois muitos desses dispositivos quando deixam de funcionar não são reaproveitados.

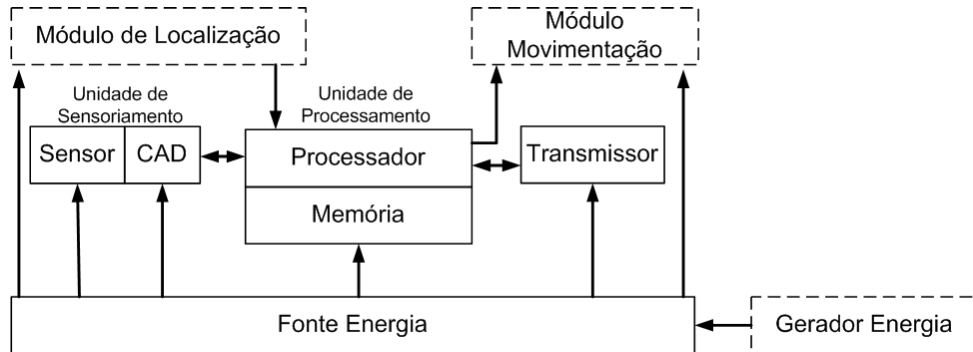


Figura 2.2: Arquitetura de um nó de uma rede de sensores sem fio - Adaptado (AKYILDIZ et al., 2002)

Como citado por (AKYILDIZ et al., 2002), os principais componentes são: (i) unidade de processamento, (ii) unidade sensor, (iii) unidade *transceiver* e (iv) unidade de energia. Além dessas unidades, outras auxiliares podem ser acopladas, como a unidade de localização que normalmente é composta por um dispositivo de geolocalização como o GPS ¹, módulo de movimentação que tem por objetivo controlar o deslocamento do sensor (denominado como controlador na Sessão 2.3), quando necessário. Por fim, o módulo gerador de energia é uma unidade suplementar e tem como objetivo de recarregar a unidade de energia.

Ainda segundo (AKYILDIZ et al., 2002), a unidade sensor é composta por duas sub-unidades: o sensor e o conversor analógico digital (ADC) ². O sinal analógico produzido pelo sensor, baseado na observação de fenômenos físicos, é convertido para o sinal digital pelo ADC, para então alimentar a unidade de processamento com essa informação. Na unidade de processamento temos o processador que normalmente é ligado em uma pequena quantidade de memória para armazenar os programas e os dados capturados pelo sensor. Alguns tipos de redes WSN utilizam de sistemas operacionais para gerenciar seu funcionamento como o TinyOS. Segundo (LEVIS et al., 2003), o *TinyOS* é um sistema operacional especialmente desenhado para redes de sensores. Ele possui um modelo de programação baseado em componentes, provido pela linguagem nesC, que é um dialeto do C. O *TinyOS* não é um sistema operacional tradicional, na verdade se assemelha a um framework de programação para sistemas embarcados com um conjunto de componentes

¹GPS - *Global Positioning System*

²ADC - *Analog to digital conversion.*

que possibilitam utilizar a abstração de um sistema operacional dentro de um nó de uma WSN.

O transmissor consiste na unidade de rádio transmissão que tem por objetivo enviar e receber informações através de um protocolo de comunicação para redes sem fio. Segundo (AKYILDIZ et al., 2002), a unidade de transceiver pode funcionar de forma passiva ou ativa utilizando componente de transmissão de dados por meio óptico ou por rádio frequência (RF)³. A transmissão por RF requer modulação, regulador de banda, filtragem, modulação e multiplexação, o que deixa o circuito mais complexo e caro.

A unidade de energia é o principal componente que fornece energia para os demais componentes da arquitetura, sendo responsável pelo tempo de funcionamento do sensor. Essa unidade é composta por baterias, que provêm uma fonte limitada de energia.

2.3.2 Tecnologias de comunicação sem fio para redes ad hoc

As principais tecnologias utilizadas, em redes sem fio e redes ad hoc, conforme (RUBINSTEIN et al., 2006) são o *Bluetooth*, o IEEE 802.11, TDMA (JAGANNATHAN, 2007) e ainda destaca-se o *ZigBee*.

O *Bluetooth* é uma tecnologia sem fio (IEEE 802.15.1) utilizada para criar redes pessoais de baixo alcance. Segundo (RUBINSTEIN et al., 2006), o *Bluetooth* pode trabalhar de duas maneiras, no modo *piconet* ou no modo *scatternets*. No modo *piconet* um dos dispositivos assume o papel de mestre e os demais de escravo. O mestre se conecta aos dispositivos escravos, e então a comunicação pode ocorrer nos dois fluxos enquanto houver transmissão de sinal entre eles. No segundo modo de funcionamento chamado de *scatternets*, ocorre quando duas *piconets* se conectam através de um dispositivo mediador, possibilitando assim a extensão da área de atuação da rede. O alcance de transmissão do dispositivo *Bluetooth* depende de sua classe, iniciando em um metro podendo chegar a uma distância de 100 metros aproximadamente. Na sua última versão de especificação, número 3.0, pode alcançar uma taxa de transmissão de 24Mbit/s.

O IEEE 802.11, segundo (RUBINSTEIN et al., 2006), é uma das tecnologias de comunicação sem fio mais adotada mundialmente. Sua especificação da camada física se subdivide em diversas outras como: IEEE 802.11a, IEEE 802.11b, IEEE 802.11g e a especificação em atual adoção no mercado, o IEEE 802.11n. Em cada versão houve um incremento na taxa e alcance de transmissão dos dados. Novos padrões estão em de-

³RF - *Radio Frequency*.

envolvimento como por exemplo o IEEE 802.11ac que permite a virtualização de vários *access points* em um mesmo dispositivo físico, e o padrão IEEE 802.11af que permite a transmissão de sinal de TV em alta definição.

Como afirma (RUBINSTEIN et al., 2006), o protocolo MAC IEEE 802.11 especifica dois algoritmos de acesso ao meio físico: o DCF (*Distributed Coordination Function*) e o PCF (*Point Coordination Function*). O DCF é um mecanismo distribuído no qual cada nó verifica se o canal de transmissão está ocioso para realizar a transmissão. Já o PCF é um mecanismo centralizado, onde o access point controla o acesso ao canal de transmissão, sendo o PCF designado para redes que possuem uma infraestrutura pré-existente. Nos dois modos o conceito de CRC (*Cyclic Redundancy Check*) é utilizado para garantir a integridade dos dados trafegados pela rede. O IEEE 802.11-1997 utiliza a banda de 2.4GHz e suporta transmissão em uma taxa de 1 a 2Mbps. Já o IEEE 802.11b suporta uma taxa de até 11Mbps, utilizando o DSSS (*Direct Sequence Spread Spectrum*), tecnologia que possibilita a modulação do espectro de propagação do sinal, aumentando a taxa de transferência. O IEEE 802.11a usa a banda de 5GHz e trabalha com a taxa de transferência de até 54Mbps e utiliza o OFDM (*Orthogonal Frequency Division Multiplexing*) que possibilita o envio multiplexado de vários sinais em frequências diferentes, também aumentando a taxa de transferência. O IEEE 802.11g utiliza a mesma frequência do IEEE 802.11b, 2.4GHz, com a taxa de transferência do IEEE 802.11a de 54Mbps e também utiliza o OFDM.

Como explica (JAGANNATHAN, 2007), que o TDMA (*Time Division Multiple Access*) é um padrão de comunicação baseado na divisão do canal de transmissão em tempos determinados. Assim cada nó, quando deseja se comunicar com outro, se inscreve e tem acesso a um determinado slot, tempo de comunicação. A principal vantagem desse método é a redução da quantidade de interferência que os demais nós da rede geram sobre um nó que deseja transmitir dados. Uma desvantagem é a necessidade de recalculer os slots de transmissão toda vez que um novo nó entra na rede, segundo (BEIN, 2009). É uma tecnologia amplamente utilizada para a telefonia celular, mas que também é aplicada a redes ad hoc devido o longo alcance de transmissão.

O ZigBee (IEEE 802.15.4), de acordo com (PAN; TSENG, 2008), é um protocolo de comunicação sem fio que tem por objetivo simplificar a forma de comunicação entre as camadas lógicas e físicas do modelo OSI. O ZigBee suporta três tipos de redes chamadas: *star*, *tree* e *mesh*. No modelo *star* existe o papel de um nó coordenador, que tem o trabalho de inicializar, manter e coordenar as mensagens na rede. Os modelos *tree* e *mesh* permitem a comunicação direta entre nós da rede, mesmo com a existência do coorde-

nador. A rede do tipo *tree* possui um adicional mecanismo com o objetivo de redução do consumo de energia chamado *beacons*, que gerencia o roteamento para a economia de energia. Segundo (SCHMIDT-EISENLOHR et al., 2010), as redes ZigBee podem adotar dois tipos de topologias: (i) estrela onde o nó centralizador também é conhecido como PAN *Coordinator* (*Personal Area Network Coordinator*), (ii) ponto-a-ponto, onde também existe a figura do PAN *Coordinator*, porém os nós podem trocar informações diretamente, como ilustra a Figura 2.3.

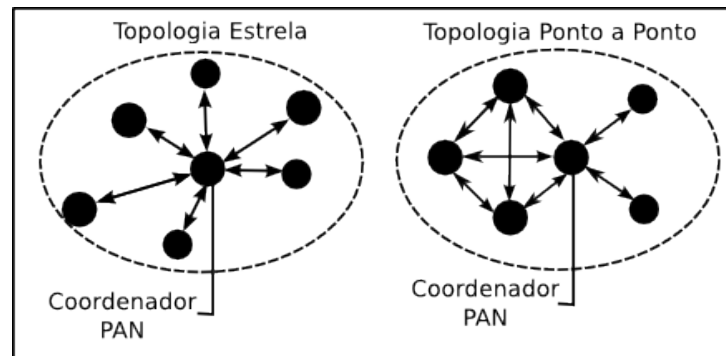


Figura 2.3: Topologias do IEEE 802.15.4 - Adaptado (SCHMIDT-EISENLOHR et al., 2010)

2.4 Roteamento de mensagens em redes de sensores sem fio

As técnicas de roteamento, descritas por (AKKAYA; YOUNIS, 2005), e aplicadas em redes de sensores sem fio se confundem com as aplicadas nas redes dinâmicas, por serem uma subcategoria das WSN. Algumas características são desejáveis nos protocolos de roteamento segundo (MISRA; WOUNGANG; MISRA, 2009):

- O funcionamento deve ser distribuído para aumentar a capacidade de recuperação em caso de falhas;
- Os algoritmos devem ter eficiência na economia de energia. Desta forma, a carga a ser realizada deve ser distribuída entre os nós participantes da rede;
- Os protocolos devem considerar questões de segurança, principalmente devido a vulnerabilidade que normalmente esse tipo de aplicação está sujeita;
- Aplicação de protocolos híbridos que agregam qualidades de outros protocolos. Onde

se prefere as implementações reativas às proativas, devido as questões que os algoritmos proativos realizam uma maior quantidade de ações que as reativas;

- Por fim, os protocolos devem estar atentos a QoS (*Quality of Service*), que consiste em garantir tempos aceitáveis para a transmissão de mensagens;

Protocolos proativos, como explica (MISRA; WOUNGANG; MISRA, 2009), são os que continuamente realizam o processo de aprendizagem sobre a topologia da rede. Quando a topologia muda constantemente existe um grande custo em manter essa informação sempre atualizada, principalmente quando não há informações para serem trocadas. Já os algoritmos reativos realizam o processo de definição da rota a ser utilizada somente quando existe a necessidade de transmitir alguma informação. Porém as duas abordagens quando exclusivamente aplicadas não são consideradas soluções conclusivas para os problemas. As pesquisas recentes utilizam o conceito de protocolos híbridos, onde atuam de forma proativa e reativa de acordo com a necessidade. Um exemplo de algoritmo híbrido é o VRP *Virtual Routing Protocol* de (ALBINI et al., 2006), que utiliza do conceito de redes virtuais sobre uma rede física para aumentar a confiança da entrega de pacotes de dados a um destino na rede. O VRP utiliza tanto a abordagem proativa como reativa para realizar a descoberta de rotas: proativamente buscando por rotas virtuais através de um processo de envio de mensagens unicast que são retransmitidas até nós intermediários chamados de *Virtual Hop*; e reativamente através de um segundo processo de descoberta de rotas quando um nó explorador recebe uma mensagem de erro indicando que uma rota não é mais válida, através da inundação da rede com mensagens de baixa prioridade para a busca de rotas.

Outro conceito aplicado na elaboração dos protocolos é a programação orientada a dados. Neste caso, todo o funcionamento do algoritmo está focado em garantir que as informações devem chegar ao seu destino. Essa abordagem é utilizada devido à forte característica das WSN e principalmente pelo meio de transmissão utilizado. De acordo com (GEORGE DOLLIMORE JEAN, 2007), são duas abordagens comuns nos protocolos para WSN: a difusão direcionada e o processamento de consulta distribuído. A difusão direcionada, conforme (HEIDEMANN et al., 2001) (apud (GEORGE DOLLIMORE JEAN, 2007)), o programador especifica interesses, que são declarações de tarefas injetadas no sistema em certos nós denominados de nós coletores (*sinks nodes*). Esses nós enviam para os demais nós da rede seu interesse por um determinado evento, então os demais nós propagam essa informação até os chamados nós fonte (*sink*). Esse tipo de nó é responsável por monitorar os eventos que os coletores têm interesse e retornar os dados aos coletores

conforme a ocorrência. A difusão direcionada pode no momento da distribuição dos dados, conforme a Figura 2.4, gerar comunicação redundante, conforme o item C, onde mais de um nó fonte pode transmitir a mesma informação para o coletor. Esse problema pode ser solucionado realizando um processamento de filtro nos nós intermediários da fonte ao coletor, evitando a transmissão de dados redundantes.

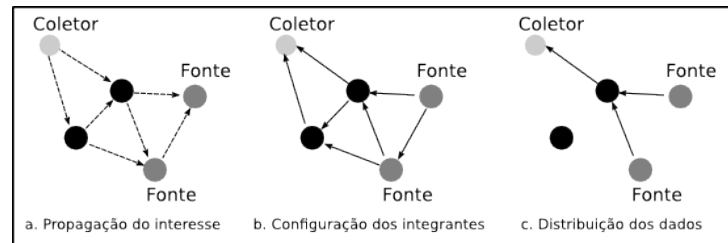


Figura 2.4: Difusão direcionada - Adaptado (GEORGE DOLLIMORE JEAN, 2007)

Outra técnica, de acordo com (GEHRKE; MADDEN, 2004) (apud (GEORGE DOLLIMORE JEAN, 2007)), é a execução de consultas distribuídas, onde o coletor dispara uma consulta a todos os nós da rede. Esses nós ao receberem essa consulta executam e realizam processos de agrupamento e filtro. Os resultados são transferidos de volta para o coletor utilizando o caminho dinamicamente descoberto durante o início do processo de consulta.

2.4.1 DSR (*Dynamic Source Routing*)

Outro protocolo que é comumente utilizado como base para comparação ou o estudo é o DSR (*Dynamic Source Routing*) (JOHNSON; MALTZ, 1996). Este algoritmo foi proposto como uma das primeiras soluções capazes de lidar com a dinamicidade existente dentro das redes ad hoc. O conceito básico desse algoritmo define que o roteamento é definido na origem, portanto, o nó remetente escolhe a sequência completa de nós pelos quais o pacote deve caminhar. O remetente explicitamente lista a rota no cabeçalho do pacote, identificando cada nó pelo seu endereço. O processo de envio de pacote se inicia com o nó de origem, que fica responsável por buscar em seu *cache* de rotas uma que tenha como destino o nó desejado. Uma vez definida a rota, ela é gravada no cabeçalho da mensagem e transmitida para o primeiro nó da rota. O nó ao receber a mensagem, primeiramente verifica, na rota gravada na mensagem, se ele é o destino da mensagem. Caso seja, o processo é finalizado pois a mensagem chegou em seu destino. Caso contrário, o nó passa a mensagem para o próximo nó da rota. Cada nó possui sua tabela de rotas, atualizada por um processo chamado *Route Discovery Protocol*. No momento em que um

nó deseja enviar um pacote mas a tabela ainda não possui uma rota para o destino, o processo é executado. O processo que garante a validação da existência de uma rota se chama *Route Maintenance*. Sempre que um nó deseja enviar uma nova mensagem, mas não possui uma rota para transmiti-la, a mensagem é armazenada temporariamente até que o *Route Discovery Protocol* descubra uma nova rota até o destino. O processo de *Route Discovery* inicia com o envio em forma de *broadcast* de um pacote para descoberta das rotas. Esse pacote basicamente possui informações a respeito da origem e destino da mensagem, e é enviado inicialmente pelo nó que deseja atualizar a tabela de rotas para todos os seus vizinhos diretos. Caso esse pacote alcance um nó que seja o nó destino, é retornado à origem um pacote contendo a rota descoberta. Cada pacote de descoberta de rotas é numerado com um identificador que é utilizado para evitar duplicações nas respostas das rotas, ou até mesmo loop durante o processo de *broadcast* para os vizinhos. Esse processo é auxiliado por uma lista mantida nos próprios nós que armazena as últimas requisições através do par de informações (Origem e Identificador da requisição). Desta forma, o algoritmo, ao receber a requisição de descoberta de rota, efetua as seguintes operações:

- Se a requisição estiver na lista das últimas requisições, descarta a mensagem;
- Se o próprio endereço do nó aparecer dentro da rota que está sendo descoberta, também descarta a mensagem;
- Se o endereço destino da rota é igual ao endereço do nó atual, retorna uma mensagem de *Route Reply* à origem informando a rota descoberta;

Caso contrário coloca seu endereço na listagem de rotas e reenvia a mensagem de requisição em forma de *broadcast* para seus vizinhos diretos. O processo de *Route Maintenance* consiste em uma ação de verificação se as rotas armazenadas ainda são válidas. Ele ocorre através de um processo onde cada nó, durante um período determinado de tempo, escuta o canal de comunicação e caso algum evento de erro ou falha ocorra, é possível detectar a indisponibilidade do nó e assim anular as rotas às quais pertence.

2.4.2 Roteamento e consumo eficiente de energia em WSN

As técnicas de roteamento descritas nas seções anteriores contribuem para o volume de mensagens que será trafegado na rede. Portanto, quanto maior o número de mensagens trafegadas, maior o consumo da fonte de energia pelo seu principal consumidor, o

transmissor. Fica claro que uma possível solução para alcançar um consumo eficiente de energia é utilizar algoritmos de roteamento que além de não desperdiçar energia, realizem um consumo equilibrado de todas as fontes para que não ocorra o desligamento de regiões importantes da rede.

Esse problema é conhecido nas WSN como *energy sink-hole problem*⁴ descrito por (SAJAL; HABIB, 2008). Em redes WSN estáticas, onde os nós não se movimentam, os nós próximos ao *sink* são extremamente utilizados por serem o principal caminho até ele. Como resultado, eles acabam esgotando suas baterias mais rapidamente que os demais nós. Uma das soluções descritas por (SAJAL; HABIB, 2008), utiliza os nós que estão localizados nos limites exteriores da rede de sensores. Normalmente esses nós fazem pouco trabalho de roteamento de mensagens para o *sink*, portanto, uma estratégia é utilizar a energia restante desses nós para dividir o trabalho de roteamento com eles e assim balancear o consumo geral da rede.

Em (NIKOLAOS; DIMITRIOS, 2008) são definidas duas técnicas diferentes para conservação de energia em WSN: técnica passiva e a técnica ativa. A técnica passiva é o desligamento do rádio transmissor para que não ocorra atividade de comunicação de dados. Ao realizar o processo de desligamento do rádio transmissor, o nó está poupando uma grande quantidade de energia, porém esse nó fica isolado da rede sem receber informações de seus vizinhos. Esses nós, da mesma forma, ficam impossibilitados de retransmitir mensagens o que pode reduzir em muito número de rotas existentes na rede. Controlar o desligamento dos nós e o processo de roteamento de mensagens é outro tipo de abordagem utilizada no desenvolvimento de algoritmos de roteamento, contudo, fica claro que existe uma perda na área de cobertura da rede o que prejudica o desempenho de funcionamento como um todo. Outros métodos passivos de redução de consumo de energia citados são: (i) a utilização de processadores que permitam o processo de VSP (*Variable Speed Processor*), onde o *clock* de funcionamento do processador pode ser regulado através da redução da voltagem; (ii) desligamento temporário de componentes do nó sensor como o conversor analógico digital, processador e memória; (iii) explorar capacidade de processamento paralelo do processador do nó; (iv) controle da potência de transmissão e controle do tratamento de mensagens de erro de transmissão. Já as técnicas ativas de conservação de energia, segundo (NIKOLAOS; DIMITRIOS, 2008), melhoram o funcionamento do nó ao contrário de realizar seu desligamento parcial, alguns exemplos de técnicas são: (i) utilização de métodos de *handshake* como RTS/CTS (*request-to-send/clear-to-send*) para evitar excesso de colisões; (ii) controle da potência de transmissão para reduzir o número

⁴Problema do buraco de energia próximo ao nó *sink*

de colisões; *(iii)* minimizar o consumo de energia por pacote, que consiste na energia necessária para transmitir um pacote desde sua origem até seu destino; *(iv)* maximizar o tempo antes que ocorra alguma partição da rede, que consiste no tempo até que a rede perde sua conectividade; *(v)* minimizar a variação no nível de energia entre os nós, evitando desta maneira que alguns nós sejam mais penalizados que outros; *(vi)* minimizar o custo por nó, que o custo que cada nó individual tem para transmitir um pacote.

A utilização de técnicas ativas no desenvolvimento de um algoritmo de roteamento que busque a eficiência energética traz uma solução muito mais flexível que a utilização das técnicas passivas, uma vez que elas estão fortemente ligadas aos aspectos físicos da implementação dos nós, como controle de desligamento de partes do nó. Porém, não é possível simplesmente implementar todas as técnicas ativas de uma só vez, visto que, algumas delas são contraditórias como por exemplo: minimizar o consumo de energia por pacote e maximizar o tempo antes que ocorra uma partição da rede. Para minimizar o consumo por pacote, o algoritmo de roteamento deve escolher rotas que utilizem o menor caminho possível até o destino, desta maneira um número menor de nós será envolvido e o consumo de energia é reduzido. Porém, se esse caminho for muito utilizado, devido ao fato de ser o menor, ele terá a energia de seus nós rapidamente esgotada, o que aumenta as chances de uma partição da rede. De todas as possibilidades, a que acredita-se ser uma solução que traga melhores resultados é minimizar a variação do nível de energia entre os nós. Essa abordagem consiste em utilizar todos os nós da rede de forma similar, para que nenhum caminho seja super utilizado e desta maneira não reduza o *lifetime*, e da mesma maneira aumente o tempo antes da partição da rede. O consumo de energia por pacote, nesta abordagem, é penalizado pois muitas vezes será necessário utilizar caminhos mais longos que consumam mais energia, porém evita-se que existam nós com altos percentuais de energia restante sem ser utilizada.

2.5 Trabalhos Relacionados

Manter o funcionamento de uma rede de sensores sem fio consiste primordialmente em balancear de forma adequada o uso dos recursos de fonte de energia disponíveis, para que as tarefas de processamento e de transmissão dos dados possam ser executadas. Com esse objetivo, diversos estudos dentro das redes ad hoc estão concentrados em desenvolver novos algoritmos de roteamento de pacotes de dados mais eficientes e que melhorem o aproveitamento dos recursos de fonte de energia. A aplicação de algoritmos de roteamento

dentro do contexto das WSN se mostra como uma alternativa viável para gerenciar o principal problema existente nesse tipo de aplicação: a fontes de energia limitadas. Conforme já discutido na contextualização das WSN, o principal consumidor de recursos da fonte de energia de um sensor é o processo de transmissão baseado em comunicação sem fio.

Dentre as soluções encontradas, para roteamento em redes WSN, observa-se o conceito de protocolos *cross-layer* como (ELBATT; EPHREMIDES, 2004), (MACEDO; GRILO; NUNES, 2009) que consiste na comunicação direta entre camadas do esquema OSI ⁵ não adjacentes. O objetivo dessa comunicação direta é que as camadas inferiores do modelo tenham seu funcionamento alterado pelas camadas de controle superiores. Esse tipo de abordagem cria uma dependência formal entre as camadas que se comunicam, gerando uma especificidade na aplicação dessas soluções.

Outra abordagem encontrada de algoritmos são os que não realizam o processo de *cross-layer*. Esses algoritmos possuem uma maior flexibilidade de aplicação, uma vez que repetam as interfaces de comunicação entre as camadas do modelo OSI.

2.5.1 UBPCR

O UBPCR (*Utility-based power control routing*) proposto por (MIN; KIM, 2007) consiste de um algoritmo de roteamento para WSN com o objetivo de redução do consumo de energia durante seu funcionamento. Este algoritmo considera: (i) SINR (*Signal-to-interference-plus-noise ratio*) que indica a quantidade do sinal de transmissão que foi corrompido por interferências no canal de comunicação, (ii) potência de transmissão, (iii) quantidade de bateria restante. Com base nessas informações ele define a rota que será adotada pelos pacotes. O algoritmo funciona por demanda, portanto, é considerado como reativo.

O processo de descoberta das rotas é dividido em duas fases principais: o agendamento do canal e controle de potência de transmissão. A fase de agendamento identifica a melhor frequência para realizar a transmissão dos dados, e o controle de potência identifica qual a potência de transmissão que deve ser utilizada. O processo de simulação, descrito por (MIN; KIM, 2007), considerou que os nós realizam o processo de deslocamento físico, e a comunicação entre um nó origem e um nó destino sempre é feita através de um conjunto de nós intermediários. Durante o processo de transmissão, o canal de comunicação fica reservado apenas para o nó transmissor, e as fontes de energia dos nós e a potência

⁵OSI - (*Open Systems Interconnection*)

dos transmissores foram ajustadas para o valor máximo suportado pelos dispositivos. A comparação de performance realizada foi feita utilizando apenas o algoritmo desenvolvido alterando parâmetros de sua forma de decisão das rotas a serem utilizadas conforme as opções a seguir: (i) UBPCR aplicando um *threshold* no nível de energia restante para selecionar a rota; (ii) UBPCR sem um *threshold* do nível de energia restante para selecionar a rota; (iii) idem (i) sem realizar a estimativa da potência de transmissão; (iv) idem a opção (ii) sem realizar a estimativa da potência de transmissão; (v) idem a opção (iii) sem utilizar o processo de selecionar a rota com maior quantidade de recursos; por fim (vi) idem a opção (iv) sem utilizar o processo de selecionar a rota com maior quantidade de recursos.

Os resultados apresentados por (MIN; KIM, 2007) demonstraram que a primeira abordagem testada, onde o algoritmo realiza todos os controles aos quais foi projetado, obteve os melhores resultados no aumento do tempo de vida da rede. Observa-se que os métodos utilizados para a comprovação do funcionamento do algoritmo não o comparam com as outras soluções já propostas, o que leva a questionar se o algoritmo realmente tem um funcionamento adequado em comparação a outras propostas.

2.5.2 *Joint Scheduling-power Control Algorithm*

O objetivo do estudo proposto em (ELBATT; EPHREMIDES, 2004) consiste em um algoritmo de roteamento de mensagens com foco no envio de mensagens entre nós diretamente ligados. O principal objetivo é reduzir a interferência gerada por vários nós no canal de comunicação, reduzir o consumo de energia e prolongar o funcionamento dos dispositivos. O algoritmo foi desenvolvido para redes TDMA (*Time Division Multiple Access*) e TDMA/*code-division multiple-access wireless ad hoc networks*. Utilizando o conceito *cross-layer*, a camada de enlace envia comandos diretos à camada física para o gerenciamento da energia e da potência de transmissão. O algoritmo inicialmente define quantos nós podem transmitir de forma segura dados simultaneamente, adaptando a potência dos transmissores para reduzir o máximo possível o índice de SINR. Como a tecnologia TDMA trabalha com slots de tempo para a transmissão, o serviço de coordenação organiza a ordem de transmissão dos nós, já o serviço que adapta à potência do sinal de transmissão auxilia na redução da interferência na rede.

Os resultados apresentados por (ELBATT; EPHREMIDES, 2004), demonstraram que houve uma redução significativa de consumo de energia quando comparado através de simulações utilizando o algoritmo proposto. Porém, observa-se novamente que a solução

proposta não compara o algoritmo com outros trabalhos o que não possibilita afirmar sua real eficiência na redução do consumo de energia.

2.5.3 RAW-E

O RAW-E (*Energy Aware Random Asynchronous Wakeup*) para (PARUCHURI; DURRESI; BAROLLI, 2005) consiste em um algoritmo que utiliza a informação de quantidade de energia de seus vizinhos para decidir se o nó deve entrar em estado de *sleep*⁶. O funcionamento do algoritmo envia uma mensagem para os nós vizinhos quando ocorre uma mudança significativa de sua quantidade de energia, os vizinhos ao receberem essa mensagem, guardam a quantidade de energia contida em seu vizinho, para então comparar com seu próprio nível de energia. Caso o nó possua uma maior quantidade de energia, ele se manterá no estado de ativo por mais tempo que seus vizinhos, caso contrário, se seu nível de energia for menor que seus vizinhos, entrará no estado de *sleep*. Foi definido um tempo mínimo em que o nó deve permanecer ativo, para possibilitar que ele receba as mensagens de controle do protocolo. O processo de descoberta de novos vizinhos ocorre cada vez que um nó sai do estado de *sleep* e retorna ao estado ativo, quando envia por *broadcast* uma mensagem a seus vizinhos diretos informando seu nível de energia. Assim, se um nó ainda não tinha conhecimento da sua existência, ele atualiza sua tabela de informações. Para realizar o roteamento das mensagens o algoritmo decide o menor caminho entre o nó de origem e o nó de destino, considerando apenas os nós vizinhos ativos. Segundo (PARUCHURI; DURRESI; BAROLLI, 2005) o RAW-E consegue atingir seu objetivo de aumentar o tempo de vida da rede, mesmo em redes com grande densidade de nós, fato esse que auxilia no sucesso do algoritmo devido a existência de diversas rotas redundantes.

2.5.4 PDSR

Em (WEI et al., 2004) foi proposto o algoritmo PDSR (*DSR with priority routing algorithm*) que consiste em um melhoramento feito no algoritmo DSR já detalhado na Sessão 2.3. O PDSR inclui ao DSR o conceito de priorização, que é traduzido na forma de um índice calculado com base na quantidade de energia que o nó possui. Conforme (WEI et al., 2004) "baixa quantidade de energia representa uma prioridade baixa", desta forma cada nó é capaz de calcular sua prioridade que varia de 0 até mp (prioridade máxima)

⁶Desligamento parcial do nó por um determinado período de tempo.

baseado na sua energia. Da mesma forma que o nó, as rotas também recebem um índice de prioridade. Sempre que uma mensagem trafega pelos nós adotando uma rota, o algoritmo verifica nó a nó se sua prioridade é igual ou maior que a rota. Caso o nó possua uma prioridade menor que a da rota, ele não repassa a mensagem e gera uma mensagem de retorno informando à origem que uma nova rota deve ser descoberta. Portanto, sempre que um nó deseja enviar uma mensagem o processo se inicia com uma busca em sua tabela de rotas, conceito já existente no DSR, para o destino desejado, caso já exista a rota, esta é utilizada, caso contrário, o processo de descoberta de novas rotas é iniciado.

O processo de descoberta funciona de forma similar ao DSR, porém o indicador de prioridade é inserido no processo com o objetivo de filtrar as rotas com maior energia possível. Quando o nó inicia o processo de descoberta, ele envia uma mensagem chamada RREQ (*Route Request*) através de *broadcast*, contendo o destino desejado e a prioridade da rota, a seus nós vizinhos diretos. Esta prioridade para a busca de rotas é inicialmente igual à prioridade do próprio nó menos um. Quando os nós vizinhos recebem a mensagem RREQ podem realizar as seguintes operações:

- Se o nó já recebeu o mesmo pacote de RREQ ele simplesmente ignora a requisição, para evitar que o processo entre em loop;
- Se a prioridade do próprio nó for menor que a prioridade da rota desejada, a requisição também é descartada;
- Caso o nó possua em sua tabela de rotas, uma que alcance o destino e a prioridade desta rota não tenha o valor menor de que a prioridade da rota desejada, a rota encontrada é retornada ao nó origem da busca;
- Caso nenhuma das opções anteriores possa ser atendida, o nó realiza o broadcast das requisição RREQ para seus vizinhos;

Conforme (WEI et al., 2004), ilustrado na Figura 2.5, o nó 0 pretende enviar uma mensagem até o nó 5, mas não conhece ainda uma rota até ele, o índice de prioridade escolhido para a busca da nova rota tem o valor quatro, então o nó origem 0 inicia o processo de descoberta de rotas enviando a mensagem RREQ via broadcast. A primeira possível rota descoberta pelo processo seria o caminho 0,2,5. Porém, o nó 2 ao receber a requisição avalia que seu índice de prioridade é menor que o da rota, então ele aborta o processo de descoberta descartando a mensagem. Outra rota que pode ser descoberta é 0,1,3,5, neste caso, todos os índices de prioridade satisfazem a regra de seus valores por

serem iguais ou maiores que o da rota desejada. Portanto, quando a RREQ chega ao nó 5 ele retorna a rota para chegar até ele.

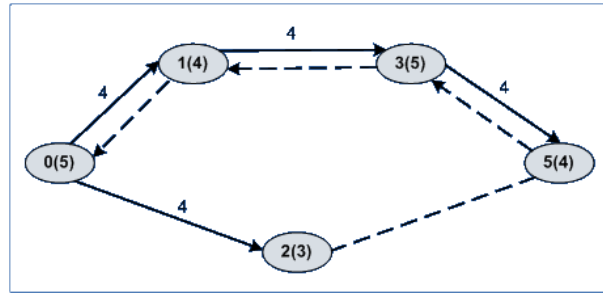


Figura 2.5: Descoberta de rota pelo PDSR - Adaptado (WEI et al., 2004)

Durante o processo de descobertas de novas rotas a mensagem RREQ além de carregar o identificador do nó destino também carrega consigo o índice de prioridade da rota desejada. Esse índice é calculado de acordo com as seguintes regras: (i) Se a tabela de rotas do nó origem estiver vazia, o índice de prioridade da rota desejada será o índice do nó -1; (ii) Se a tabela de rotas do nó origem não estiver vazia, porém não exista uma rota que chegue ao destino, o índice de prioridade da rota desejada será a média dos índices de prioridade das rotas existentes.

Com o funcionamento da rede, a quantidade de energia dos nós tende a reduzir, alterando o valor do índice de prioridade do nó. Quando um nó recebe uma mensagem de dados onde ele faz parte da rota, o nó compara seu índice de prioridade com o índice de prioridade gravado na mensagem. Caso seu valor seja menor que o da rota ele interrompe o envio e retorna uma mensagem de erro de rota à origem. Quando essa mensagem chega ao nó origem, a rota é invalidada na tabela e caso não exista outra opção de rota, o processo de descoberta de rotas é novamente iniciado. Desta maneira, o algoritmo tenta impedir que uma rota seja muito utilizada e rapidamente os nós que a compõem tenham sua bateria esgotada.

A escolha de qual rota utilizar quando existe mais de uma opção na tabela do nó de origem também requer atenção. A decisão é sempre utilizar as rotas de maior índice de prioridade. (WEI et al., 2004) descreve dois processos complementares que são executados juntamente com o algoritmo PDSR para melhorar seu desempenho, o primeiro está relacionado a falha na descoberta de novas rotas. Caso, após um tempo determinado, nenhuma mensagem de descoberta de rota retorne com um caminho completo até o destino desejado, uma nova mensagem de descoberta é gerada mas com a prioridade da requisição anteriormente enviada decrescida em um. Segundo (WEI et al., 2004) esse processo pode gerar a situação de flood na rede, consumindo mais recursos dos nós. Esse problema

é resolvido através de uma melhor distribuição das faixas de valores para os índices de prioridade e a criação de um temporizador que protege as rotas descobertas, evitando que, durante um tempo determinado, novas rotas sejam excluídas. Essa distribuição define intervalos não simétricos do percentual de energia e os associa com uma prioridade. Quanto maior a prioridade, maior o intervalo de percentual de energia vinculado a ele, a medida que a prioridade reduz, esse intervalo do percentual também é reduzido. Com isso, durante o processo inicial de funcionamento da rede, o nó demora um tempo maior para trocar de prioridade, porém, à medida que sua bateria se esgota e chega próximo ao percentual zero, sua prioridade cai mais rapidamente.

O segundo processo que melhora o desempenho do PDSR está ligado ao temporizador para a proteção das rotas descobertas. É definido um valor de tempo constante para todas as rotas descobertas, sempre que a diferença entre o tempo atual e o tempo em que a rota foi descoberta possua um valor menor que o tempo de proteção da rota, essa rota será utilizada mesmo que sua prioridade tenha um valor maior que a prioridade de qualquer nó que a compõe. Essa proteção é obtida porque no momento em que o nó decide enviar uma mensagem de dados e define sua rota, caso esteja dentro do período de proteção, ao copiar a prioridade da rota para a mensagem o algoritmo recalcula a prioridade subtraindo a prioridade da rota de mp . Desta maneira a prioridade será negativa, e nenhum nó da rota possuirá uma prioridade menor, com isso nenhum nó irá gerar a mensagem de erro de rota o que impede desta rota ser descartada.

Segundo (WEI et al., 2004), para o teste do algoritmo foi utilizado o simulador NS-2 utilizando uma área de $1000 \times 1000 \text{ m}^2$, quantidades de nós distintas, uma área de alcance de transmissão de 250m, velocidades de movimentação aleatória entre 5 a 20m/s com um tempo de pausa de 4s. Constatou-se que o *lifetime*⁷ da rede foi aumentado em média 306,598% utilizando o PDSR em relação ao DSR, também foi constatado que a sobrecarga de mensagens para o controle do consumo de energia é pequeno em relação ao DSR original melhorando em muito o tempo de vida da rede.

A principal desvantagem do protocolo PDSR consiste no processo de descoberta de novas rotas herdado do protocolo DSR. O custo do processo de descoberta de rotas é muito grande o que acaba por consumir uma quantidade significativa de energia dos nós da rede.

⁷Tempo em que o primeiro nó da rede deixa de funcionar devido o esgotamento da bateria.

2.5.5 Roteamento Baseado em Atração de Mensagens

Dois estudos realizados por (LIMA; CALSAVARA, 2010) e (CALSAVARA; LIMA, 2010) definiram uma nova proposta de algoritmos para roteamento de mensagens dentro de uma rede. O algoritmo *Short Path*, criado por (CALSAVARA; LIMA, 2010), trabalha de forma proativa, definindo para cada nó da rede para qual próximo nó a mensagem deve ser enviada. Essa definição é feita através do conceito de atração magnética. Fazendo uma analogia com a física, os magnetos são corpos metálicos com a propriedade de atrair outros corpos metálicos. Já o algoritmo *Quick Path*, criado por (LIMA; CALSAVARA, 2010), utiliza o mesmo princípio de atração magnética, porém sem utilizar o menor caminho como é feito no *Short Path*.

O conceito de atração magnética foi aplicado no trabalho realizado por (CALSAVARA; LIMA, 2010), onde cada nó da rede possui um índice de atração. Esse índice é compartilhado pelo nó com seus vizinhos diretos, desta forma, ao receber uma mensagem, cada nó sabe quem é seu vizinho direto com maior força de atração e para ele destina a mensagem. A atualização da força de atração ocorre de forma contínua através dos vizinhos diretos e indiretos e é feito pelo algoritmo. O indicador de atração pode ser atribuído a diversos fatores dentro do contexto de uma aplicação real. Por exemplo, caso o propósito da aplicação é realizar o balanceamento de carga entre uma rede de nós, os pontos da rede com maior ociosidade possuem um valor de atração maior, puxando para si as mensagens de tarefas que trafegam pela rede em busca de um destino para serem processadas. À medida que esses nós acumulam tarefas, sua ociosidade é reduzida e outros nós da rede podem ter uma menor carga de trabalho, passando então esses nós a ter uma força de atração maior. Para um melhor entendimento do trabalho estudado, uma breve descrição do funcionamento do algoritmo será feita a seguir.

A Figura 2.6 ilustra a topologia de uma rede composta por 5 nós, onde cada nó possui um valor que indica sua força de atração, este valor é descrito por (CALSAVARA; LIMA, 2010) como $F(x)$. Observa-se que as setas indicam as relações de atração que um nó influencia seus vizinhos, a direção da flecha indica quem é o nó influenciado, por exemplo, o nó 3 exerce uma influência sobre o nó 1. Existe também a possibilidade da influência ser bidirecional, como no caso dos nós 1 e 2, a definição desta influência entre os nós é definida de forma estática por (CALSAVARA; LIMA, 2010). Como o algoritmo *Short Path* funciona sobre o conceito de uma rede *overlay*, considera-se que as relações de atração somente podem existir se na camada de rede abaixo desta definida pelo algoritmo existir um canal de comunicação estabelecido entre os nós. Cada nó possui seu pivô local,

representado por $M(x)$, e pivô global, representado por $P^*(x)$. O pivô local trata-se do nó vizinho que tem o menor caminho até o pivô global, e o pivô global é o nó da rede que possui a maior força de atração.

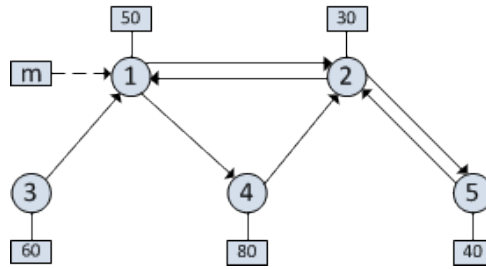


Figura 2.6: Rede magnética virtual - Adaptado (CALSAVARA; LIMA, 2010)

Além dessas informações, a estrutura de dados do algoritmos ainda prevê mais as seguintes informações:

- $S(x)$ - consiste no conjunto de nós que magnetizam x . No caso do nó 1, temos os nós 1, 2, 3, observando que um nó magnetiza-se a si próprio;
- $P^*(x)$ - consiste no Pivô Global do nó x ;
- Distância $S(x)$ até $P^*(x)$ - é a distância em saltos do nó que magnetiza x até o pivô global;
- *Timestamp* última atualização - marcador de hora que armazena quando a última informação foi recebida sobre os nós $S(x)$;
- $T(x)$ - conjunto de nós que x exerce influência magnética, no caso do nó 1 temos 1, 2, 4;
- $K(x)$ - conjunto de nós conhecidos por x , que exercem força de magnetização direta ou indireta sobre x ;

Ainda para cada $K(x)$ são armazenados a força de atração, o timestamp da última atualização e a distância de x até cada $K(x)$.

Essa estrutura provê as informações necessárias para a criação e envio das mensagens que o protocolo necessita para seu funcionamento. Sendo as mensagens:

- Mudança da força de atração - é enviada quando o nó verifica que sua força de atração modificou em um percentual significativo previamente parametrizado. Essa mensagem é enviada a todos os nós que fazem parte do conjunto $T(x)$.

- Mudança do pivô - é enviada quando o nó verifica que seu pivô global foi modificado, da mesma forma a mensagem é destinada a todos os elementos do conjunto $T(x)$.

Na Figura 2.6 uma mensagem m deseja ser enviada pelo nó 1, que será enviada a seu pivô local $M(x)=2$, pois este nó é o vizinho mais próximo do pivô global. O nó 2 ao receber a mensagem repete a operação enviando a mensagem a seu pivô local $M(x)=4$, que trata-se do pivô global.

As simulações realizadas em (CALSAVARA; LIMA, 2010) foram feitas através de um simulador desenvolvido especialmente para o estudo proposto. Foi utilizada uma topologia de rede estática, onde os nós foram representados por um grafo já conhecido por todos os nós da rede. Além disso, a definição da relação de magnetização entre os nós também foi definida de forma estática pelo simulador. Os estudos demonstraram que o algoritmo teve um bom desempenho no cenário proposto, principalmente devido ao número reduzido de mensagens trocadas pelo protocolo. Segundo o estudo publicado, esta solução foi aplicada apenas a redes estáticas com o objetivo de realizar um balanceamento de carga de processamento entre nós de uma rede física. Apesar do modelo de teste utilizado pelo estudo publicado, acredita-se que o conceito de atração de mensagens por um nó da rede pode ser adaptado ao contexto das WSN, principalmente pelo dinamismo e aleatoriedade com que as conexões ocorrem, devido a criação e destruição de conexões frequentes.

2.6 Considerações Finais

Neste capítulo foram apresentados diversos conceitos, técnicas e pesquisas realizadas na área de redes de sensores sem fio. Observou-se a relevância do tema desta dissertação. Os trabalhos relacionados mostram que diversas soluções foram propostas utilizando as mais diferentes técnicas para o consumo eficiente das fontes de energia de uma WSN. Porém conclui-se que novas propostas de algoritmos podem ser desenvolvidas com o mesmo propósito, obtendo melhores resultados dos já apresentados, visto que as propostas pesquisadas utilizam em sua maioria técnicas reativas de conservação de energia, que reduzem a eficiência em relação a cobertura da área de interesse, e as propostas que utilizam técnicas ativas de conservação de energia, necessitam de muitas mensagens de controle o que acaba reduzindo o tempo total de funcionamento da rede. O próximo capítulo apresenta a principal contribuição deste trabalho, o desenvolvimento de um novo algoritmo de roteamento para redes de sensores, focado no consumo eficiente da fonte de energia dos nós. Trata-se do algoritmo proposto para este trabalho, SP/WSN. Este

algoritmo utiliza os conceitos, também aplicados por (CALSAVARA; LIMA, 2010), de atração magnética como técnica ativa para a conservação de energia, além de evitar a utilização excessiva de mensagens de controle.

Capítulo 3

Algoritmo de roteamento SP/WSN

3.1 Introdução

O algoritmo *Short Path*, de (CALSAVARA; LIMA, 2010) e descrito na Seção 2.5.5, foi adaptado para possibilitar a aplicação no ambiente das WSN, dando origem ao SP/WSN (*Strongest Path for Wireless Sensor Networks*), com o objetivo de maximizar a cobertura da rede, isto é estender o tempo de funcionamento dos nós e aumentar a área monitorada.

3.2 Funcionamento

Cada nó da rede tem inicialmente sua força de atração definida antes que as trocas de mensagens se inicie. Considera-se que o nó *sink* possui o valor máximo de força de atração da rede, pois ele tem por objetivo atrair todas as mensagens para si. A atração das mensagens pelos nós está diretamente ligada à existência de uma conexão que possibilite um caminho entre os nós até o *sink*. Uma vez estabelecida uma conexão de comunicação entre dois nós de uma rede, a troca de mensagens pode ocorrer nos dois sentidos. É importante destacar que o nó *sink* não gera mensagens de dados, apenas mensagens de controle, quando necessário. A principal modificação realizada no algoritmo proposto por (CALSAVARA; LIMA, 2010) é a introdução do conceito de *força de relutância* (RF - *Reluctance Force*), que consiste na força pela qual dois objetos magnéticos se repelem. Desta forma, a força de relutância entre dois nós indica qual força atua entre eles; quanto maior esta força, maior o custo da troca de mensagens.

A força de relutância de um nó $RF(x)$, calculada de acordo com a Fórmula 3.1,

considera os seguintes componentes:

- a energia residual (*Residual Energy Percent*) em x , $REP(x)$: a força de relutância é inversamente proporcional à energia residual.
- a força de relutância do seu pivô local, $M(x)$: a força de relutância do pivô local é adicional.

$$RF(x) = \frac{1}{REP(x)} + RF(M(x)) \quad (3.1)$$

Desta forma, cada nó, ao enviar uma mensagem para o vizinho direto, busca em sua tabela de rotas qual vizinho possui a menor força de relutância. Com isso, ele seleciona o caminho até o *sink* com a maior quantidade de energia. A Figura 3.1 apresenta um exemplo: no instante de tempo T0, os nós {A, B, C} possuem o mesmo percentual residual de energia, isto é, 100%, logo, seus valores de força de relutância são iguais a 1. Entre os nós B, C até o *sink* considera-se que a força de relutância é zero, visto que o *sink* não possui limitação de fonte de energia. No instante de tempo T1 o nó B tem seu percentual de energia restante em 50% e sua força de relutância é recalculada para 2. Então, as mensagens geradas pelo nó A serão roteadas pelo nó C, até que sua força de relutância se torne maior que a força de B. Para o algoritmo proposto a distância em saltos entre os nós continua a mesma, sendo que a força de relutância atua como um fator para a escolha da rota com maior quantidade de energia.

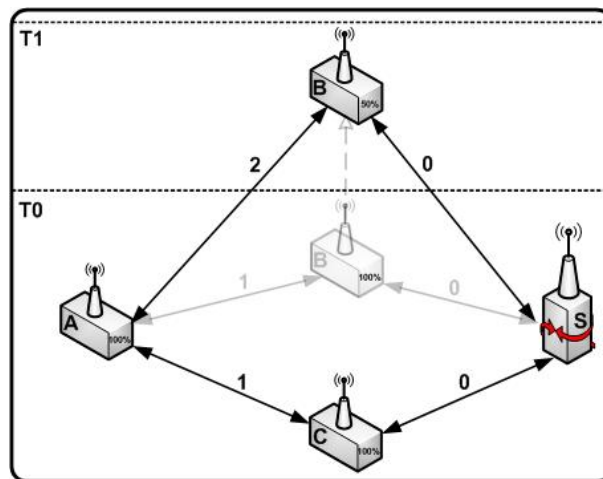


Figura 3.1: Força de relutância agindo sobre uma rede magnética virtual

3.3 Mensagens

No algoritmo SP/WSN, o funcionamento de uma WSN, basea-se em dois tipos de mensagens:

- *Reluctance Change*: mensagem enviada quando a força de relutância é modificada.
- *Send Message to sink*: consiste na mensagem da camada de aplicação, que contém os dados gerados pelo nó sensor, para ser roteada até o nó *sink*.

3.3.1 Reluctance Change

A mensagem *Reluctance Change* é enviada pelos nós da rede a seus vizinhos sempre que: (i) o nó recalcula seu pivô local; (ii) sua energia residual é reduzida em um percentual maior que o *threshold* estabelecido. Isso indica que a energia residual contida nos nós que compõem o caminho até o *sink*, no qual este nó está contido foi modificada e, portanto, a força de relutância deverá ser recalculada. A mensagem é composta pelas seguintes campos:

- Identificador do nó origem x .
- Identificador do nó destino y .
- A força de relutância de x em relação ao *sink*.
- *Timestamp* em x no momento em que a mensagem foi gerada.

O Algoritmo 1 descreve o tratamento que um nó y dá a uma mensagem do tipo *Reluctance Change* enviada de um nó x .

Como o cálculo da diferença de alteração do *threshold*, conforme o Algoritmo 1, é feito em módulo, caso ocorra uma redução ou aumento do percentual de energia restante da bateria, o algoritmo é capaz de, nas duas situações, identificar a relevância da modificação do percentual para propagar aos vizinhos quando necessário.

3.3.2 Send Message to Sink

Esta mensagem tem por objetivo transportar os dados gerados pelos nós da rede até o *sink*, sendo que ela será roteada pelo algoritmo SP/WSN através do caminho mais forte

Algorithm 1 Tratamento que um nó y dá a uma mensagem do tipo *Reluctance Change* enviada de um nó x

Quando o nó y recebe de x uma mensagem do tipo *Reluctance Change*

```

if  $timestamp > lastTimestamp$  then
   $neighbors[x].RF \leftarrow msg.RF$ 
   $neighbors[x].timestamp \leftarrow msg.timestamp$ 
   $oldRF \leftarrow RF(y)$ 
  Recalcula  $RF(y)$  e  $M(y)$ 
  if  $|RF(y) - oldRF| > threshold$  then
    Propagar  $RF(y)$  pela mensagem Reluctance Change
  end if
  if  $oldRF = \infty$  and  $RF(y) \neq \infty$  then
    Notificar
  end if
else
  Descarta a mensagem
end if

```

até o *sink*, considerando a sua força de relutância. Basicamente a mensagem é composta pelos seguintes campos:

- Identificador do nó origem x .
- Dados capturados pelo sensor.
- *Timestamp* em x no momento em que a mensagem foi gerada.

O Algoritmo 2 descreve o tratamento que um nó y dá a uma mensagem do tipo *Send Message to sink* enviada de um nó x . É importante destacar que caso o algoritmo tente enviar a mensagem para seu pivô local $M(x)$ e o envio não seja confirmado com a mensagem de ACK, o algoritmo busca um novo vizinho com a maior força de relutância. Aguardar a mensagem de ACK se mostrou importante para possibilitar o processo de descoberta do desligamento de um nó vizinho, sem a necessidade de criar mensagens extras para verificar se os nós ainda continuam em funcionamento. Caso o nó não possua nenhum vizinho direto, devido a uma momentânea desconexão, o algoritmo guarda a mensagem em *Buffer*, até que receba uma mensagem *reluctance change*, e com isso conheça um novo caminho até o *sink*. Essa desconexão pode ocorrer quando, por exemplo, um nó próximo ao *sink* esgota sua energia e os nós que dependiam dele para chegar ao *sink* precisam olhar para sua tabela de rotas para encontrar outro pivô local que os leve até o destino.

Algorithm 2 Tratamento que um nó y dá a uma mensagem do tipo *Send Message to sink* enviada de um nó x

```

Quando o nó  $y$  recebe de  $x$  uma mensagem do tipo Send Message to sink
sucesso = false
while não sucesso do {mensagem Send Message to sink não for entregue}
  while  $RF(y) = \infty$  do {Aguarda notificação, até que o nó tenha uma rota até o
  sink}
    Aguardar Notificação
  end while
  Enviar mensagem Send Message to sink para  $M(y)$ 
  Aguardar confirmação de  $M(y)$  (ACK), por um tempo  $T$ 
  if Não recebeu de  $M(y)$  a confirmação ACK then
    Retirar  $M(y)$  da lista de vizinhos
    Recalcular  $M(y)$  e  $RF(y)$ 
  else
    sucesso = true
  end if
end while

```

3.3.3 Descoberta das Rotas

Inicialmente, os nós são dispostos aleatoriamente dentro do espaço de interesse. Conforme a Figura 3.2, o nó *sink* envia uma mensagem de *Reluctance Change* em *broadcast* para todos seus vizinhos diretos. O processo de envio de mensagens em *broadcast* permite que todos os vizinhos recebam a mensagem e verifiquem se devem ou não processar o pacote, considerando a energia consumida pelo nó transmissor, esse processo se torna mais econômico, visto que diversos nós recebem a mensagem com apenas um processo de envio. Esta primeira mensagem contém o valor de força de relutância igual a zero. Os vizinhos ao receberem a mensagem, incluem o nó em sua tabela de rotas e como o nó C não possuía um pivô local, ele elege o nó *sink* como seu pivô. Como houve a mudança do pivô local, o nó C recalcula sua força de relutância, conforme a Fórmula 3.1, e a propaga através de uma nova mensagem do tipo *Reluctance Change* em *broadcast* para seus vizinhos. Neste ponto tanto o nó B quanto o *sink* recebem esta mensagem, porém só o nó B irá processá-la, visto que o nó *sink* descarta mensagens que não sejam do tipo *Send Message to Sink*. O nó B, ao receber a mensagem *Reluctance Change* de C, também a adiciona em sua tabela de rotas e recalcula sua força de relutância e também a propaga. Por fim, a mensagem propagada pelo nó B é recebida pelos nós A e C, o nó C simplesmente guarda as informações em sua tabela e não realiza a propagação, pois B sempre terá um valor de RF superior ao existente entre o nó C e o *sink*, já o nó A, repete o processo de atualização de sua tabela de rotas, e como o nó B é seu pivô local, o nó A recalcula sua força de

relutância e a propaga para seus vizinhos diretos. O nó B ao receber a atualização de A, simplesmente atualiza sua tabela, pois o nó C possui uma força de relutância menor.

Após cada um dos nós receber uma mensagem, atualizar sua tabelas de rotas e eleger um pivô local, o nó está pronto para enviar mensagens do tipo *Send Message to Sink*.

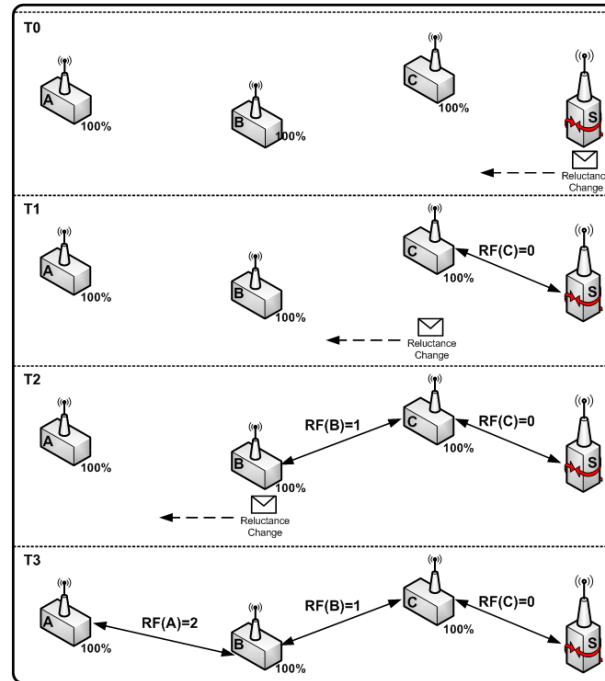


Figura 3.2: Construção da rede

3.3.4 Propagação da alteração da força de relutância

A Figura 3.3 ilustra outro processo que ocorre durante o funcionamento da WSN. O nó C, por ser o único nó que liga a rede ao *sink*, sofre do problema *energy sink-hole problem* descrito na Sessão 2.4.2. No instante de tempo T1, o nó C sofre uma queda de percentual de energia de 50%. Como esse percentual ultrapassa o *threshold* estabelecido, o nó envia uma mensagem *Reluctance Change* em *broadcast* informando seu novo valor de força de relutância. O nó B ao receber a mensagem, atualiza sua tabela de rotas e verifica que o nó C é seu pivô local. Como o nó C informou uma mudança no valor de sua força de relutância, o nó B recalcula sua própria força de relutância e também propaga através da mensagem *Reluctance Change* em *broadcast* seu novo valor de RF. Durante todo esse processo, o nó continua tendo a possibilidade de enviar mensagens do tipo *Send Message to Sink*, quando o meio de transmissão não está ocupado, pois não houve um rompimento da rede.

Observa-se que a propagação das atualizações não ocorre constantemente, mas apenas

quando é realmente de interesse para os demais nós da rede. Isso garante ao algoritmo uma grande redução no número de mensagens de controle enviadas, contribuindo para a redução do consumo de energia da rede.

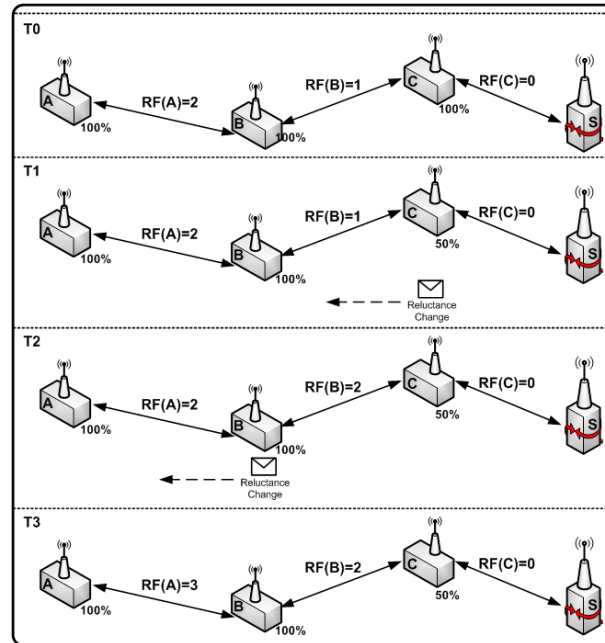


Figura 3.3: Propagação da alteração da força de relutância

3.3.5 Manutenção das Rotas

O processo de propagação da alteração da força de relutância, descrito na subseção 3.3.4, é um dos responsáveis pela manutenção das rotas entre os nós e o *sink*. Sempre que um nó propaga sua RF a seus vizinhos diretos, também está informando que esse nó é uma possível rota até o *sink*, conforme já ilustrado pela Figura 3.1. Caso seu RF possua um valor diferente de *INFINITO*, isso significa que o nó possui uma conexão até *sink* através de uma rota, ficando a cargo do nó que recebe esta atualização decidir qual de seus vizinhos possui a menor RF para eleger como pivô local.

Quando um nó informa a seus vizinhos que sua RF possui o valor igual a *INFINITO*, significa que este nó perdeu todas as suas conexões que o levavam até o *sink*, e portanto não possui mais um pivô local. Esse processo é o segundo mecanismo que realiza a manutenção das rotas. Conforme o Algoritmo 3.3.2, ao enviar uma mensagem de dados para o pivô local, o nó aguarda uma mensagem de *ACK* para confirmar o recebimento. Caso a mensagem de *ACK* não tenha sido recebida após um determinado período de tempo, indica que o nó destino não está mais operando para enviar e receber mensagens, com isso o algoritmo retira este nó de sua lista de vizinhos e recalcula o pivô local. Porém,

com o esgotamento da energia de uma grande quantidade de nós, pode ocorrer que este nó apesar de ainda possuir energia para se comunicar, não encontre nenhum vizinho que possua uma rota até o *sink*. Em ocorrendo esta situação, o nó propaga a seus vizinhos que sua RF possui o valor *INFINITO* o que os leva a também recalcularem seus pivôs locais. Ao recalcularem seu pivô, o nó encontra em sua tabela um novo vizinho com menor relutância e o elege para $M(x)$, desta maneira a mensagem que estava sendo roteada pelo nó que perdeu sua conexão, pode tomar um outro caminho até chegar ao *sink*. Caso não exista nenhum vizinho direto com relutância menor, o nó fica em estado de espera, armazenando as mensagens, até que receba uma mensagem de alteração da força de relutância de outro vizinho.

Desta forma a manutenção de rotas feita pelo algoritmo SP/WSN ocorre sem a necessidade de mensagens adicionais, reduzindo o consumo de energia para realizar a tarefa de controle do algoritmo.

3.3.6 Diagrama de estados do nó durante a execução do algoritmo

É importante entender os estados que cada nó assume e principalmente como ocorre a transição entre eles durante o funcionamento do algoritmo, sendo ilustrado pela Figura 3.4 com o digrama dos estados. O primeiro estado do nó no momento em que a rede entra em funcionamento é chamado de **espera** e ocorre quando o nó ainda não possui um pivô local. Considera-se que o estado de **espera** está contido em um estado mais amplo chamado **desconectado**. No momento em que o nó é informado de que houve a mudança da relutância de um vizinho, ocorre a transição do estado **desconectado** para **ativo**. Neste estado o nó já é capaz de gerar mensagens *Send Message to Sink* e também de propagar suas alterações de relutância pela mensagem *Reluctance Change*.

O estado **zumbi** indica que o nó ainda possui um percentual de energia residual em sua fonte de energia que o permite funcionar, porém este nó perdeu sua conexão com seu pivô local. Essa transição se dá de duas maneiras: (i) o nó pode realizar diversas tentativas de enviar uma mensagem *Send Message to Sink* para seu pivô local $M(x)$ e não receber a mensagem *ACK* que confirma o recebimento. Com isso, o nó x descobre que seu pivô local não está mais funcionando devido o esgotamento de sua fonte de energia. Então o nó x procura um novo vizinho em sua tabela de rotas e o elege como pivô local, o que o força a recalcularem sua RF e propagá-la. Caso o nó x não encontre nenhum vizinho em sua tabela de rotas, ele recalcula sua RF para o valor infinito e propaga essa informação, neste momento o nó entra no estado de **zumbi**. Isso indica que o nó está temporariamente

desconectado, portanto, ele armazena as mensagens *Send Message to Sink* que estavam sob sua responsabilidade para o roteamento e aguarda receber uma nova atualização de relutância de um vizinho. A segunda forma de transição para o estado **zumbi** ocorre em consequência da primeira transição já detalhada. Quando um nó recebe de seu pivô local que ele possui uma RF com o valor infinito e não possui outro nó em sua tabela de rotas, esse nó também se torna **zumbi** propagando que também sua RF tem o valor infinito.

Os nós **ativos** possuem mais uma transição que ocorre quando a fonte de energia do nó é totalmente esgotada. Quando isso ocorre o nó é completamente desligado sem a possibilidade de retornar a atividade, como indicado pelo diagrama de estados.

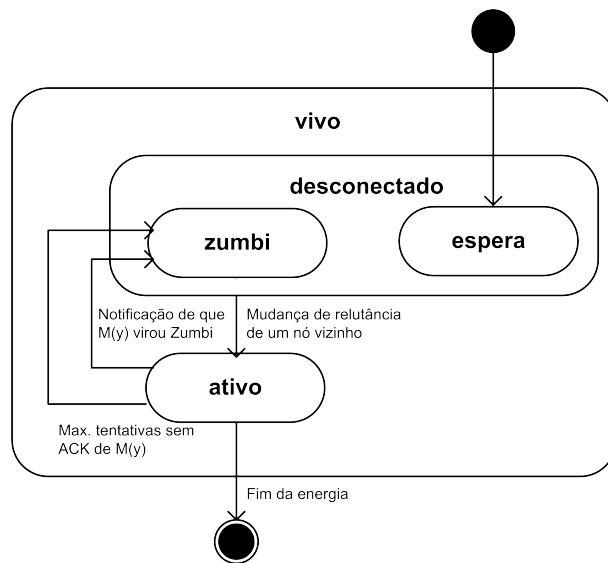


Figura 3.4: Diagrama de estados de um nó da rede

3.4 Considerações Finais

Neste capítulo foram apresentados os conceitos ligados ao algoritmo SP/WSN, descrevendo seu funcionamento, suas mensagens e seu comportamento durante o funcionamento da rede de sensores. No próximo capítulo será apresentado o método de validação adotado para o algoritmo SP/WSN e como sua simulação foi comparada ao algoritmo PDSR, pois ambos os algoritmos utilizam técnicas ativas de conservação de energia, definidas por (NIKOLAOS; DIMITRIOS, 2008).

Capítulo 4

Validação

4.1 Introdução

O método escolhido para validar a proposta apresentada nas seções anteriores foi o desenvolvimento dos algoritmos SP/WSN e PDSR dentro de um simulador de redes. Essa alternativa foi escolhida pois a utilização de um simulador de redes auxilia no desenvolvimento do algoritmo de roteamento, possibilita a utilização de parâmetros reais além de facilitar a avaliação dos resultados das simulações. Porém, o ponto mais importante está na garantia de que os algoritmos foram implementados e validados dentro de um ambiente que simula condições reais de uma rede.

4.2 Cenário

A Figura 4.1 ilustra um exemplo do modelo proposto para aplicação do algoritmo SP/WSN. Trata-se de uma rede de sensores com a quantidade fixa de nós para cada cenário de simulação. Os nós são aleatoriamente distribuídos dentro de uma área com limites definidos. Após distribuídos, os nós permanecem em seus lugares até o fim da simulação, pois não é considerado o fator mobilidade dos nós para este trabalho. Entre os nós não há obstáculos que possam gerar problemas de sombreamento ou redução do sinal.

Cada nó é composto por um dispositivo sensor contendo: *(i)* microcontrolador; *(ii)* memória ROM para armazenamento dos programas; *(iii)* uma unidade de memória RAM utilizada para o processamento das operações; *(iv)* um módulo de rádio frequência que

utilize o padrão IEEE 802.11b, com uma taxa de transmissão máxima de até 11Mbps e com um raio de alcance de transmissão de 134,94 metros, conforme valores calculados pelo plugin MiXiM considerando a parametrização realizada. Adotamos o padrão IEEE 802.11b por ser uma tecnologia de comunicação utilizada em vários estudos já publicados, e por não haver um processo de controle nativo de consumo de energia em sua especificação. Cada sensor possui sua fonte de energia própria sendo ela limitada, não recarregável e inicialmente 100% carregada. Caso ocorra o esgotamento total da fonte de energia do nó, o dispositivo se desligará por completo, não retornando ao funcionamento e perdendo todas as informações que continha. Por fim cada dispositivo é equipado com um sensor de temperatura com o objetivo de ser a fonte geradora de informação para o funcionamento da rede.

Existem dois tipos de nós na rede: os que são considerados fonte, que tem como tarefa realizar o sensoriamento da área, gerar pacotes com as informações coletadas pelos sensores e em seguida, definir através do algoritmo SP/WSN qual rota será adotada para realizar a transmissão nó-a-nó até o destino. Haverá um único nó destacado na Figura 4.1 com a letra "S" que será o *sink* (coletor dos dados): este nó é o ponto de destino para todas as mensagens geradas pela rede de sensores, da mesma forma, esse nó possui os mesmos atributos de configuração de hardware dos nós da rede, exceto pela restrição da fonte de energia que não se aplica a ele.

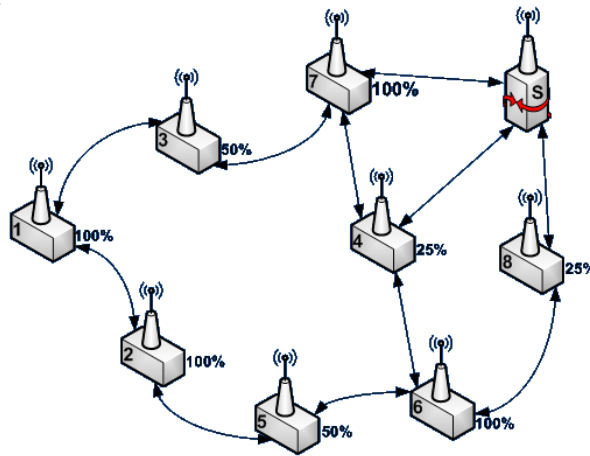


Figura 4.1: Topologia da rede de sensores

Na Figura 4.2, observa-se que cada nó da rede possui seu raio de transmissão, da mesma forma os nós possuem sua amplitude de recepção que é a distância a qual um nó x é capaz de receber informações corretas enviadas por outro nó y . Para o cenário proposto, supõe-se que a amplitude de recepção dos nós da rede nunca se altera, sendo essa amplitude ilustrada pelos círculos acinzentados na Figura 4.2.

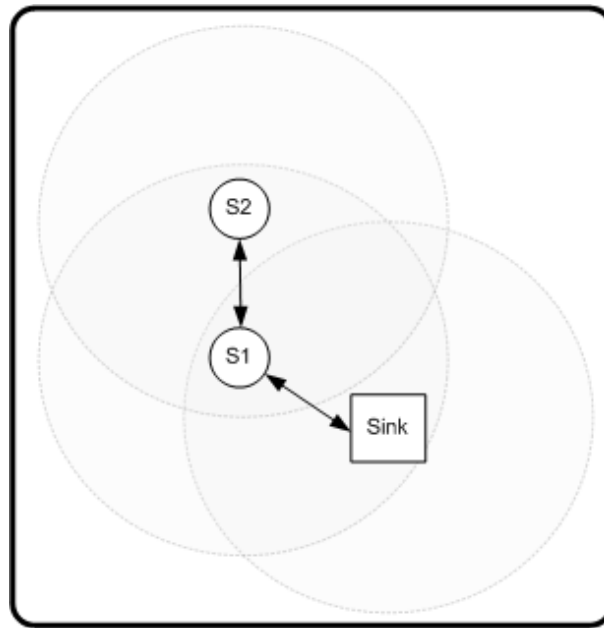


Figura 4.2: Área de transmissão e recepção entre nós

As setas descrevem as conexões estabelecidas pelos nós de acordo com o raio de transmissão e a amplitude de recepção de cada nó, assim considera-se que conexões são estáveis e sem interferências até que o raio de transmissão seja reduzido ou o posicionamento do nó afaste os transmissores a uma distância que impossibilite sua comunicação. A comunicação ocorre de forma bidirecional sempre que o raio de transmissão de um nó alcance um vizinho qualquer e a amplitude de recepção do vizinho alcance o nó transmissor, sendo esta situação representada pelas setas que indicam a comunicação nos dois fluxos.

4.2.1 Aspectos de Implementação

De acordo com (AKYILDIZ et al., 2002), a arquitetura de comunicação de um sensor pode ser observada na Figura 4.3, todo o processo de gerenciamento das tarefas, mobilidade do nó sensor e gerenciamento da fonte de energia está diretamente ligado às camadas da pilha de protocolo de comunicação do dispositivo sensor. Segundo (AKYILDIZ et al., 2002), dependendo da tarefa de sensoriamento, diferentes tipos de aplicação de software podem ser construídos e usados na camada de aplicação. A camada de transporte pode ajudar a manter o fluxo de dados se a aplicação necessitar. A camada de rede cuida do roteamento das informações suportada pela camada de transporte. Se houver interferência do ambiente e o sensor tiver a capacidade de se mover, o protocolo MAC deve estar preparado para reduzir a quantidade de colisões ao realizar broadcast de informações para os vizinhos. A camada física deve ser simples porém fortemente modularizada para

transmissão e recepção de informações.

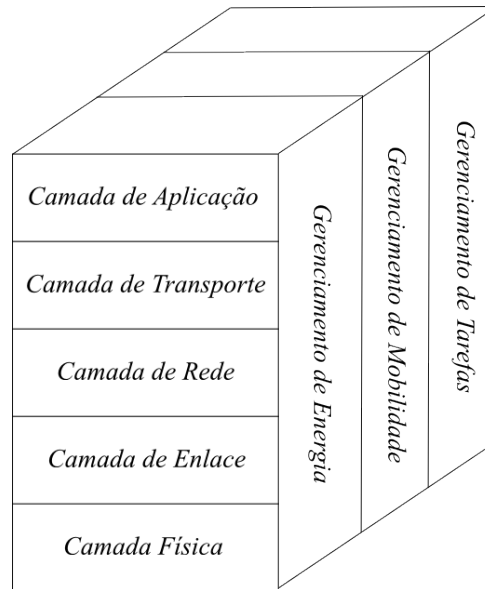


Figura 4.3: Pilha de protocolos de um sensor - Adaptado (AKYILDIZ et al., 2002)

Observa-se que há a possibilidade de implementação do algoritmo SP/WSN de duas maneiras distintas dentro do modelo OSI. A primeira proposta seria a implementação do algoritmo nas camadas de transporte e de rede, onde são feitas as decisões de roteamento dentro de uma arquitetura de rede. A segunda proposta consiste na implementação do algoritmo nas camadas de sessão, apresentação e aplicação do modelo OSI, como uma virtualização de uma rede sobre outro protocolo de comunicação já conhecido. Para a simulação do algoritmo SP/WSN foi adotada a abordagem onde as camadas de transporte e de rede foram modificadas para a implementação. Este método de implementação foi adotado por ser uma das formas mais comuns para algoritmos de roteamento em WSN.

4.2.2 Segurança em redes de sensores sem fio

Outro aspecto importante diz respeito a segurança da rede. Como descreve (AVIZIENIS et al., 2004), os itens: privacidade como detalhado por (KAMAT; ZHANG; OZTURK, 2005) e (LI et al., 2009), confiabilidade, recuperabilidade, confidencialidade, integridade, disponibilidade e irretratabilidade estão diretamente relacionados ao funcionamento de uma WSN. No momento em que os nós de uma WSN estiverem trocando informações, um nó invasor pode adentrar a rede e tentar capturar, substituir ou criar mensagens falsas para comprometer o funcionamento da rede. Ao capturar as mensagens, o nó invasor poderá quebrar os pacotes para ver seu conteúdo, analisar o identificador de origem e destino para rastrear o sequenciamento dos nós e por comparação encontrar a localização

física de um nó como descrito em (MEHTA; LIU; WRIGHT, 2007), (SHAO et al., 2008) e (LI; REN, 2009). Utilizando o processo de interceptação e substituição de pacotes, o nó invasor poderá modificar as mensagens de controle enviando valores inválidos, direcionando as mensagens para nós que possuem menor quantidade de energia residual na fonte de energia e acelerando o desligamento desses nós. Utilizando-se do mesmo artifício, um nó pode criar mensagens com o identificador de outros nós, se passando por eles, e enviando mensagens de atualização de dados sobre nós que já não estão mais em funcionamento, fazendo com que o roteamento direcione mensagens para pontos da rede sem saída. Todos os casos anteriores descritos, também podem ocasionar a quebra de integridade das mensagens trocadas, que pode afetar diretamente a disponibilidade da rede durante seu funcionamento. Por fim, um nó invasor pode gerar mensagens de controle e de aplicação em quantidade excessiva, gerando assim, um congestionamento de mensagens. Esses aspectos relevantes de segurança são conhecidos, mas seus desdobramentos e propostas de solução ficaram como objetivos de estudos futuros para este trabalho. Quanto às questões de falhas no funcionamento da rede como definido por (GEORGE DOLLIMORE JEAN, 2007), assume-se que não ocorrerão falhas por omissão, que consiste em um nó não realizar uma ação a qual estava prevista. Portanto, para este trabalho considera-se que não ocorrem falhas bizantinas tanto no canal de comunicação, quanto nos resultados das decisões de rotas do processo de roteamento do algoritmo proposto.

4.3 Simulador

Para a validação do SP/WSN foi escolhido o simulador OMNeT++. Segundo (VARGA, 2010) o OMNeT++ nasceu com o objetivo de servir como um simulador de redes de comunicação e sistemas distribuídos, mas foi modelado para ser utilizado como um simulador de propósito geral, possibilitando desta forma a simulação de filas de mensagens, redes Wireless, redes Ad-Hoc, redes peer-to-peer ou ainda processos de negócios. O OMNeT++ utiliza o conceito de componentes para possibilitar a simulação de todos os itens que compõem uma rede, desde dispositivos de rede, protocolos e até canais de comunicação, permitindo várias formas diferentes de combinação, como a customização dos módulos através do processo de herança. A definição dos módulos é feita através de um arquivo de definição com a extensão NED, parâmetros que influenciam no funcionamento da simulação podem ser externalizados através de arquivos de configuração com extensão INI. E a programação do comportamento dos módulos pode ser implementada utilizando as linguagens C++ e o paradigma de programação orientado a objetos, nativo no sim-

ulador, ou ainda em Java com a instalação de um plugin. Devido o propósito geral do simulador, foi utilizado também um plugin chamado MiXiM, que segundo (KÖPKE et al., 2008) adiciona características para simulação de redes sem fio e a mobilidade dos nós da rede, possibilitando simulações mais realistas no contexto de redes de sensores sem fio. O MiXiM possui diversos exemplos de simulações prontas que podem ser modificadas ou estudadas para entender o funcionamento de sua API.

4.4 Implementação do SP/WSN no simulador

A implementação do algoritmo no simulador inicia com a criação de um novo projeto no IDE do OMNet++ que é baseado na plataforma Eclipse, sendo possível importar o contexto de uma *Basic MiXiM network* onde o projeto já recebe um conjunto de arquivos de configuração mínimo para iniciar o desenvolvimento. O próximo passo é definir o conteúdo do arquivo BaseNode.ned, a Figura 4.4 ilustra a representação gráfica do arquivo NED que define os componentes base do nó da rede de sensor proposta para esse trabalho. Observa-se a pilha do protocolo, que define as camadas de comunicação. Essa estrutura é composta por apenas por três componentes: camada física que já incorpora a camada MAC, camada de rede e a camada de aplicação. Para a camada física, foi utilizado o componente *Nic80211Battery*, que prove a camada MAC e o dispositivo físico que implementa o padrão IEEE 802.11b, mas principalmente possibilita que a simulação monitore a quantidade de energia consumida pelo rádio transmissor. No arquivo de configuração INI do simulador, é possível definir parâmetros como: tamanho do cabeçalho dos pacotes, potência necessária para transmitir pacotes, sensibilidade do rádio transmissor para receber informações, potência máxima do transmissor, tempo de desligamento do rádio transmissor, dentre outros parâmetros. Na camada de aplicação, foi utilizado o módulo *SensorApplLayer*, que permite a geração de mensagens simulando o funcionamento de um nó sensor. Esse módulo permite a configuração de métodos diferentes de geração de mensagens como: periódico, uniforme ou exponencial. Destaca-se ainda os componentes *SimpleBattery* e *BatteryStats*, esses dois módulos são integrados e simulam o funcionamento de uma bateria como fonte de energia do nó, e o módulo que permite o monitoramento da drenagem da energia da bateria. Apesar da simplicidade dos componentes, eles permitem configurar informações como: voltagem da bateria, capacidade em miliampere, número de baterias e o tempo em que as informações sobre energia residual serão atualizados para o nó. A camada de rede foi customizada para a implementação do algoritmo SP/WSN, estendendo o módulo *BaseNetwLayer*. Neste módulo é possível

receber através de métodos mensagens provenientes dos módulos de aplicação e físico, além de receber eventos publicados por outros componentes do nó, como a atualização do status de carga de energia da bateria. O módulo *arp* é responsável por realizar a tradução dos endereços lógicos de cada nó, gerenciado pelo próprio simulador, e convertê-los para endereços MAC necessários para o envio de mensagens. Por fim, o módulo *utility* funciona como um facilitador para a troca de informações entre os demais módulos que compõem um nó da rede. Seu princípio é o mesmo de um serviço de notícias *publish/subscribe*, todos os componentes podem se registrar no *utility* gerando eventos com informações ou aguardando por determinados eventos gerados por outros módulos. A adoção do *design pattern publish/subscribe* pelo *utility* simplifica em muito a troca de informações, e foi amplamente utilizado para receber informações do módulo *BatteryStats* a respeito do percentual residual de energia.

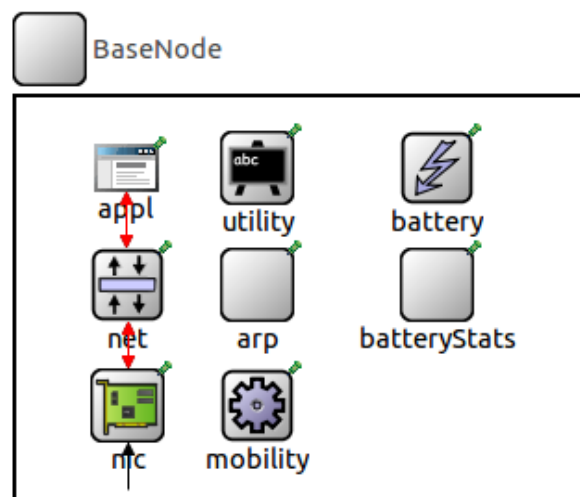


Figura 4.4: Arquivo NED de definição do nó da rede de sensores.

O próximo passo é configurar os objetos que compõem a rede através do arquivo criado automaticamente pelo simulador com o mesmo nome do projeto com a extensão NED, ilustrada pela Figura 4.5. Os arquivos de configuração NED podem ser criados usando a forma gráfica através do processo de arrastar componentes de uma palheta já definida, como também através da linguagem de definição ilustrada no item (b) da Figura 4.5. A linguagem NED permite uma liberdade maior ao desenvolvedor, possibilitando a criação de variáveis e a instanciação de objetos já existentes na biblioteca do OMNet++ e MiXiM ou a utilização de objetos desenvolvidos de forma customizada. Pode-se observar na figura dois componentes obrigatórios para todas as redes criadas no simulador OMNet++: *world* que é responsável por parâmetros globais da simulação e o *connectionManager* que gerencia a conexão entre os nós da rede. É importante destacar que no item (b) da Figura

4.5 o código define que existem nós na rede que são definidos pelo arquivo *BaseNode*, e utiliza a notação de vetor para informar ao simulador que a quantidade de objetos representando os nós não é fixa, sendo definida pela variável *numNodes*. Isso permite que diversos cenários de simulações com quantidades diferentes de nós possam ser facilmente configurados através do arquivo de parametrização do simulador chamado *omnetpp.ini*.

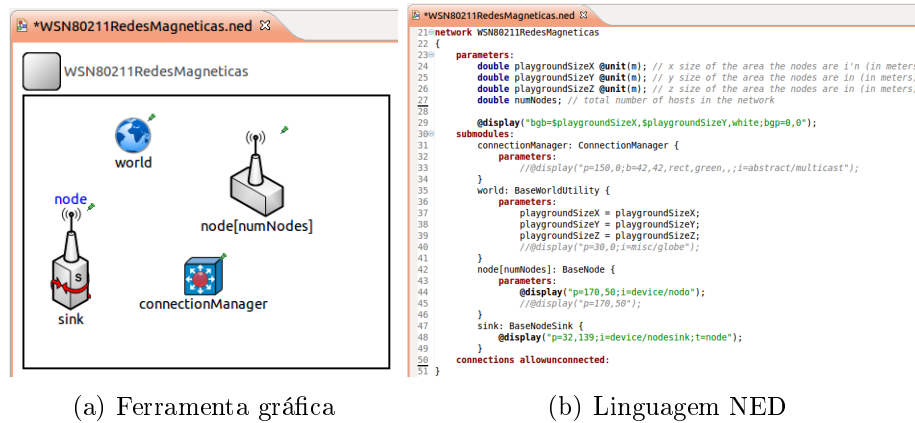


Figura 4.5: Arquivo NED de definição da rede

O próximo passo é a criação das mensagens que serão transmitidas pelos nós da rede, sendo que sua definição é feita através de um arquivo com extensão ponto "m" utilizando um código C++ muito semelhante a uma *struct* da linguagem C. O simulador, através da linguagem C++, permite uma grande liberdade na utilização dos tipos de dados para os atributos que são enviados por uma mensagem, possibilitando por exemplo, o envio de uma estrutura de mapas genéricos como o *std::map<>*. Existem dois tipos básicos de mensagens representadas pelas classes *cMessage* e *cPacket*, respectivamente, o primeiro tipo consiste de um tipo básico de mensagem, já os segundo permite o processo de encapsulamento de mensagens geradas por camadas superiores na pilha de protocolo. Como o algoritmo SP/WSN foi desenvolvido sobre a camada de rede, utilizou-se o tipo de mensagem *cPacket* para o encapsulamento das mensagens de dados/aplicação geradas pelas camadas superiores. A modificação do cabeçalho desses pacotes é que permite o roteamento das mensagens, modificando o endereço de destino.

O último e mais importante passo é a implementação do algoritmo no simulador. Para isso, é necessário criar uma nova classe que especialize o comportamento que representa a camada da pilha do protocolo. Como ilustrado pela Figura 4.6 com o diagrama da classe *BaseNetwLayer*, esta classe abstrata possui os métodos necessários para a implementação da camada de rede de uma pilha de protocolo que são:

- *virtual void handleUpperMsg (cMessage *msg)* - responsável por receber as men-

sagens geradas por camadas superiores da pilha de protocolo;

- *virtual void handleLowerMsg (cMessage *msg)* - responsável por receber as mensagens recebidas pelas camadas inferiores da pilha de protocolo;
- *virtual void handleSelfMsg (cMessage *msg)* - responsável por receber mensagens agendadas pela própria camada, permitindo a implementação de processos de re-tentativa de envio de mensagens;
- *virtual void handleLowerControl (cMessage *msg)* - responsável por receber mensagens de controle recebidas por camadas inferiores da pilha de protocolo;
- *virtual void handleUpperControl (cMessage *msg)* - responsável por receber mensagens de controle geradas por camadas superiores da pilha de protocolo;
- *virtual cMessage * decapsMsg (NetwPkt *)* - responsável por desencapsular as mensagens recebidas de camadas inferiores da pilha de protocolo;
- *virtual NetwPkt * encapsMsg (cPacket *)* - responsável por encapsular as mensagens recebidas antes de enviar para camadas inferiores da pilha de protocolo.

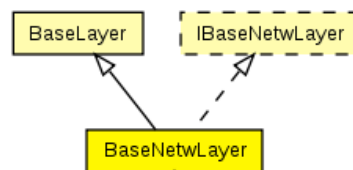


Figura 4.6: Diagrama de classes BaseNetwLayer

O método *handleSelfMsg* foi utilizado para implementar rotinas de tentativa de reenvio de mensagens que são necessárias quando o algoritmo recebe uma solicitação de envio de mensagem da camada de aplicação, porém ainda não possui um pivô local. Essa mensagem de aplicação é agendada, através do método *scheduleAt()*, para ser tratada pelo nó após um determinado período de tempo, e o simulador automaticamente gerencia esses eventos. Para o SP/WSN, foi parametrizado que o processo de reenvio de mensagens de aplicação deve realizar três tentativas, enquanto não houver rotas, e posteriormente descartar a mensagens.

4.5 Implementação do PDSR no simulador

Os mesmos passos descritos na Sessão 4.4 foram realizados para a implementação do algoritmo PDSR no simulador. A principal diferença que pode ser destacada na implementação do PDSR se dá devido a característica do protocolo ser proativo, desta maneira o algoritmo precisa realizar o processo de descoberta de rota, o que implica não apenas no envio de uma mensagem mas no armazenamento do endereço de rede dos nós pelo qual a mensagem passa. No momento em que a mensagem de descoberta de rota atinge o *sink*, a mesma mensagem precisa retornar pela rota que descobriu para voltar ao nó origem. Durante a implementação desta rotina de descoberta de rota, observou-se que constantemente um percentual de mensagens entrava em *loop*, o que aumentava em muito o tempo total para a execução de um cenário de simulação. No artigo original que descreve o PDSR de (WEI et al., 2004), não foi detalhado nenhum mecanismo para evitar problemas de *loop* no algoritmo. Já no trabalho que deu origem ao PDSR, o DSR de (JOHNSON; MALTZ, 1996), detalha alguns mecanismos como tabelas em cada um dos nós que armazenam identificadores de mensagens de descoberta de rota com o objetivo de evitar que sejam retransmitidas mais de uma vez.

Outra diferença importante do PDSR é a mensagem de controle chamada RPEP (*Route Priority Error Package*), seu objetivo é impedir que uma rota seja utilizada excessivamente, consumindo rapidamente o recurso de energia dos nós que a compõe. Esta mensagem é enviada a um nó origem que enviou uma mensagem de dados por uma determinada rota. A cada salto pelos nós, o algoritmo compara a prioridade da rota selecionada que foi gravada no cabeçalho da mensagem de dados com a atual prioridade calculada pelo nó que está retransmitindo a mensagem com base em seu percentual residual de energia. Caso a prioridade do nó possua um valor menor que a prioridade da rota que está em uso, a mensagem de dados não é retransmitida para o próximo nó da rota, mas sim uma nova mensagem RPEP é enviada com a cópia da rota utilizada para chegar até o nó origem da mensagem de dados. Quando a mensagem chega ao nó origem, a rota que gerou erro é removida da tabela de rotas do nó e um novo processo de descoberta é iniciado.

Da mesma forma que o SP/WSN o PDSR também implementa a rotina de tentativa de reenvio de mensagens geradas pela camada de aplicação, sendo que esse processo realiza três tentativas de envio, quando não existem rotas disponíveis até o *sink*, antes de descartar a mensagem.

Parâmetro	Valor
Capacidade Bateria	700mAh
Voltagem	1.5v
Taxa de Atualização da Carga Residual	10s
Potência de Transmissão	100.0mW
Potência Máxima de Transmissão	100.0mW
<i>Threshold</i> envio mensagem Reluctance Change	2.5%

Tabela 4.1: Parametrização da simulação

Atividades do Nó	Quantidade de Energia Consumida
<i>Sleep</i>	5mA
Transmissão de Dados	20mA
Recepção de Dados	20mA

Tabela 4.2: Parametrização do Consumo de Energia do Nó

4.6 Parametrização da Simulação

Os nós da rede foram parametrizados conforme a Tabela 4.1, sendo que esses valores são os sugeridos pelo plugin MiXiM para WSN. Cada nó da rede possui uma fonte de energia limitada e não renovável, com exceção do nó *sink* que possui uma fonte de energia infinita. A taxa de atualização da carga residual da bateria consiste da frequência de tempo que o módulo da bateria gera eventos para o nó da rede atualizando sua quantidade de energia. Além disso, foi aplicado um percentual de *threshold*, apenas no algoritmo SP/WSN, que define quando a drenagem da bateria é relevante. Este valor de *threshold* foi escolhido através de testes da simulação considerando o tempo total de execução, pois dependendo do número de mensagens geradas durante a simulação o tempo total para um cenário pode ser aumentado ou reduzido. Porém cabe destacar a necessidade de estudos mais aprofundados sobre o impacto da alteração deste *threshold* sobre o desempenho do algoritmo SP/WSN. Também é importante observar que os nós transmissores trabalham com sua potência máxima durante todo o processo de funcionamento da rede.

Com relação ao consumo de energia, a Tabela 4.2 apresenta as quantidades consumidas da fonte de energia de acordo com a atividade realizada pelo nó. As quantidades são informadas na unidade de medida miliampere e utilizadas na configuração do módulo simulador da fonte de energia, que se encarrega de calcular o consumo com base nos dados. Os valores utilizados são os sugeridos pelo simulador MiXiM, através de exemplos de ambientes de simulação de WSN já existentes para estudo.

Base	Altura	Número de Nós
870	870	100
1231	1231	200
1468	1468	300
1666	1666	400
1838	1838	500

Tabela 4.3: Parametrização área do cenário de simulação com densidade constante

Foram definidos dois cenários diferentes para as simulações realizadas com os dois algoritmos, a saber:

- Cenário I: Neste cenário, varia-se a quantidade de nós distribuídos sobre uma área fixa, ou seja, varia-se a densidade de nós. Foi utilizada uma área de teste de 1000m x 1000m para todas as simulações, variando o número de nós dentro desta área, iniciando com 100 até 500 nós.
- Cenário II: Neste cenário, varia-se a quantidade de nós distribuídos sobre uma área também variável, tal que, a densidade seja constante. O cálculo da área para cada quantidade de nós é feito de forma similar a (FREITAS et al., 2010), que utiliza a Fórmula 4.1, definida por (BEN-ASHER MORAN FELDMAN; GURFIL, 2008), que garante que existe um percentual mínimo de conectividade entre os nós, independentemente da disposição física dos nós. Na Fórmula 4.1, $P(d_{min} > 0)$ é o grau mínimo de probabilidade de conexão entre os nós, ρ é a densidade, r é o raio de alcance de transmissão e n é o número de nós da rede. O valor do raio de alcance de transmissão, foi obtido através do simulador OMNeT++: 134,94m. A Tabela 4.3 mostra as medidas de base e altura de uma área retangular para a distribuição de 100 a 500 nós.

$$P(d_{min} > 0) = (1 - e^{-\rho\pi r^2})^n \quad (4.1)$$

Em cada um dos cenários, para cada uma das quantidades de nós na rede simulada, foram executadas 100 vezes a simulação, utilizando um método aleatório para a disposição dos nós dentro da área de testes. Sendo que o posicionamento do nó *sink* também foi feito de forma aleatória a cada simulação. O simulador OMNeT++ permite dois modos de execução das simulações: modo gráfico ilustrado pela Figura 4.7, e o modo em linha de comando. No modo gráfico o simulador permite acompanhar visualmente o funcionamento do algoritmo; devido o fato de ser um simulador de eventos discretos, é possível acompanhar passo a passo a execução dos eventos e o envio de mensagens que no ambiente

real acontecem de forma simultânea. No modo em linha de comando o simulador pode chegar a executar o algoritmo até cinco vezes mais rápido que o modo gráfico, pois apenas algumas linhas de log são impressas no console. Nos dois métodos é possível realizar o *debug* do algoritmo, ou ainda habilitar a geração de registro dos eventos em log que permite a geração de um diagrama de sequência para acompanhar todos os eventos a cada segundo durante a execução do simulador.

Os dados coletados foram processados para o cálculo da média dos valores obtidos a fim de evitar desvios dentro do processo de avaliação dos resultados.

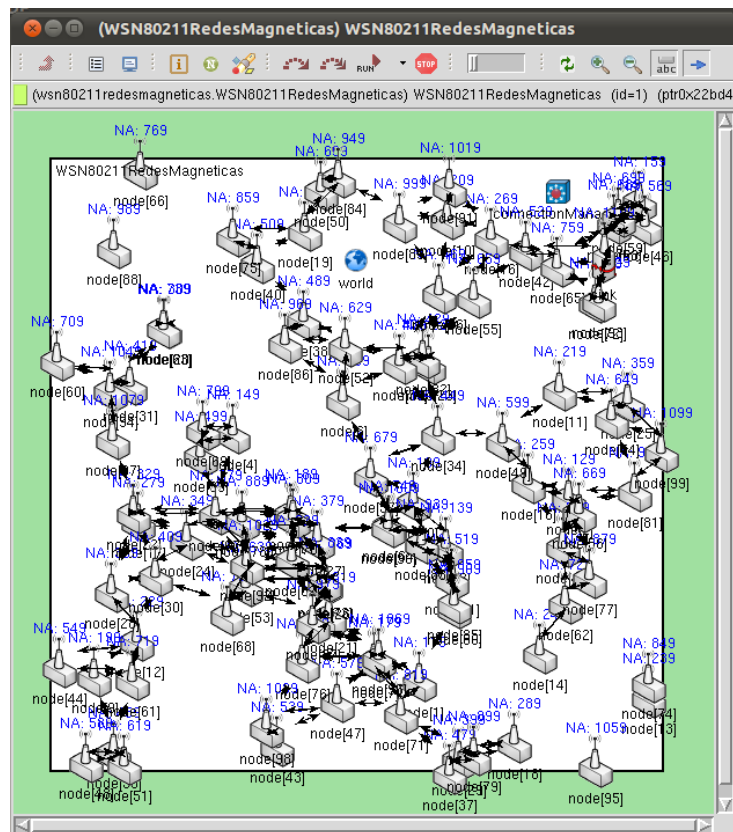


Figura 4.7: Algoritmo SP/WSN em execução, com 100 nós, no simulador OMNet++ com o MiXim

4.7 Resultados

4.7.1 Taxa de Entrega de Mensagens

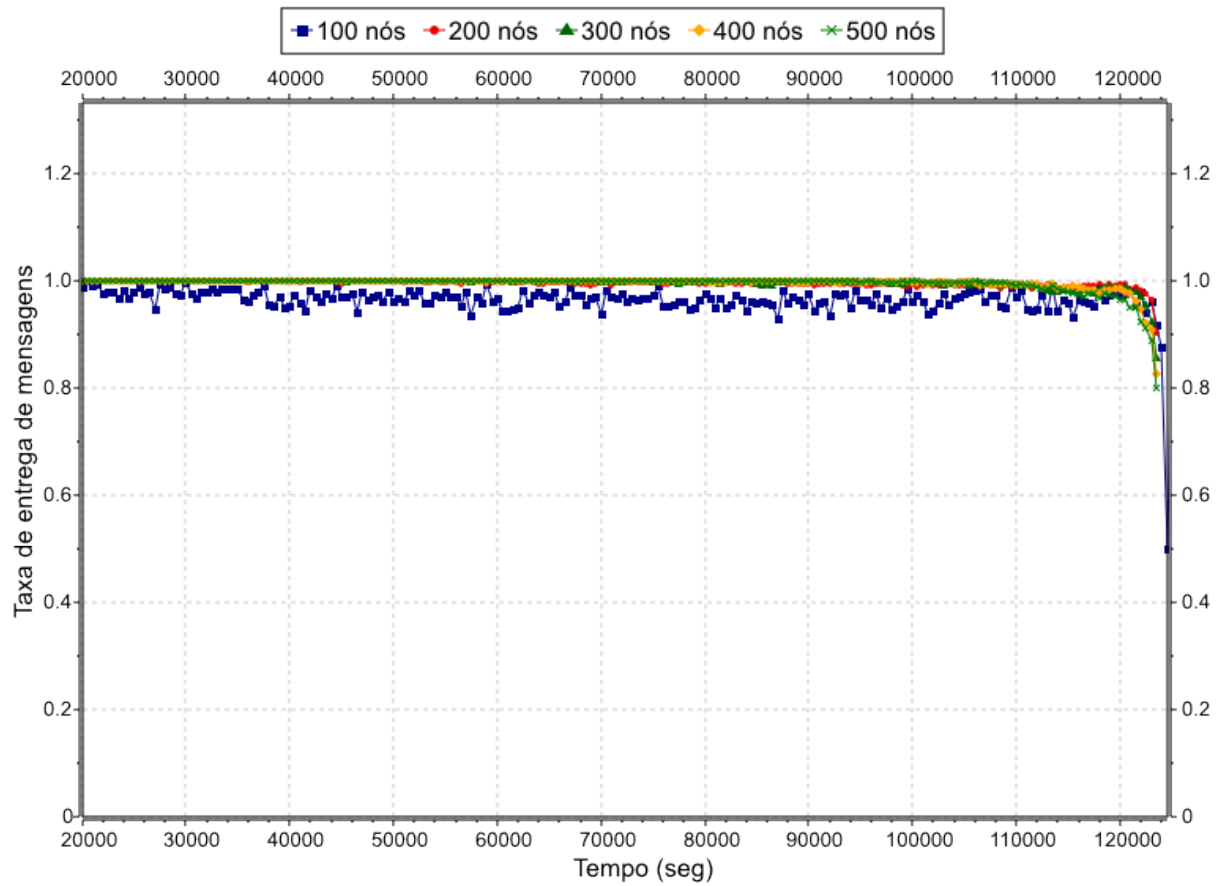
O primeiro indicador gerado foi a taxa de entrega (*delivery ratio*) das mensagens de aplicação geradas pelos nós. Esse indicador mostra o percentual de mensagens entregues com sucesso ao *sink*. O principal motivo para a redução da taxa de entrega é a inexistência

de rotas devido: (i) isolamento de um nó de forma definitiva, fato que não permite que ele estabeleça rotas; (ii) desconexão temporária, devido todas as suas rotas serem invalidadas. Para contornar a desconexão temporária, foi adotado para o SP/WSN e o PDSR o número três tentativas de reenvio de mensagens de aplicação.

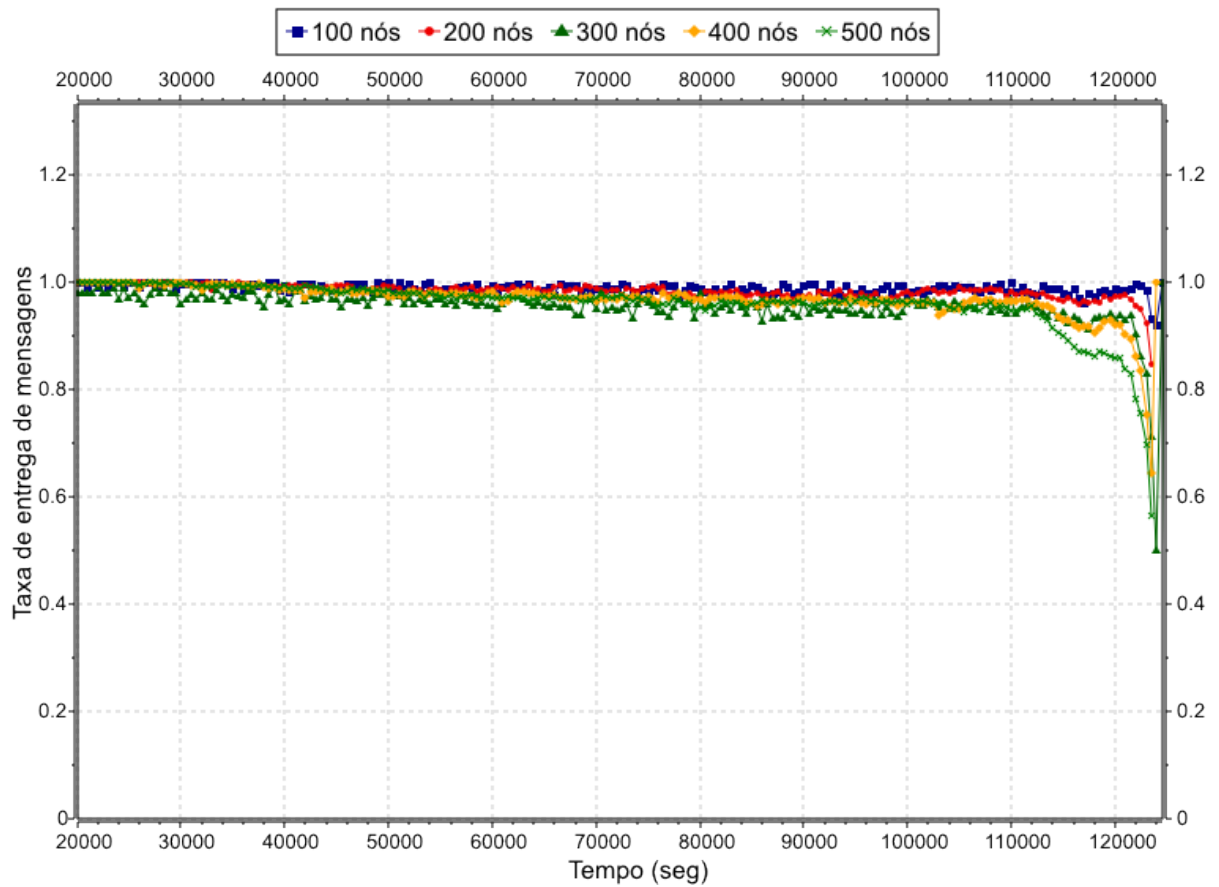
O cálculo da taxa de entrega é realizado da seguinte forma: são totalizadas todas as mensagens geradas por todos os nós ativos da rede dentro de espaços de tempo de 500 segundos correspondente a hora de envio da mensagem; as mensagens que chegam ao *sink* são também totalizadas de acordo com os espaços de tempo correspondentes a hora de envio da mensagem; então é calculado, por espaço de tempo, a taxa de entrega das mensagens. A Figura 4.8 item (a) apresenta o resultado da simulação no Cenário I (densidade variável) com o SP/WSN. Observa-se que a taxa de entrega de mensagens permaneceu muito próxima de 100% durante quase todo o tempo em que a rede estava operacional, isto é, enquanto havia energia nos nós. É importante destacar que com o aumento da densidade de nós, devido ao aumento da quantidade de nós no cenário, houve uma pequena melhora na taxa de entrega, isso ocorre porque uma maior densidade permite um maior número de conexões entre os nós facilitando a criação de novas rotas e, conseqüentemente, melhora a taxa de entrega.

A Figura 4.8 do item (b) apresenta o resultado da simulação do Cenário II (densidade fixa) com o SP/WSN. Observa-se que a taxa de entrega permanece próxima a 100% em todas as quantidades de nós simuladas por quase todo o tempo de funcionamento da rede, porém a queda na taxa de entrega ocorre mais precocemente do que no Cenário I. Isso ocorre porque os nós estão mais bem distribuídos, gerando rotas com um maior número de saltos, o que ocasiona um consumo maior de energia, e o desligamento dos nós mais precocemente. Isso mostra que o algoritmo consegue manter uma alta taxa de entrega das mensagens, balanceando o consumo de energia por todos os nós, o que acarreta no prolongamento do tempo de funcionamento total da rede.

Com o algoritmo PDSR, obteve-se um resultado diferente para o indicador da taxa de entrega. Observa-se na Figura 4.9 que em ambos os cenários a taxa de entrega ficou abaixo do obtido com o SP/WSN. No Cenário I, item (a), a taxa de entrega melhorou sensivelmente com o aumento da densidade de nós na rede. É possível observar a influência do processo de descoberta de rotas reativo sobre a taxa de entrega, como inicialmente boa parte dos nós consegue definir uma rota, a taxa de entrega é alta, porém com o passar do tempo a energia dos nós é consumida e as rotas são destruídas devido a queda da prioridade dos nós. Com isso é necessário reiniciar o processo de descoberta de rotas, que, até sua conclusão, gera substanciais quedas na taxa de entrega. Já no Cenário II,



(a) Cenário I



(b) Cenário II

Figura 4.8: Taxa de entrega x tempo - SP/WSN

item (b), observa-se que a taxa de entrega permaneceu muito similar não importando o número de nós da rede. Isso indica que o algoritmo PDSR também sofre influência em seu desempenho de acordo com a densidade de nós utilizadas, porém o algoritmo SP/WSN mantém uma taxa de entrega superior, mesmo com a variação da densidade.

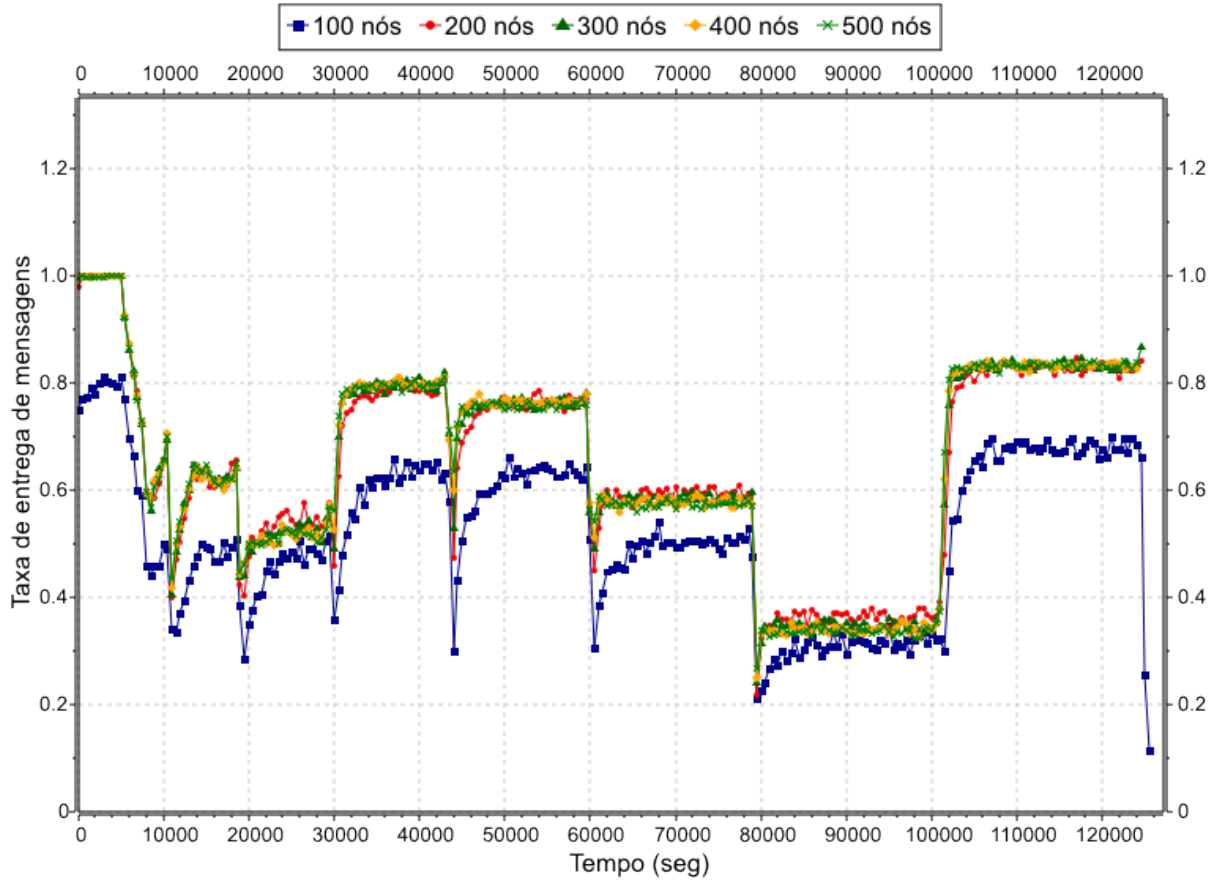
Observa-se que mesmo com o processo de tentativas de reenvio de mensagens, o PDSR não obteve a mesma taxa de entrega obtida com o SP/WSN. É importante destacar que o aumento no número de tentativas de reenvio pode modificar o resultado da taxa de entrega, sendo que esse número de tentativas para o algoritmo PDSR alcançar uma taxa de entrega ideal (próximo a 100%) não foi estudado. Porém, é relevante que o SP/WSN, mesmo com um baixo número de tentativas de reenvio, obteve um resultado melhor em comparação ao PDSR.

Outro ponto em comum no resultado da taxa de entrega para o PDSR nos cenários I e II, é um desvio abrupto da curva logo no início, isso ocorre devido a intensificação do processo de descoberta de rotas pelos nós da rede; a inundação de mensagens de controle acaba por reduzir a taxa de entrega de mensagens de dados.

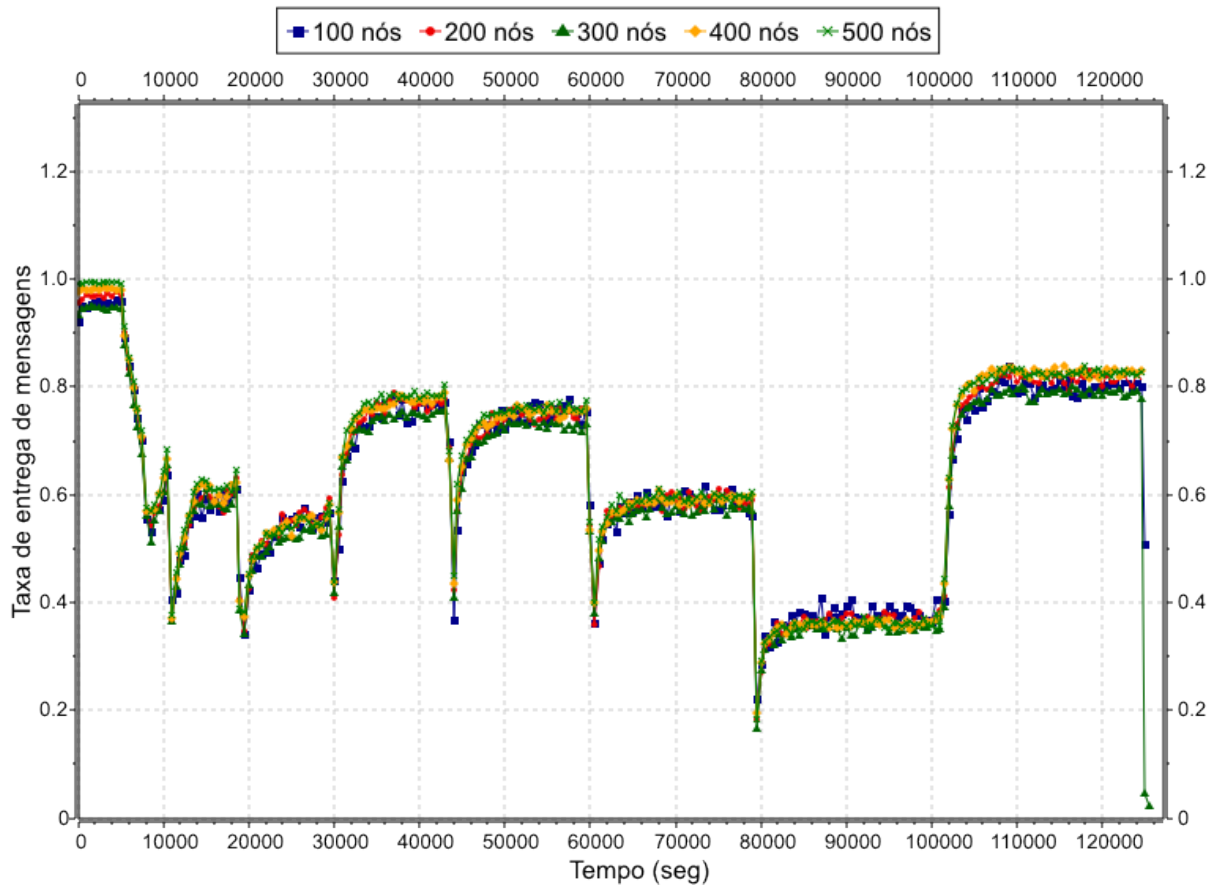
É possível concluir com este resultado que o SP/WSN comparado ao PDSR, possui uma taxa de entrega com índice melhor, alcançando o primeiro objetivo desejado para o algoritmo que é garantir a cobertura da rede.

4.7.2 Atraso na Entrega de Mensagens

O Atraso na Entrega de Mensagens apresenta o tempo médio de demora para a entrega de uma mensagem dentro de intervalos de 5000 segundos. Ele é calculado no momento de chegada das mensagens no *sink*, através da soma dos tempos que cada mensagem levou para chegar ao *sink* dividido pelo número de mensagens que foram entregues com sucesso, para então aplicar a média de todos os resultados das 100 simulações. Para melhor compreender o resultado deste indicador, a Figura 4.10 ilustra um esquema que representa como esse indicador é gerado. O tempo total da simulação é representado pela linha, que foi seccionada em intervalos de 5000 segundos do simulador (representado por: S0, S1, S2, S3 e S4). Dentro desses intervalos de tempo, os nós da rede enviaram mensagens de aplicação de acordo com os dados coletados pelos sensores. O tempo que uma mensagem leva para ser enviada por um nó da rede até chegar ao *sink* é calculado no momento da chegada da mensagem no destino, através da subtração do tempo do simulador no nó *sink* do tempo de envio gravado na mensagem. É importante destacar



(a) Cenário I



(b) Cenário II

Figura 4.9: Taxa de entrega x tempo - PDSR

que o simulador garante a sincronização dos relógios internos de todos os nós da rede. Algumas mensagens geradas podem ser entregues dentro do intervalo onde foram criadas, como por exemplo: M1, M4, M8. Portanto, seus tempos de entrega são computados dentro do intervalo onde a mensagem foi entregue. Já algumas mensagens não conseguem ser entregues dentro do mesmo intervalo de tempo onde foram criadas como: M2, M3, M5, M6 e M7, sendo computadas no instante em que chegam ao destino. E, por fim, dentro do instantes S2 e S3 ocorreram mensagens que nunca foram entregues devido a falta de uma rota para chegar até o nó *sink*. As mensagens destacadas em tracejado não são computadas por este indicador, devido o fato de não serem entregues. A Figura 4.11,

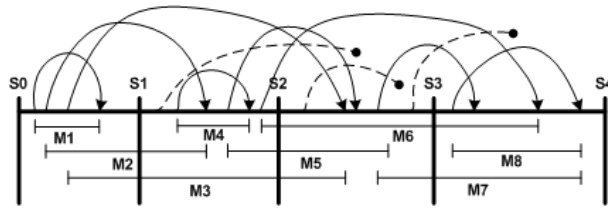
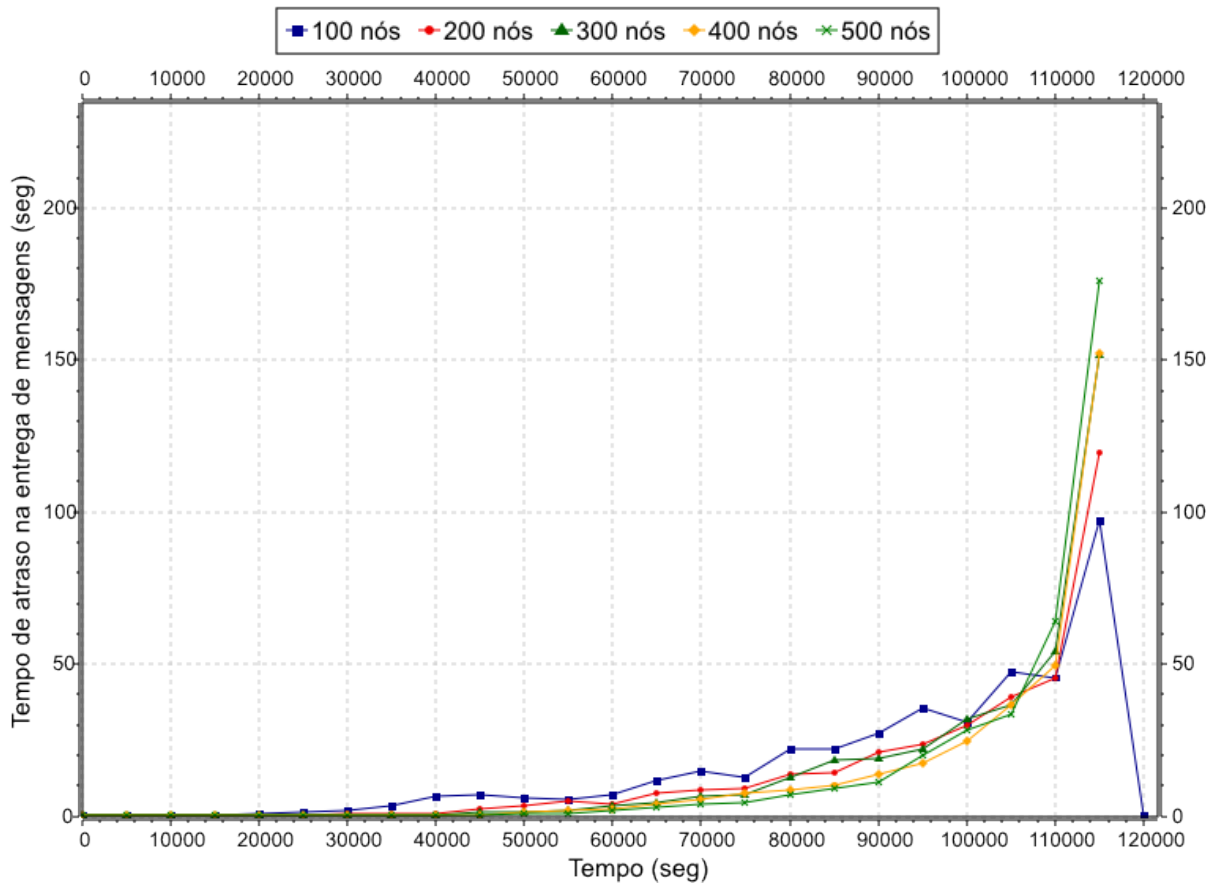


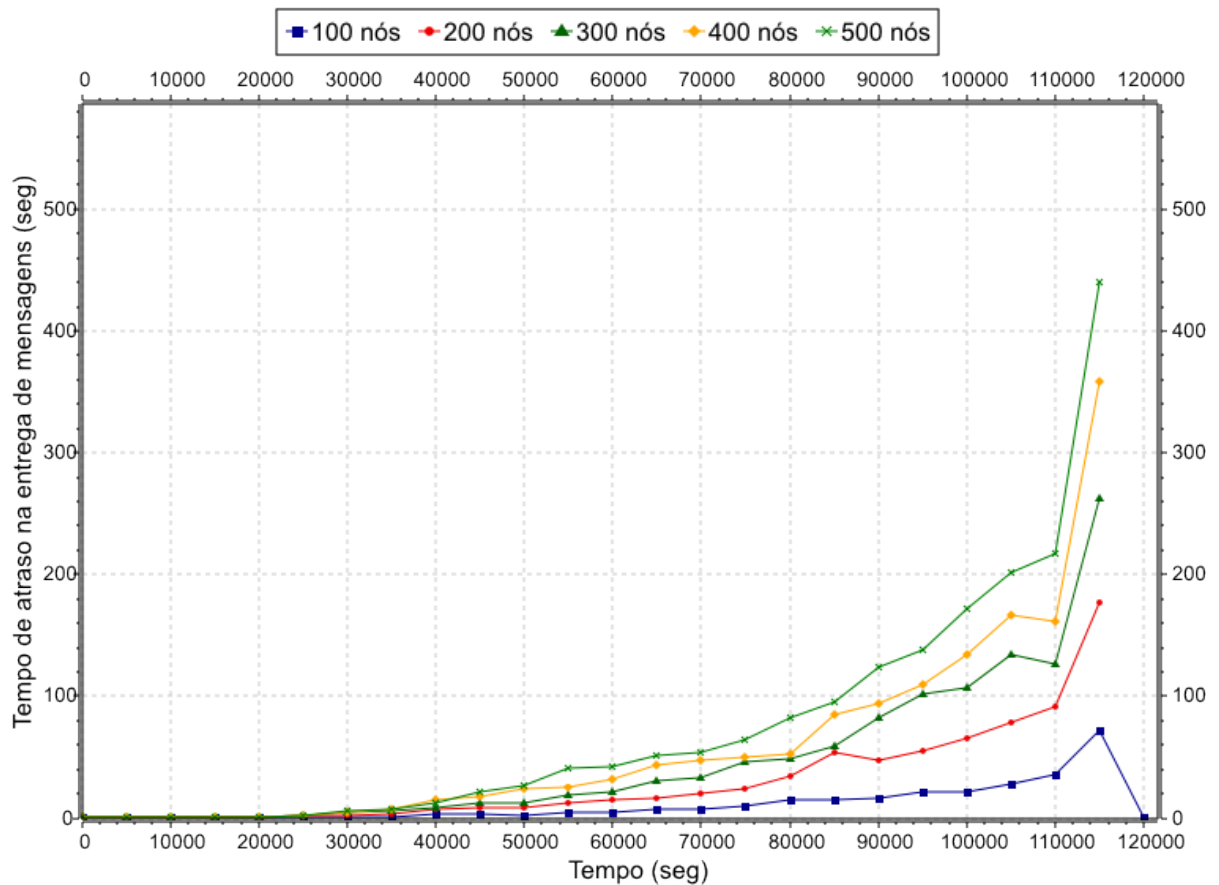
Figura 4.10: Agrupamento dos nós por estados - SP/WSN

item (a), apresenta o resultado no Cenário I para o SP/WSN, onde observa-se que o atraso na entrega permanece muito próximo de zero na maior parte do tempo de funcionamento da rede. O crescimento do atraso ocorre próximo do momento onde praticamente todos os nós da rede estão sem energia, fato este que reduz o número de rotas e gera um maior atraso. No Cenário II, item (b), o atraso na entrega começa a aumentar antes, em comparação com o Cenário I, pois a densidade fixa induz uma melhor distribuição dos nós que acaba criando uma maior distância entre eles e aumentando o número de saltos necessários para chegar até o *sink*, o que acarreta por aumentar o tempo de atraso na entrega. Observa-se que nos dois cenários, o algoritmo SP/WSN manteve um nível de atraso baixo, na maior parte do tempo de simulação, o que demonstra um desempenho adequado para o funcionamento em redes de sensores sem fio devido o baixo número de mensagens de controle necessárias.

O algoritmo PDSR, conforme a Figura 4.12, mostra nos Cenários I e II um elevado tempo de atraso inicial devido ao processo de descoberta das rotas utilizado pelo algoritmo. Esse atraso é reduzido no intervalo 2, onde nos dois cenários, ocorre a estabilização da rede. Observa-se no decorrer da simulação pequenos desvios no tempo de atraso ocorridos pela desestabilização da rede, devido o cancelamento de rotas que é ocasionado pela redução da carga de energia dos nós. Próximo do fim da simulação, da mesma forma como ocorre no SP/WSN, surge um significativo aumento no atraso em decorrência da morte de diversos nós o que limita o número de rotas e intensifica a realização do processo de descoberta de



(a) Cenário I



(b) Cenário II

Figura 4.11: Atraso na entrega de mensagens - SP/WSN

novas.

Em comparação os dois algoritmos tiveram bons resultados no indicador de atraso de entrega. O comportamento observado está diretamente ligado aos mecanismos de manutenção de rotas utilizados pelos algoritmos. O PDSR obteve um tempo de atraso na entrega de mensagens menor, na maior parte do tempo de simulação em comparação ao SP/WSN. Observando o resultado da taxa de entrega juntamente com o atraso para o algoritmo PDSR, pode-se supor que o menor tempo de atraso na entrega de mensagens se deu porque apenas mensagens geradas muito próximas ao *sink* foram entregues, e devido a proximidade seu atraso foi pequeno. Já mensagens geradas mais distantes do *sink*, acabaram se perdendo durante o processo de roteamento, uma vez que o número de tentativas foi limitado a três, e não foram computadas nesse processo.

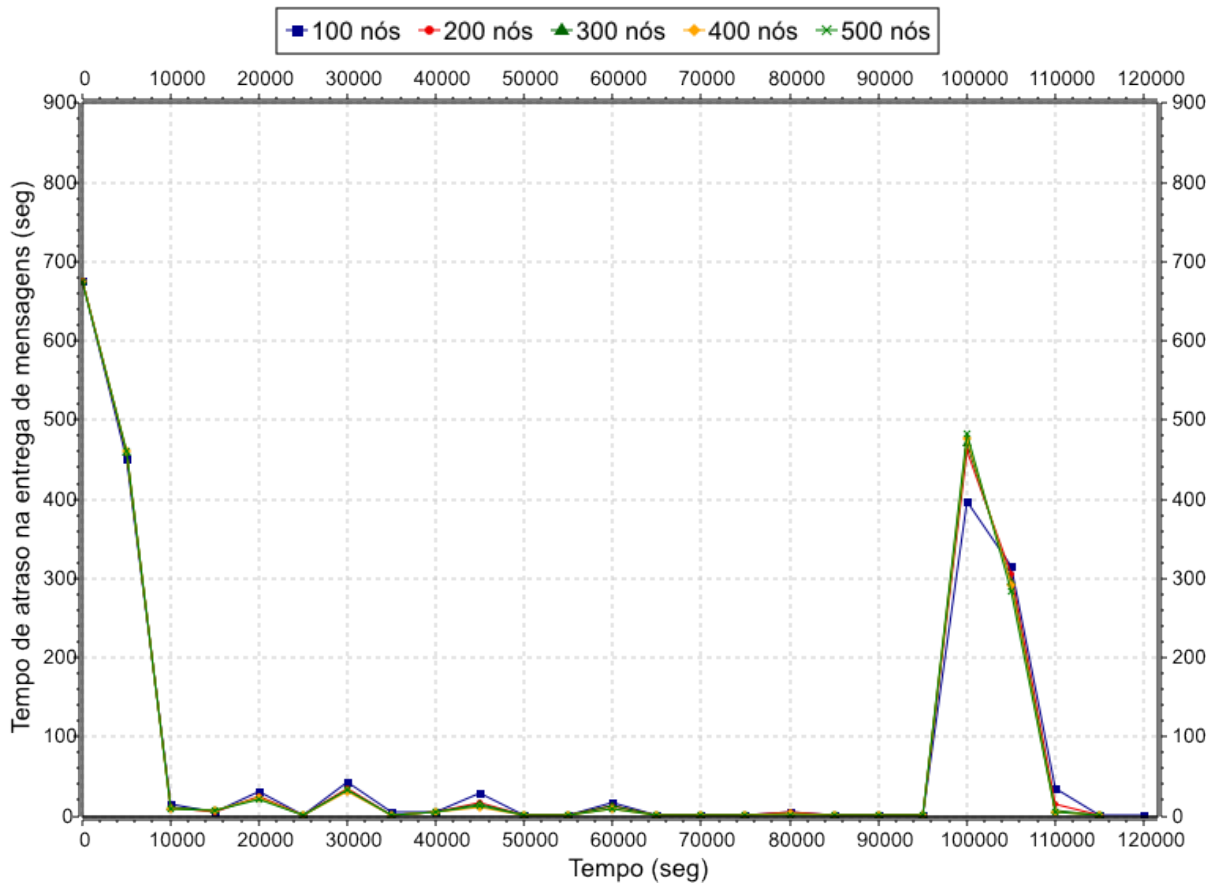
Porém os tempos de atraso observados no SP/WSN são aceitáveis na maior parte da simulação, ocorrendo um grande aumento no tempo apenas no momento final da simulação, onde a maior parte dos nós para de responder.

4.7.3 Variação da Energia Restante

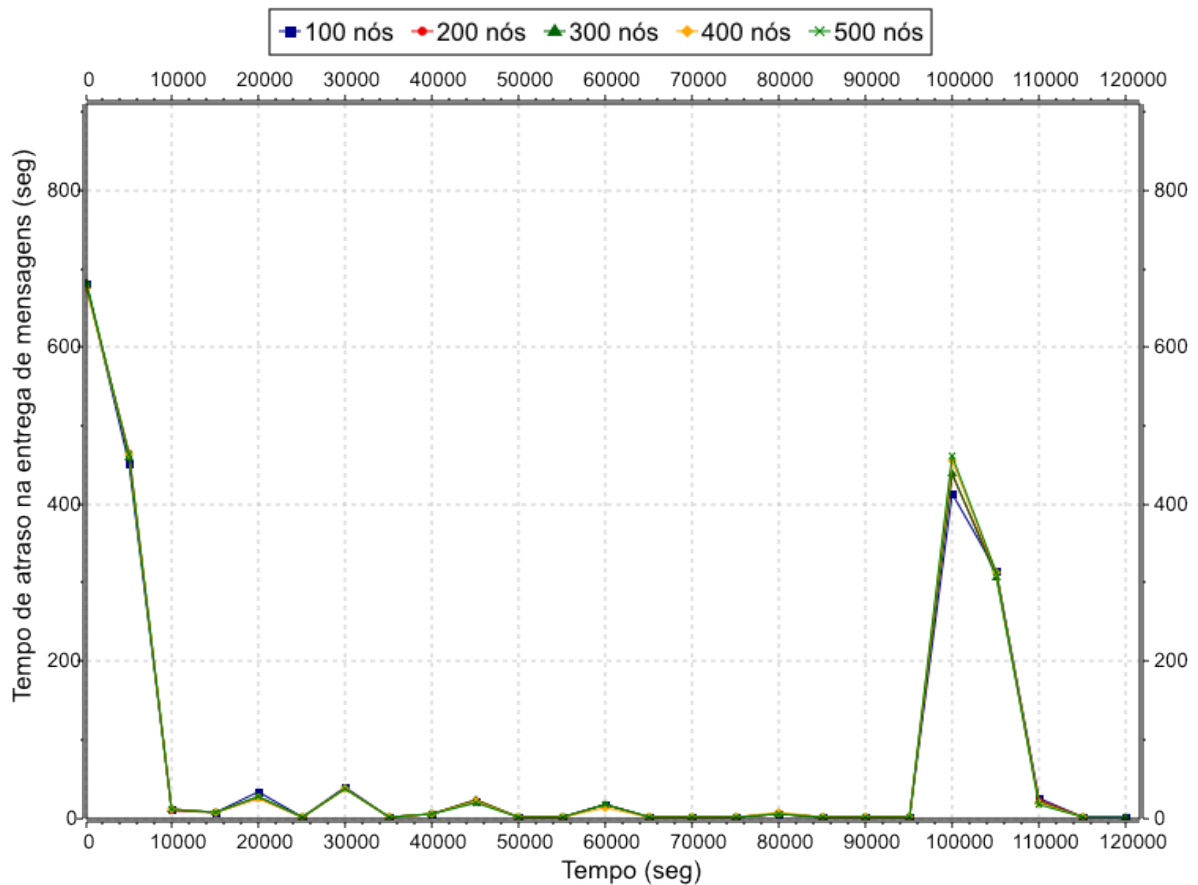
A variação da energia restante apresenta os valores: máximos, médios e mínimos de energia existente em todos os nós da rede em intervalos de tempo de 500 unidades. O objetivo deste indicador é demonstrar como o algoritmo consome a energia total da rede durante seu tempo total de funcionamento.

Para compreender este indicador, é importante relembrar os principais estados que um nó pode assumir durante uma simulação conforme a Figura 3.4. Foram identificados cinco estados distintos durante as simulações para os nós: **vivo**, **desconectado**, **espera**, **zumbi** e **ativo**. Todos os nós da rede no início da simulação estão no estado de **vivo**, e assim permanecem até que sua fonte de energia se esgote totalmente. Dentro do estado de **vivo** temos o estado **desconectado** que compreende os nós que ainda não possuem um pivô local, e os ativos que possuem o pivô. Dentro dos **desconectados** temos o estado em **espera** que são os nós com ou sem possibilidade de conexão ao *sink*, e os **zumbis** que compreende os nós que temporariamente perderam seu caminho para todos seus vizinhos diretos.

Este indicador é calculado da seguinte maneira: são considerados apenas os nós que durante todo o tempo de simulação, estiveram sob o estado **ativo** pelo menos uma vez, portanto são desconsiderados nós que estão isolados da rede; o simulador gera eventos



(a) Cenário I



(b) Cenário II

Figura 4.12: Atraso na entrega de mensagens - PDSR

internos com o percentual residual da bateria de cada um dos nós que compõe a rede, cada vez que um nó recebe este evento ele calcula o percentual de energia que foi consumido desde o último evento de atualização; esse percentual de energia consumida é acumulada de forma global para todos os nós e então dividida pelo número de nós total que passaram pelo estado de **ativo**; o resultado desta divisão é o percentual total acumulado de energia consumida que é subtraído do valor 100 para indicar como a energia é consumida durante a simulação.

A Figura 4.13 itens (a,b), Figura 4.14 itens (a,b) e Figura 4.15 do Cenário I, mostram que o comportamento do SP/WSN para o indicador de Variação da Energia Restante em cada uma das quantidades de nós simuladas. Observa-se que com o aumento da densidade, a variação da energia restante se aproxima em muito da média o que indica um consumo bem distribuído da energia contida em todos os nós **ativos**. Mesmos os itens (a) e (b), onde ocorreu um desvio da variação, pode-se observar que o desvio foi mínimo, o que permite concluir que o consumo de energia ocorre de forma balanceada.

Já a Figura 4.16 itens (a,b), Figura 4.17 itens (a,b) e a Figura 4.18 do Cenário II, ilustra que a utilização de uma densidade fixa de nós, cria pequenas oscilações entre os valores máximos e mínimos devido a melhor distribuição dos nós e o aumento do número de saltos. Mas da mesma forma que os resultados do Cenário I, o desvio obtido é muito pequeno, e os valores máximos e mínimos praticamente acompanham os valores da média.

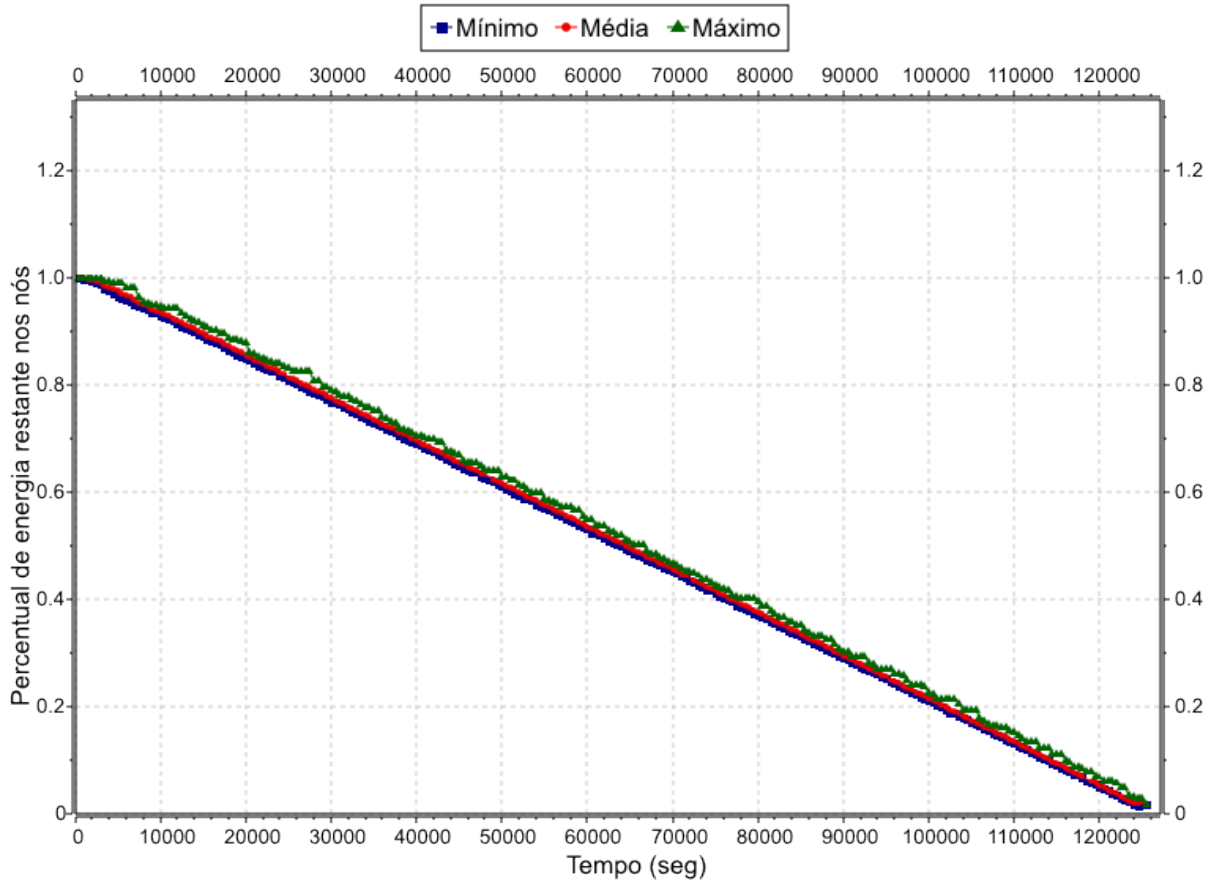
No algoritmo PDSR, conforme mostra as Figuras 4.19, 4.20 e 4.21 no Cenário I o consumo ocorreu de forma balanceada, sem grandes variações dos valores máximos e mínimos, o que indica que o SP/WSN possui um desempenho similar comparado ao PDSR.

O mesmo resultado pode ser observado para o PDSR, de acordo com as Figuras 4.22, 4.23 e 4.24 no Cenário II. A densidade fixa e a melhor distribuição dos nós, criaram pequenas oscilações dos valores máximos e mínimos, porém eles acompanham o valor da média.

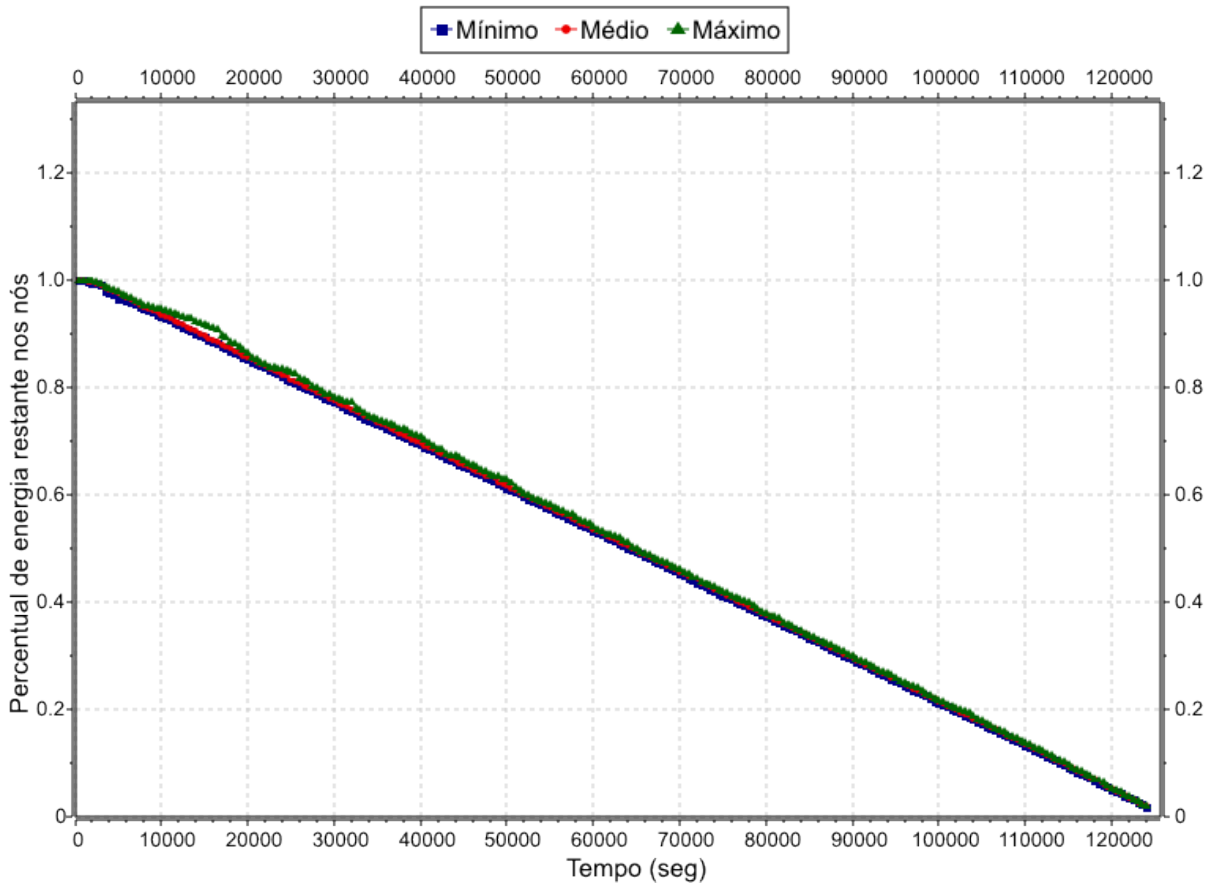
É possível concluir que o SP/WSN se iguala ao PDSR alcançando o objetivo de realizar um consumo equilibrado de todas as fontes de energia dos nós da rede, reduzindo a ocorrência de *energy sink-hole problem*.

4.7.4 Número de Nós Ativos

Este indicador apresenta o número médio de nós ativos que é definido pelo conjunto de nós que possuem energia e conhecem um caminho até o nó *sink*, totalizando o valor

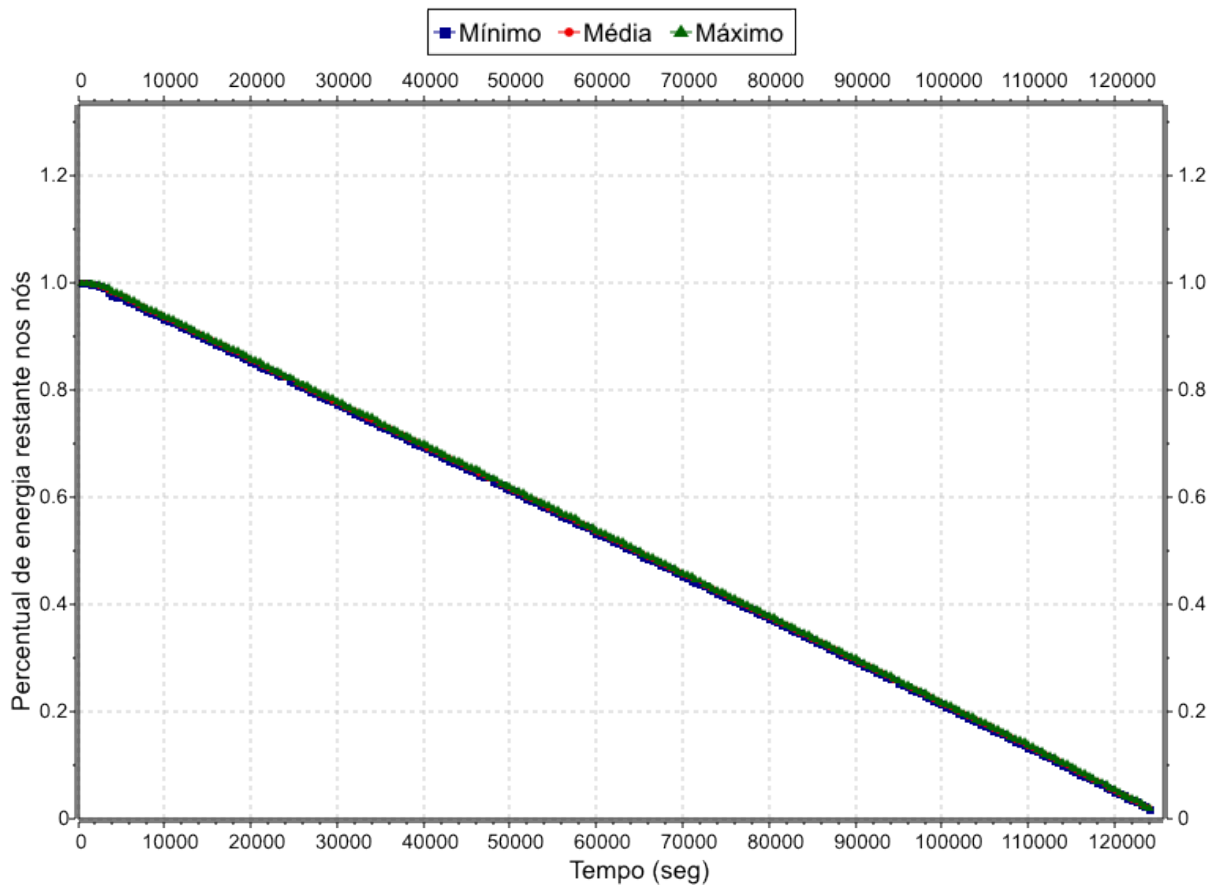


(a) 100 nós

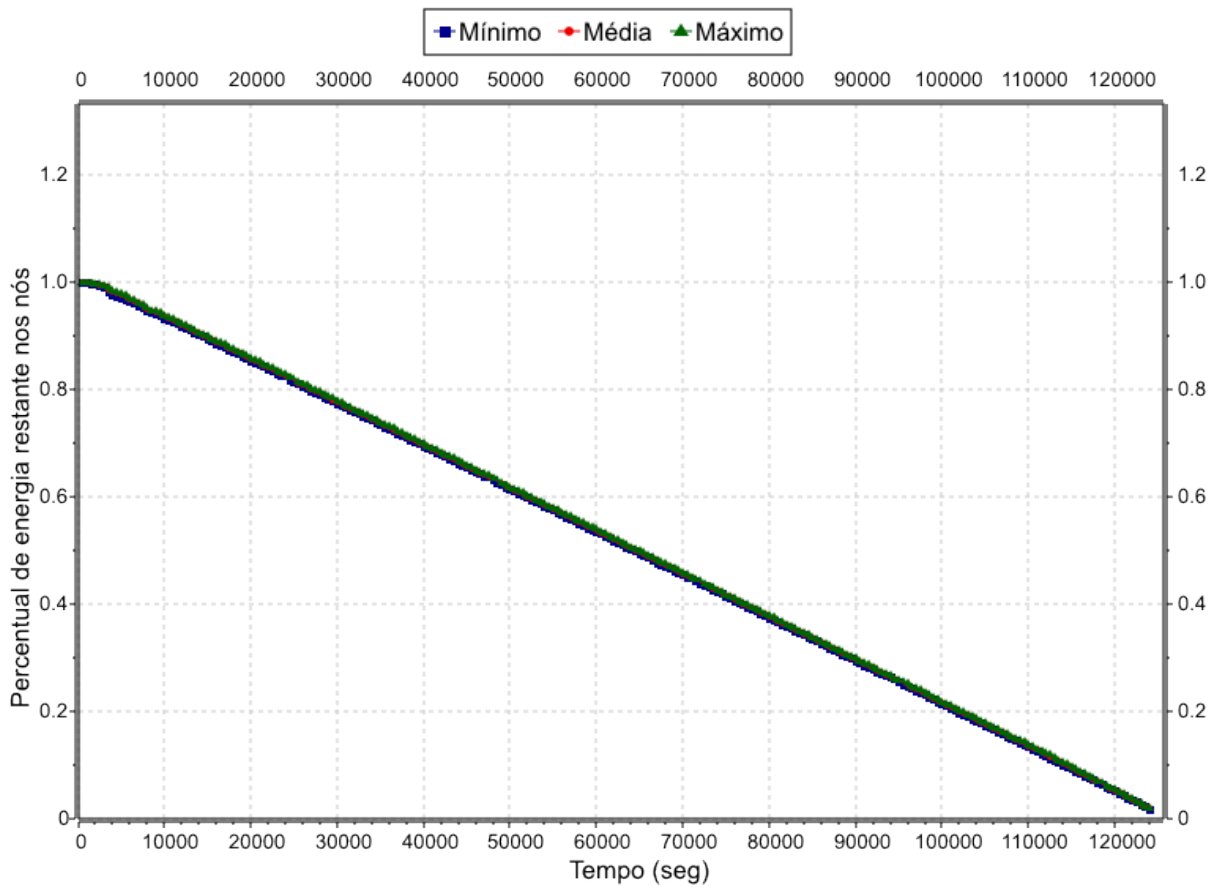


(b) 200 nós

Figura 4.13: Variação do percentual de energia restante no cenário I - SP/WSN

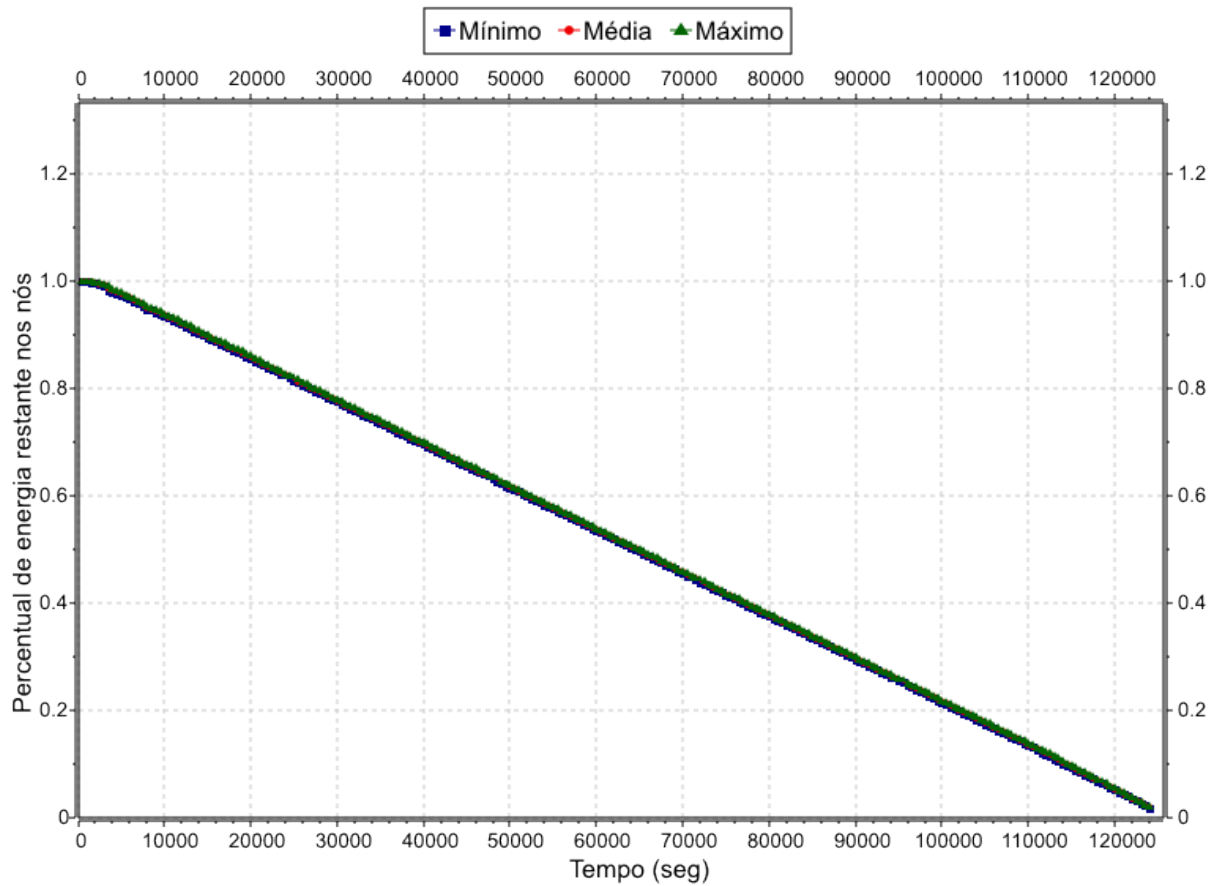


(a) 300 nós



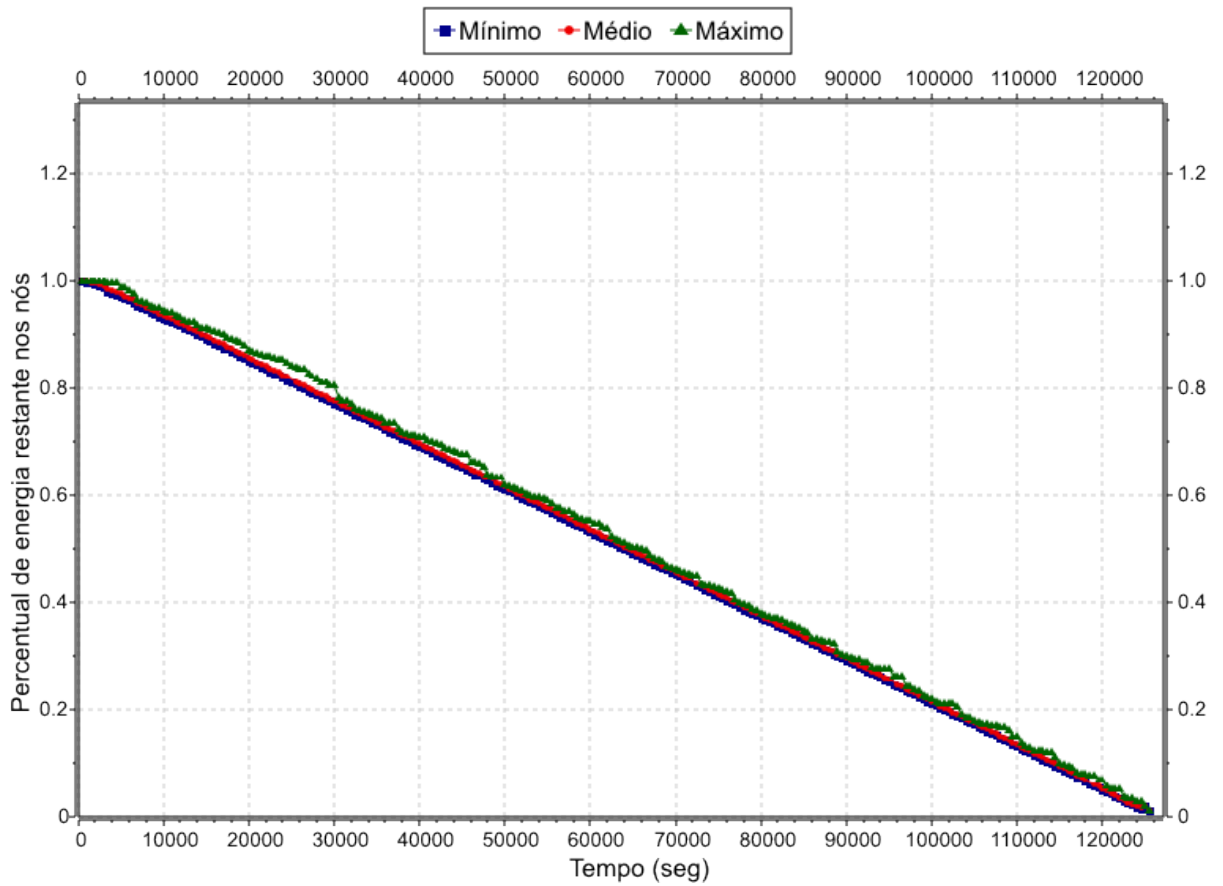
(b) 400 nós

Figura 4.14: Variação do percentual de energia restante no cenário I - SP/WSN

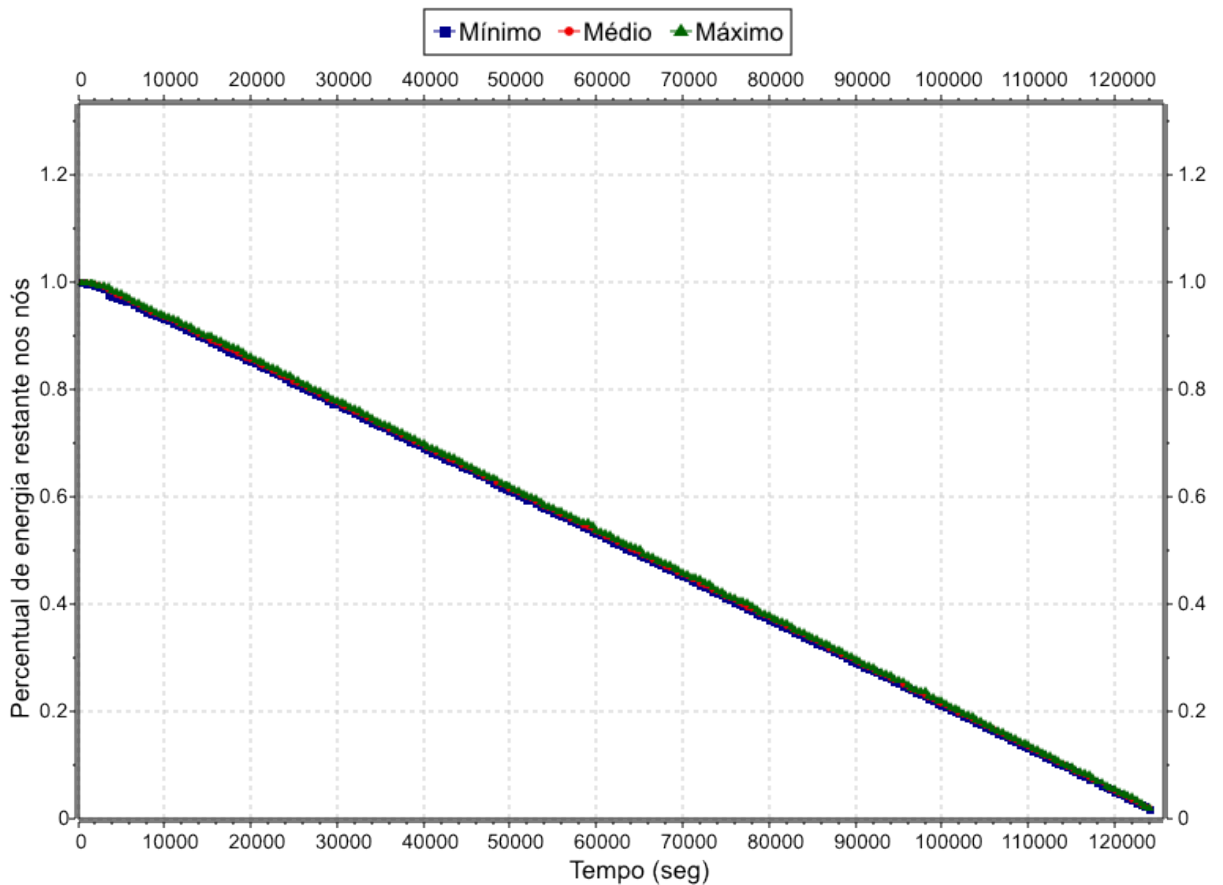


(a) 500 nós

Figura 4.15: Variação do percentual de energia restante no cenário I - SP/WSN

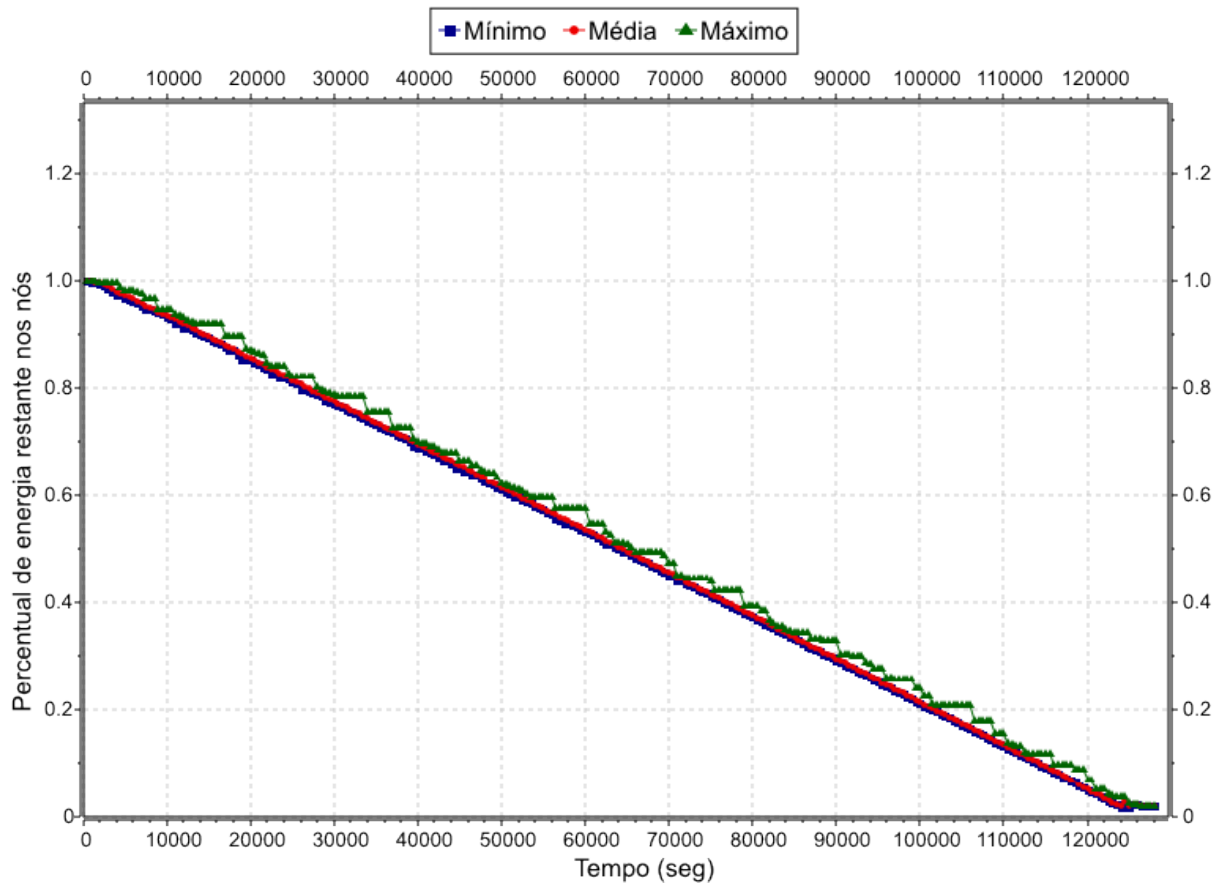


(a) 100 nós

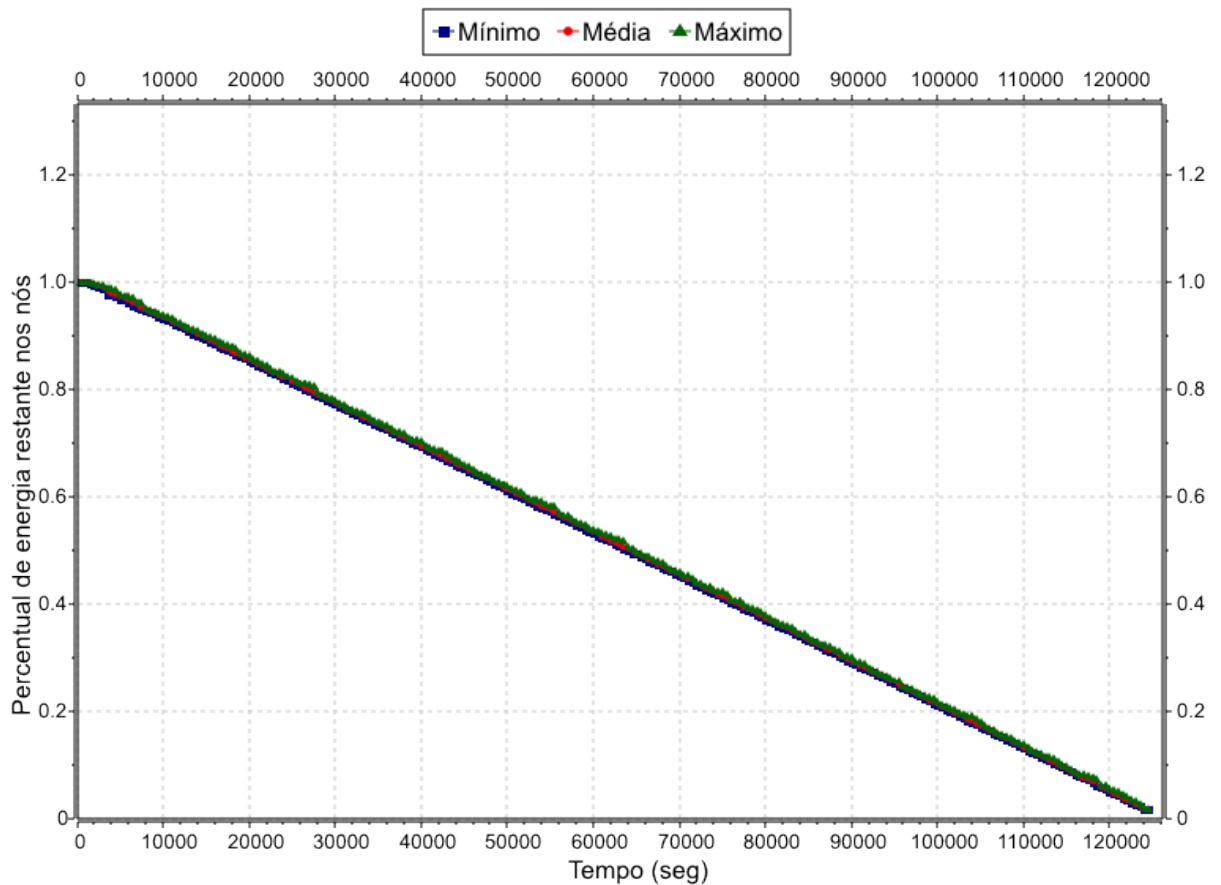


(b) 200 nós

Figura 4.16: Variação do percentual de energia restante no cenário II - SP/WSN

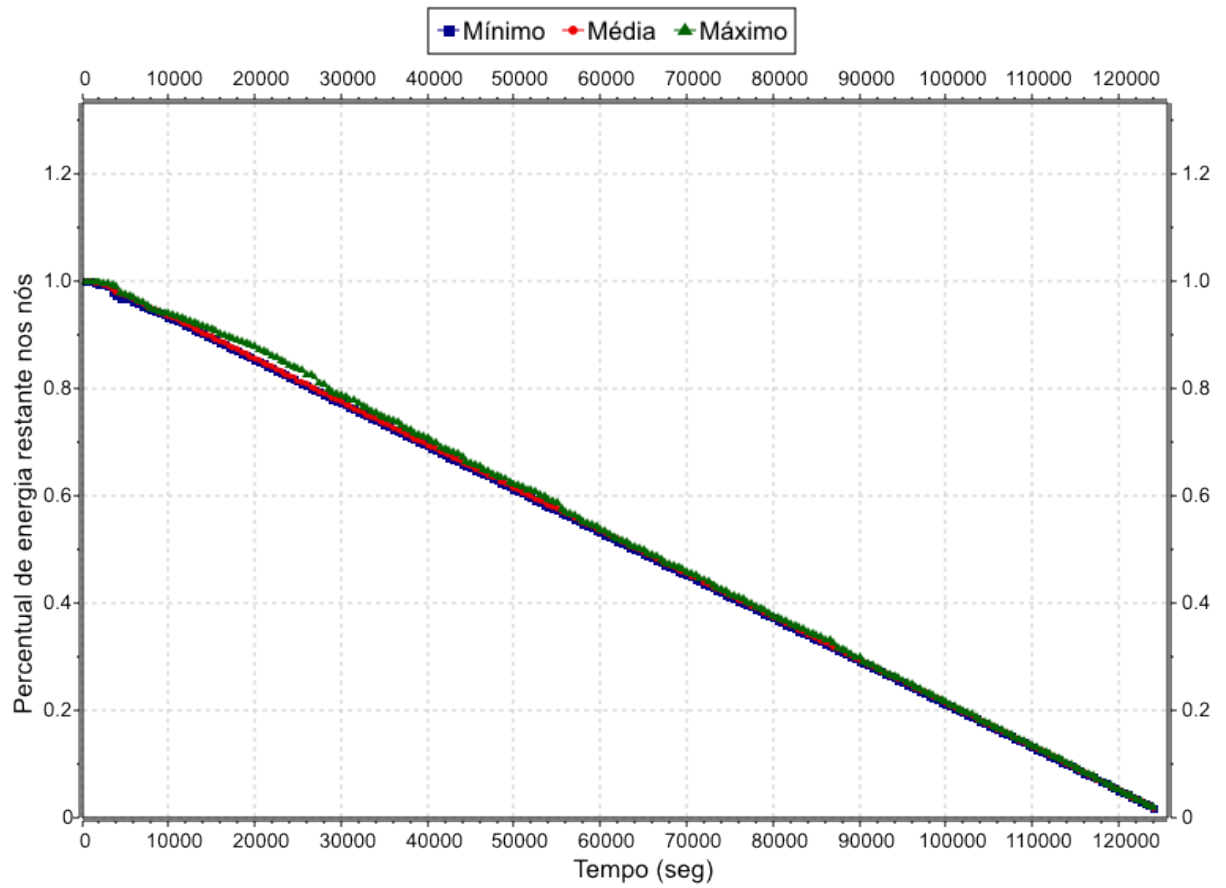


(a) 300 nós



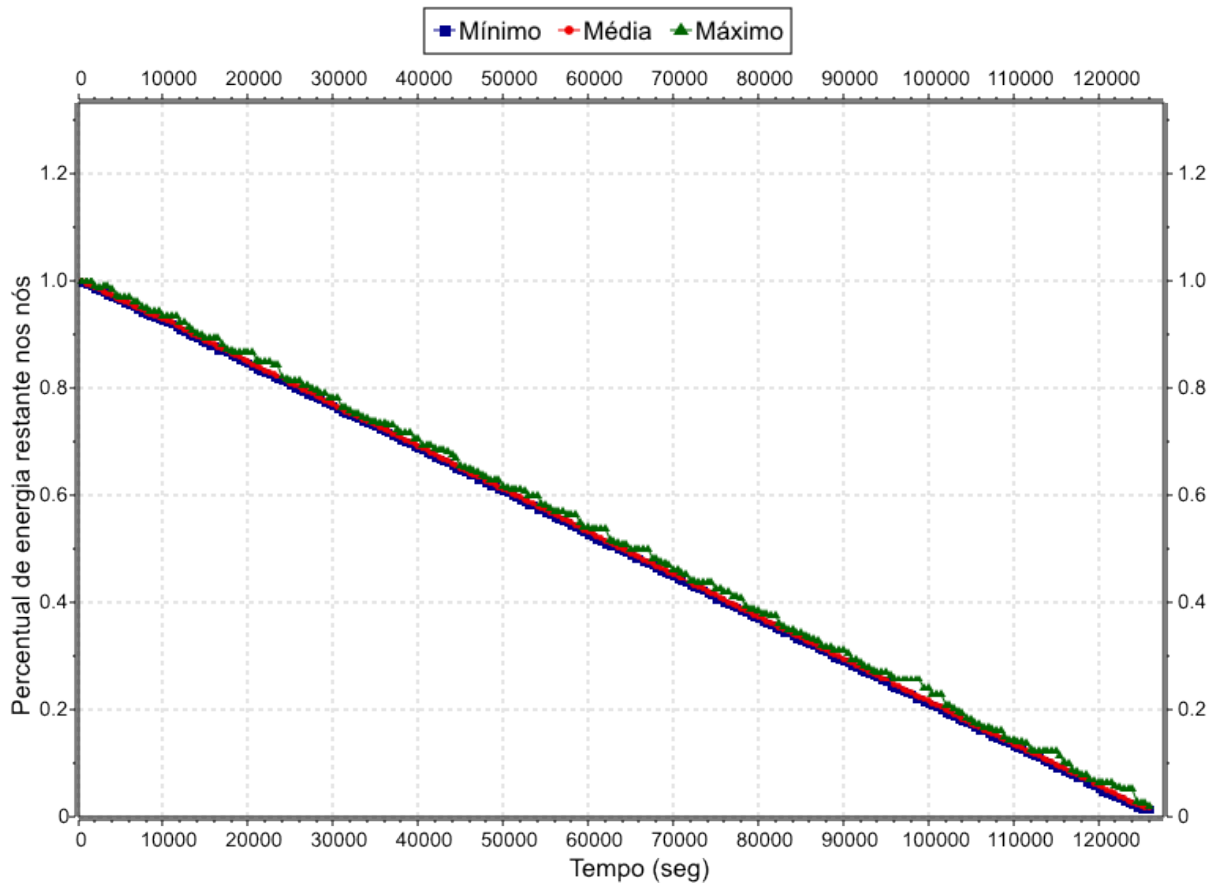
(b) 400 nós

Figura 4.17: Variação do percentual de energia restante no cenário II - SP/WSN

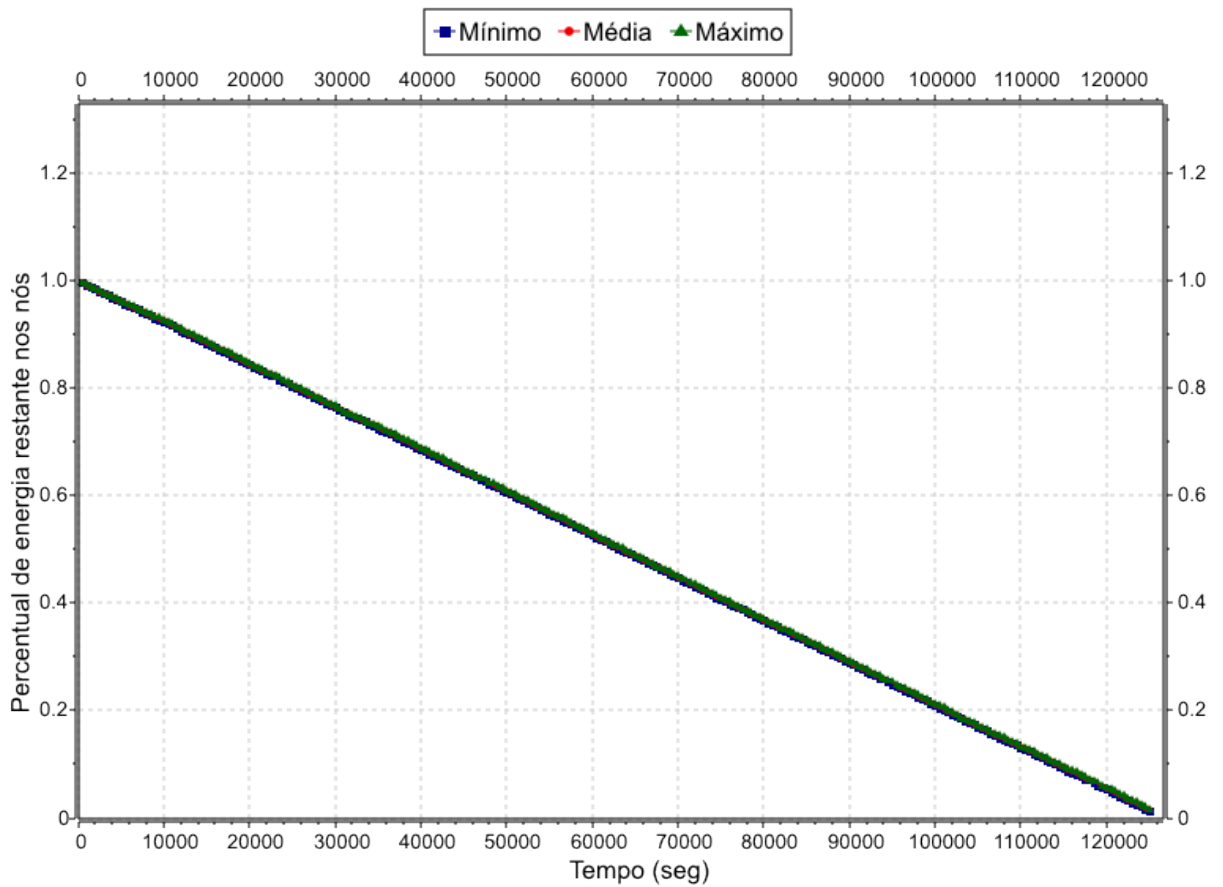


(a) 500 nós

Figura 4.18: Variação do percentual de energia Restante no cenário II - SP/WSN

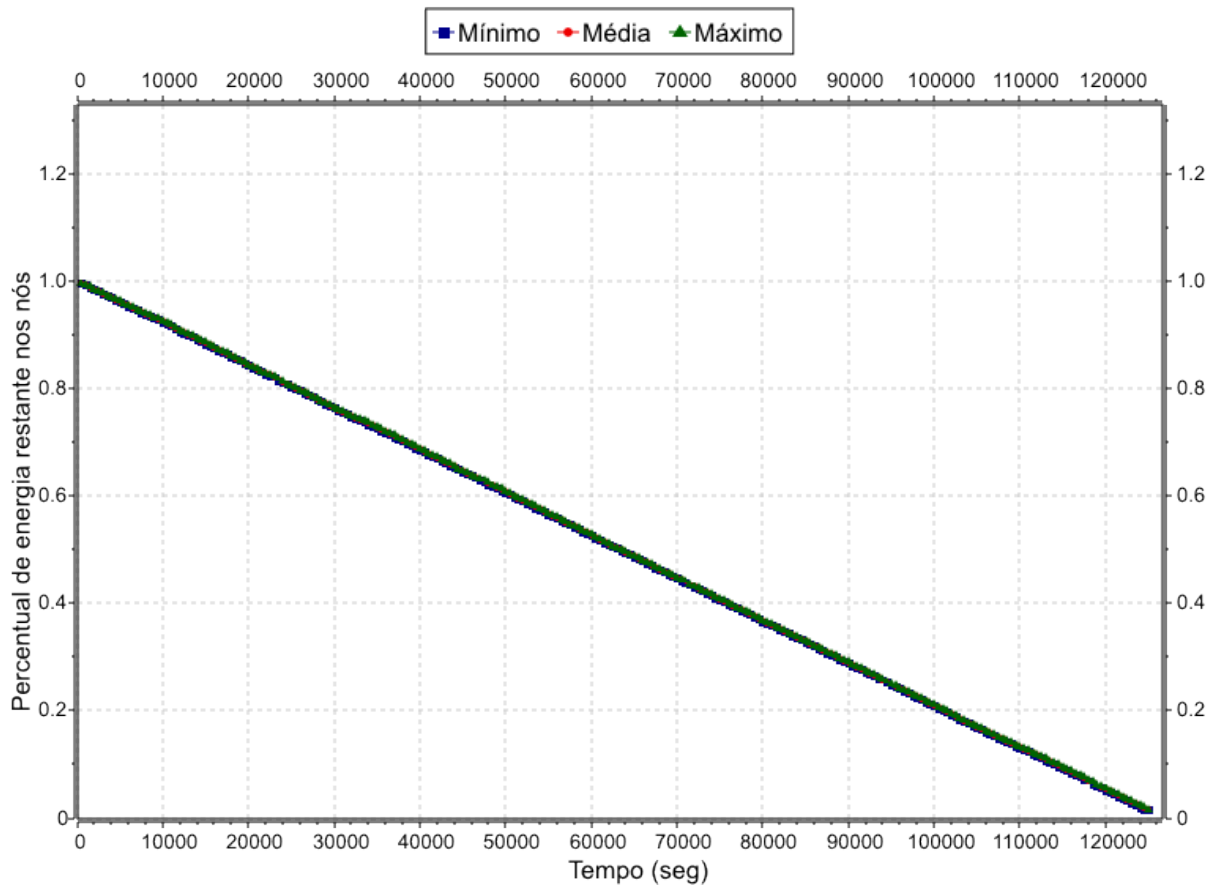


(a) 100 nós

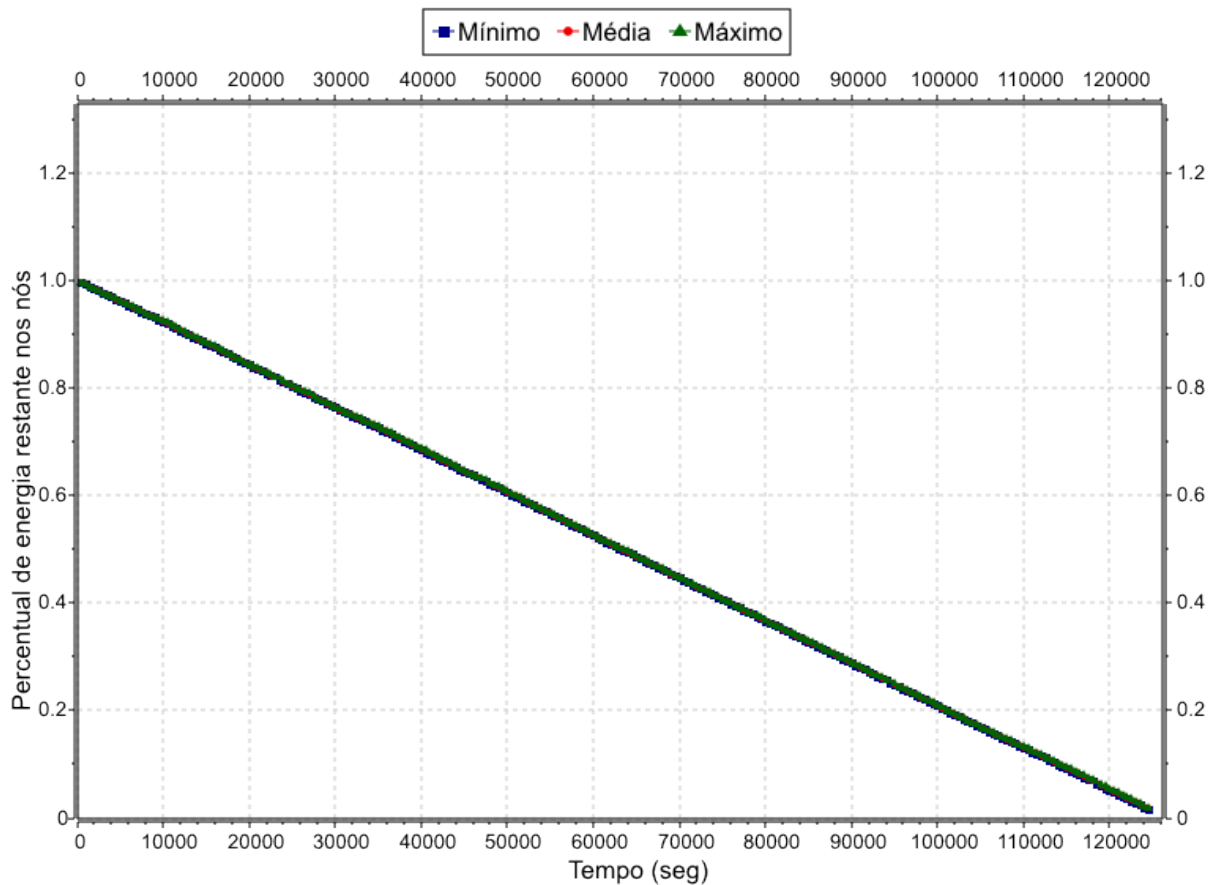


(b) 200 nós

Figura 4.19: Variação do percentual de energia restante no cenário I - PDSR

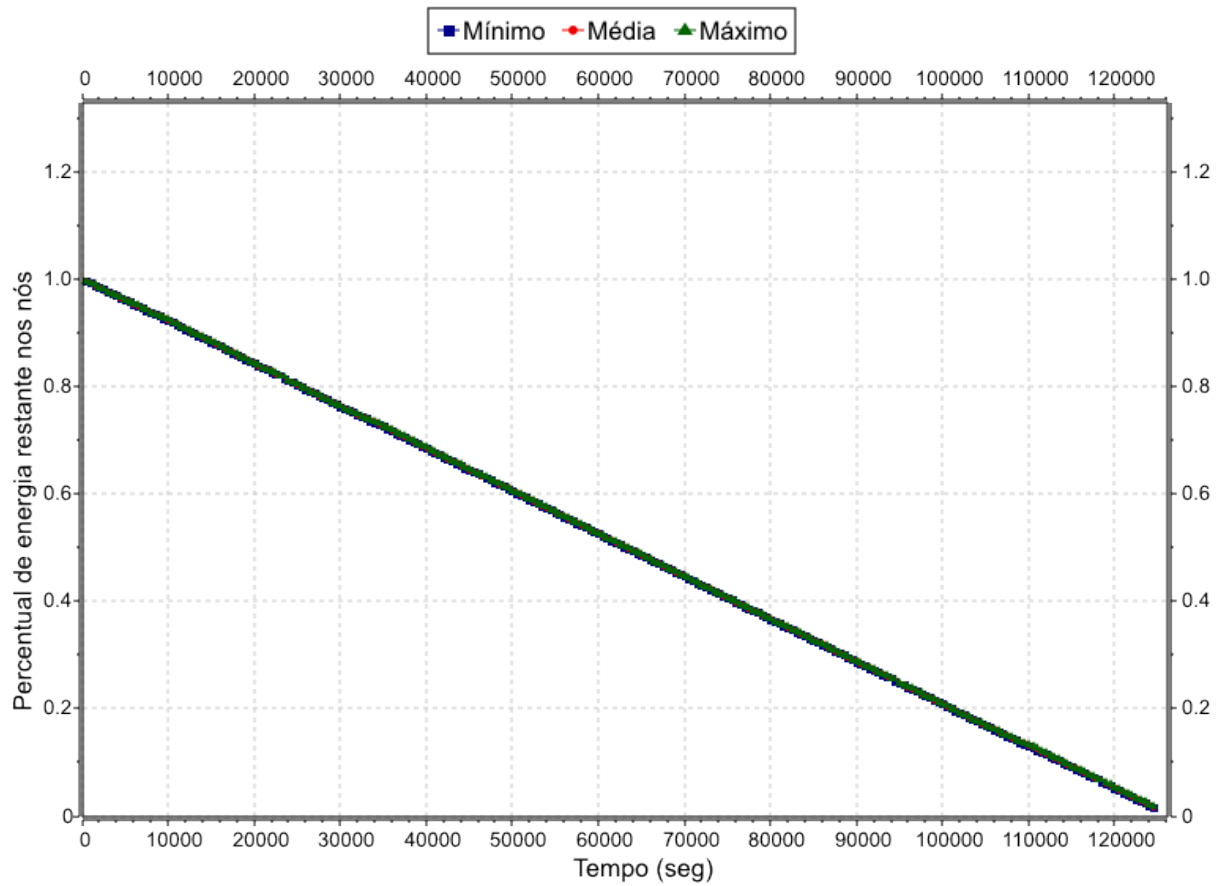


(a) 300 nós



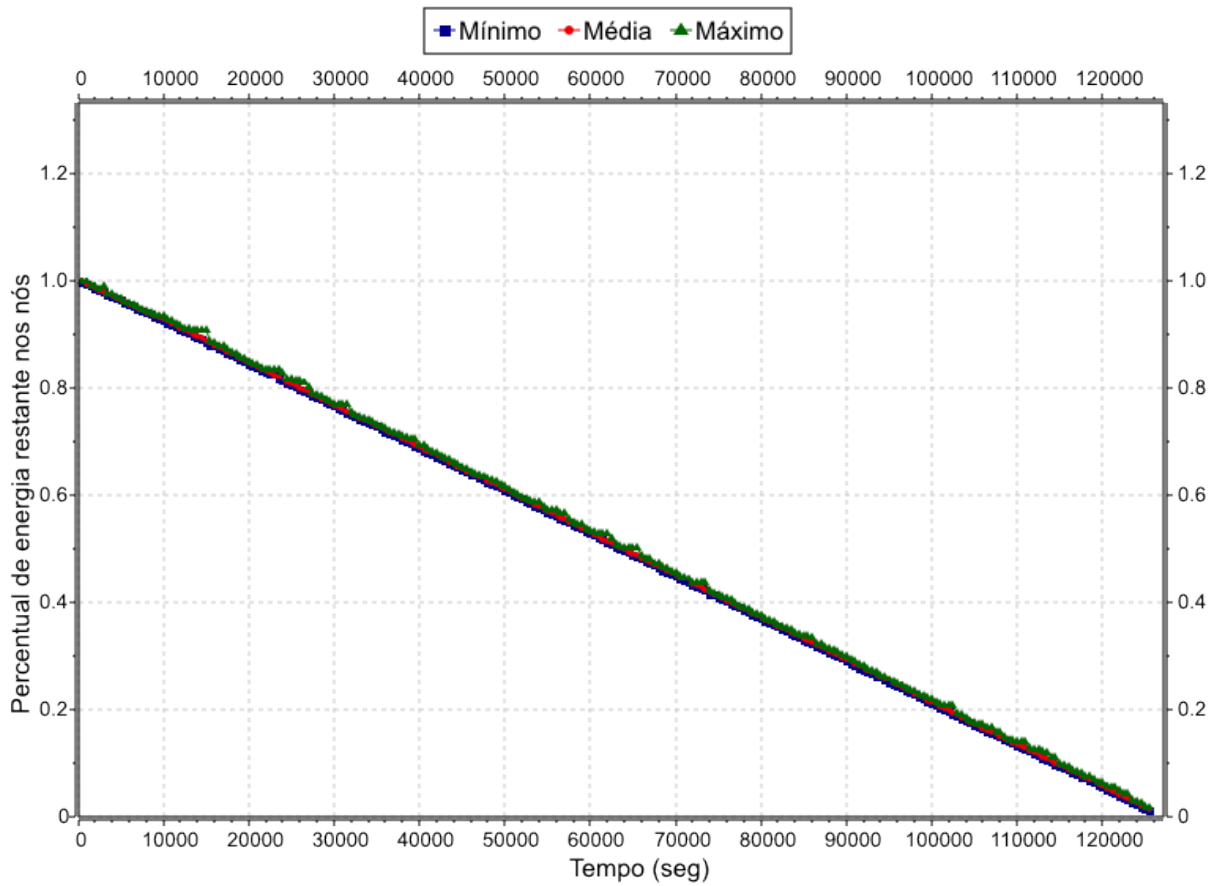
(b) 400 nós

Figura 4.20: Variação do percentual de energia restante no cenário I - PDSR

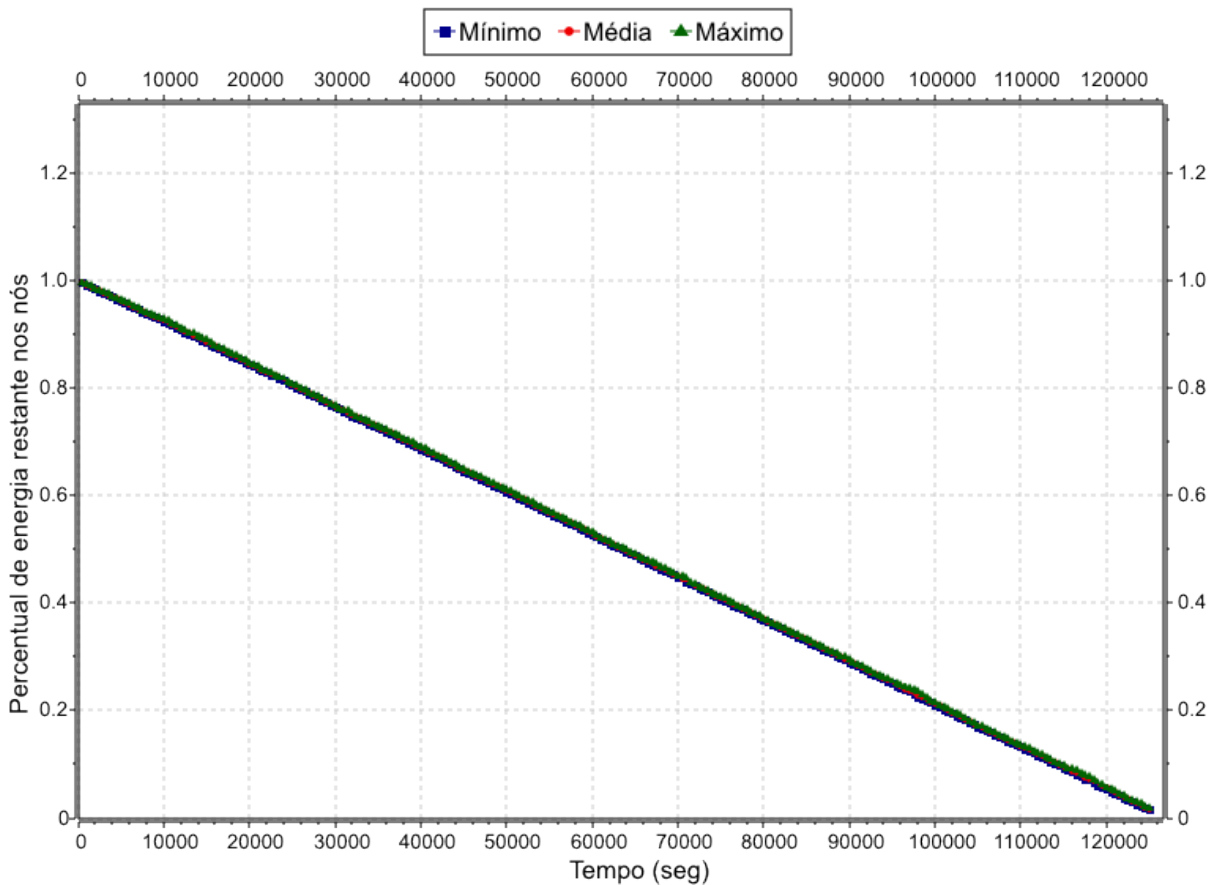


(a) 500 nós

Figura 4.21: Variação do percentual de energia restante no cenário I - PDSR

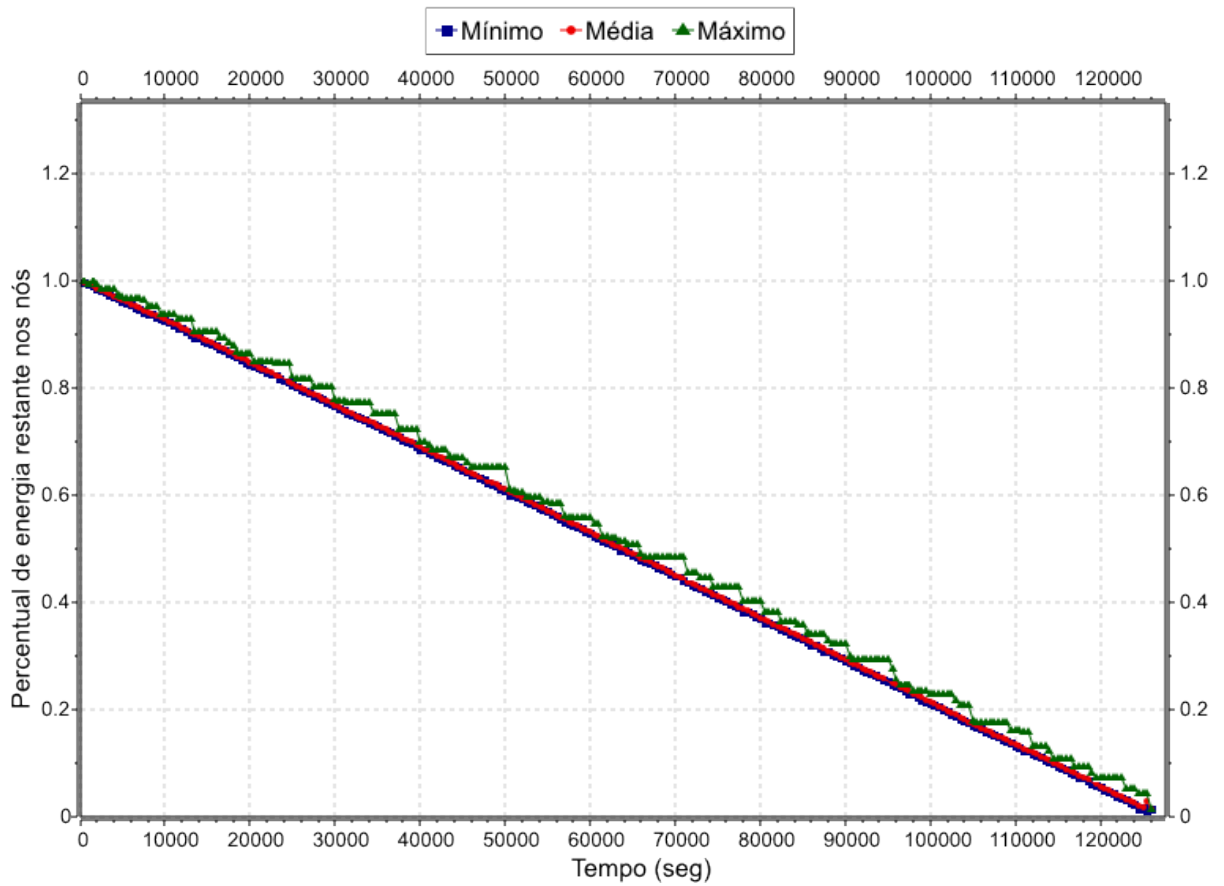


(a) 100 nós

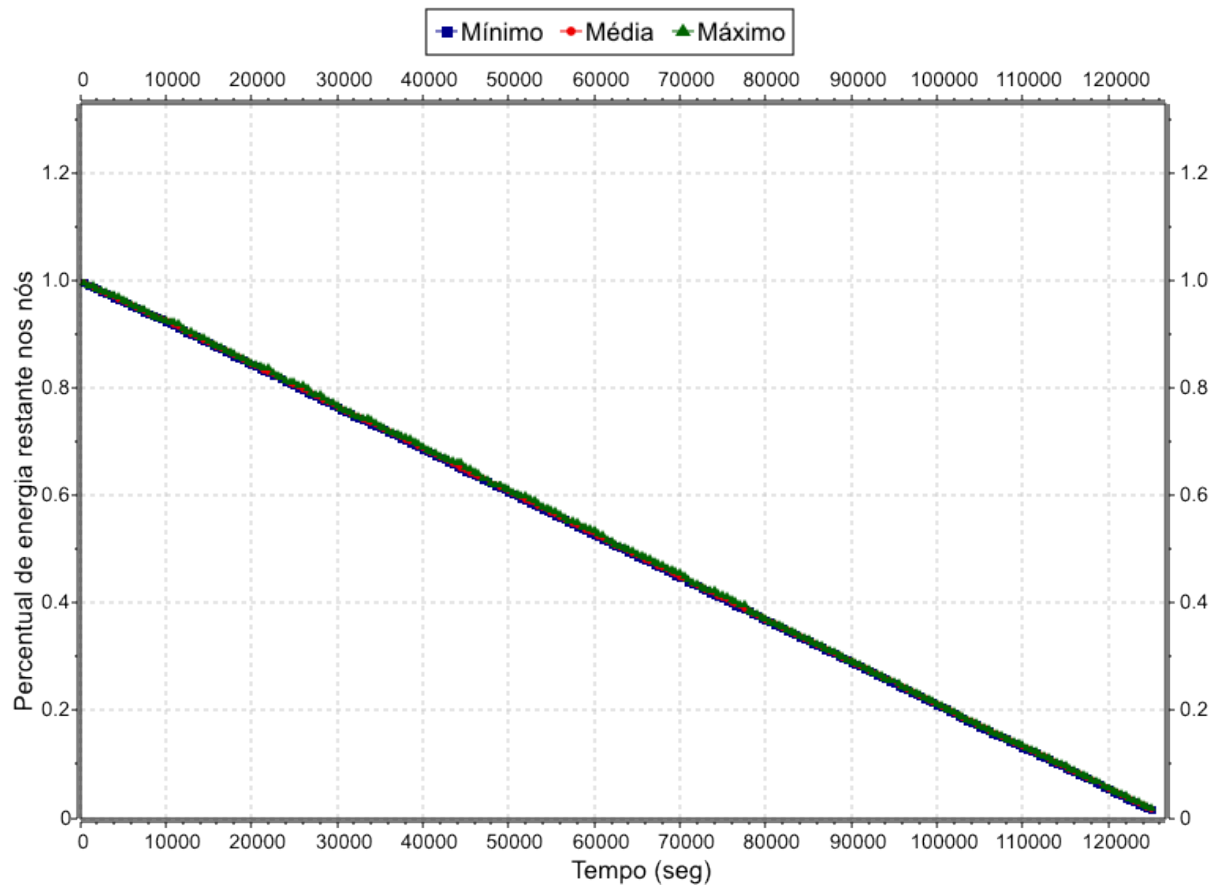


(b) 200 nós

Figura 4.22: Variação do percentual de energia restante no cenário II - PDSR

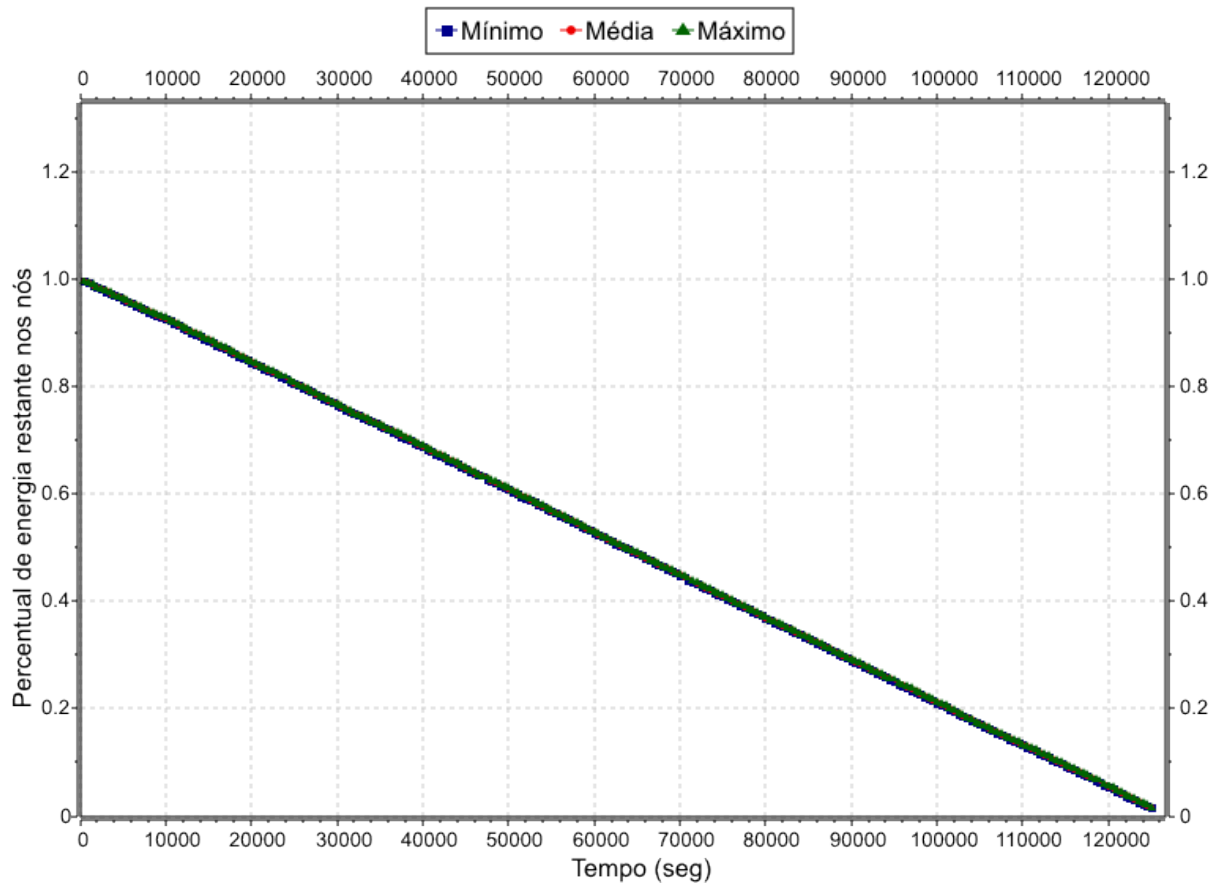


(a) 300 nós



(b) 400 nós

Figura 4.23: Variação do percentual de energia restante no cenário II - PDSR



(a) 500 nós

Figura 4.24: Variação do percentual de energia restante no cenário II - PDSR

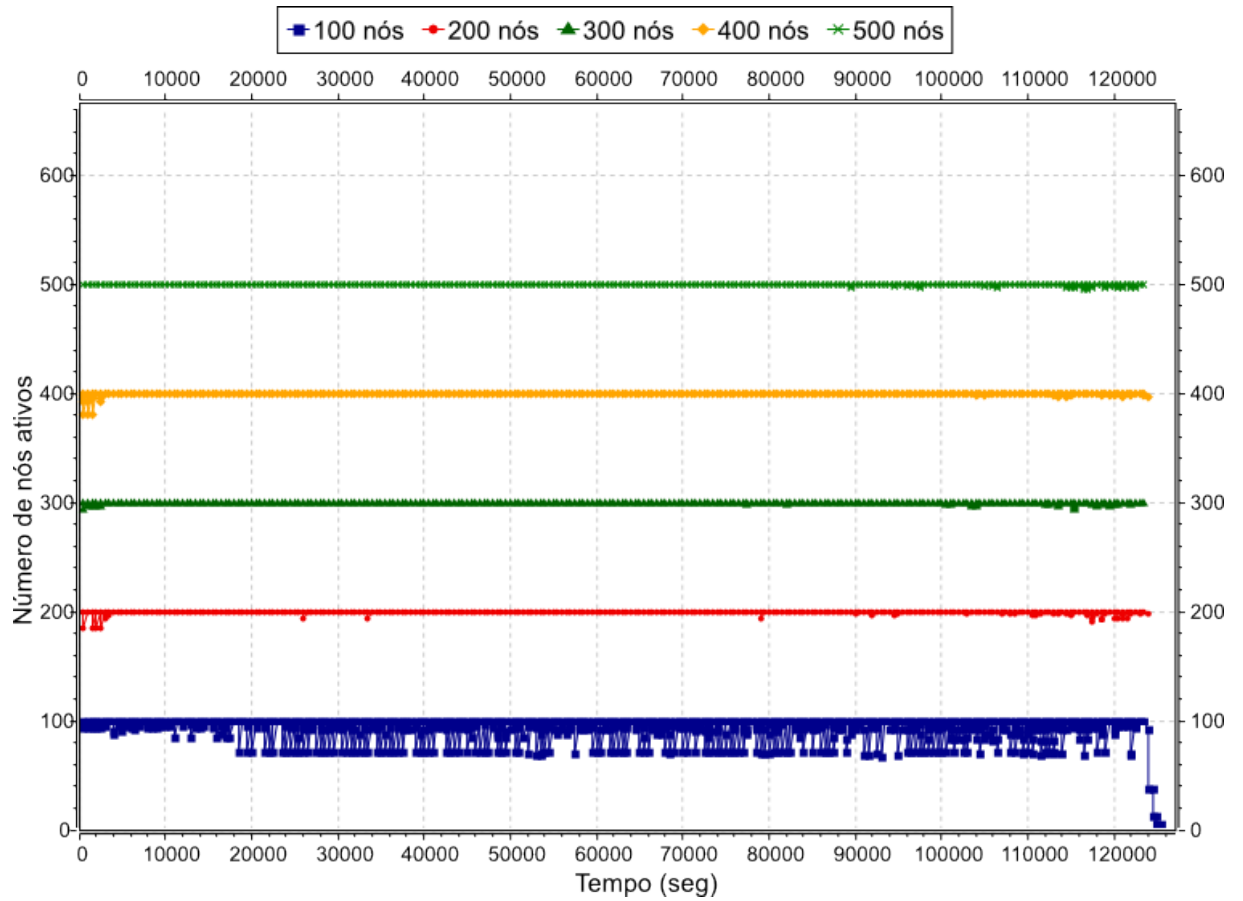
dentro de unidades de tempo com o valor de 500 segundos. Conforme mostra a Figura 4.25, itens (a) e (b), o SP/WSN proporciona uma cobertura ampla da área monitorada, visto que na maioria do tempo os nós permanecem ativos, sem sofrer influência da variação de densidade da rede.

Já o algoritmo PDSR não demonstrou o mesmo resultado como ilustra a figura 4.26, em ambos os cenários o número de nós ativos é menor. Para melhor entender esse resultado é importante compreender que nós ativos para o PDSR consiste em um nó que possui uma rota completa ao *sink* após um processo de descoberta ser realizado, o que está diretamente ligado a característica reativa do algoritmo. Já para o SP/WSN, um nó ativo é aquele que possui um pivô local, ou seja, um nó vizinho informou que possui um caminho para o *sink*, caracterizando um algoritmo pró-ativo. Em ambos os Cenários I e II, observa-se um rápido crescimento no número de nós ativos nos primeiros instantes de tempo da simulação. Posteriormente com a queda do nível de energia nos nós, ocorrem oscilações devido a destruição das rotas e a execução do processo de redescoberta de novas rotas.

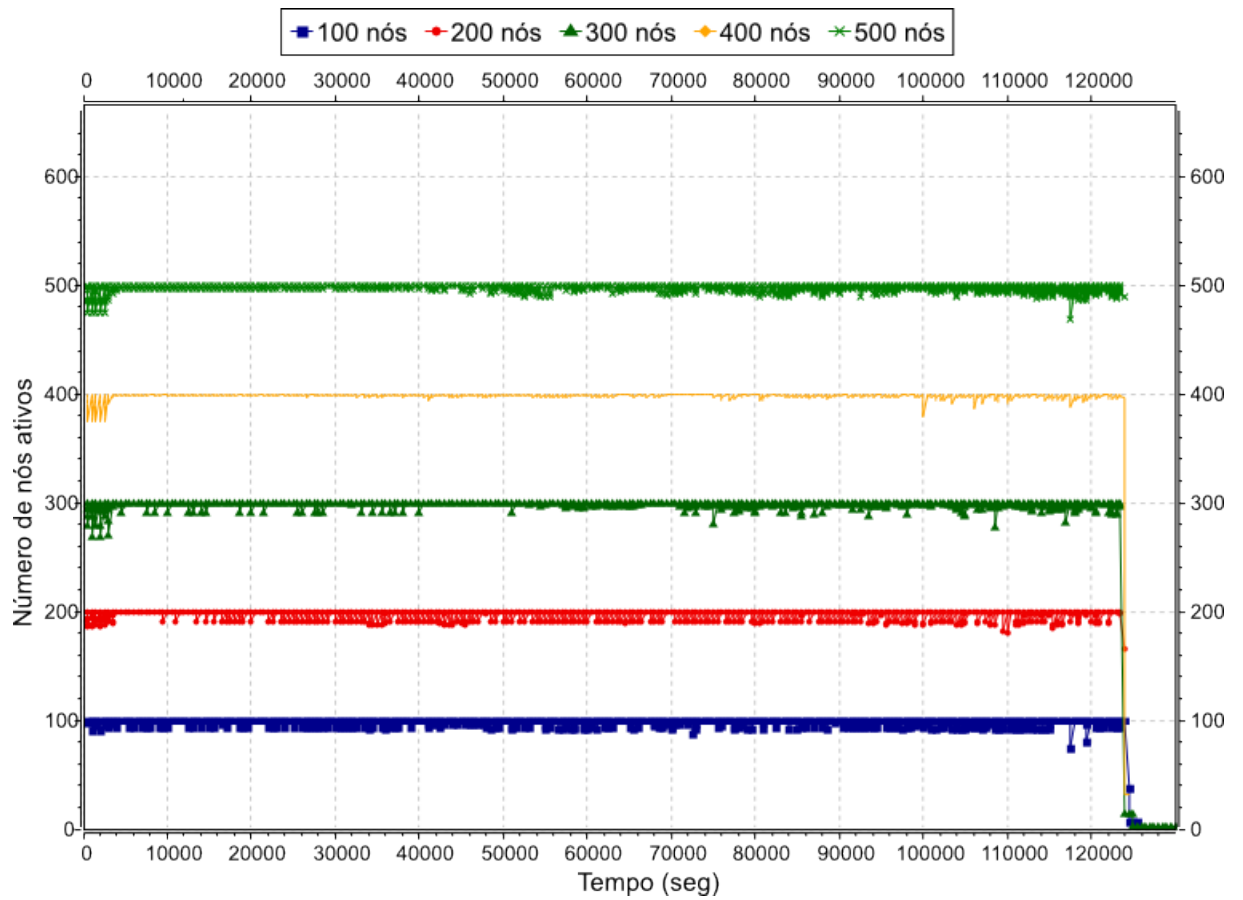
Com estes resultados, pode-se observar que a cobertura da rede oferecida pelo SP/WSN é superior ao PDSR possibilitando um funcionamento mais eficaz.

4.8 Considerações Finais

Este capítulo apresentou detalhes sobre a implementação dos algoritmos SP/WSN e PDSR utilizando o simulador OMNet++ e o plugin MiXiM. Importante destacar que a adoção desse ambiente de simulação traz confiabilidade para a validação dos algoritmos. Em seguida, foram apresentados os resultados obtidos com as simulações, realizando uma análise comparativa entre os indicadores dos dois algoritmos. O SP/WSN proveu uma maior cobertura, em comparação ao PDSR, devido propiciar uma maior taxa de entrega mesmo com um número de tentativas reenvio da mensagem limitado, e além de alcançar uma maior média de nós ativos durante o tempo de vida da rede. O próximo capítulo apresenta as conclusões obtidas e os possíveis trabalhos futuros para esta pesquisa.



(a) Cenário I



(b) Cenário II

Figura 4.25: Número de nós ativos - SP/WSN

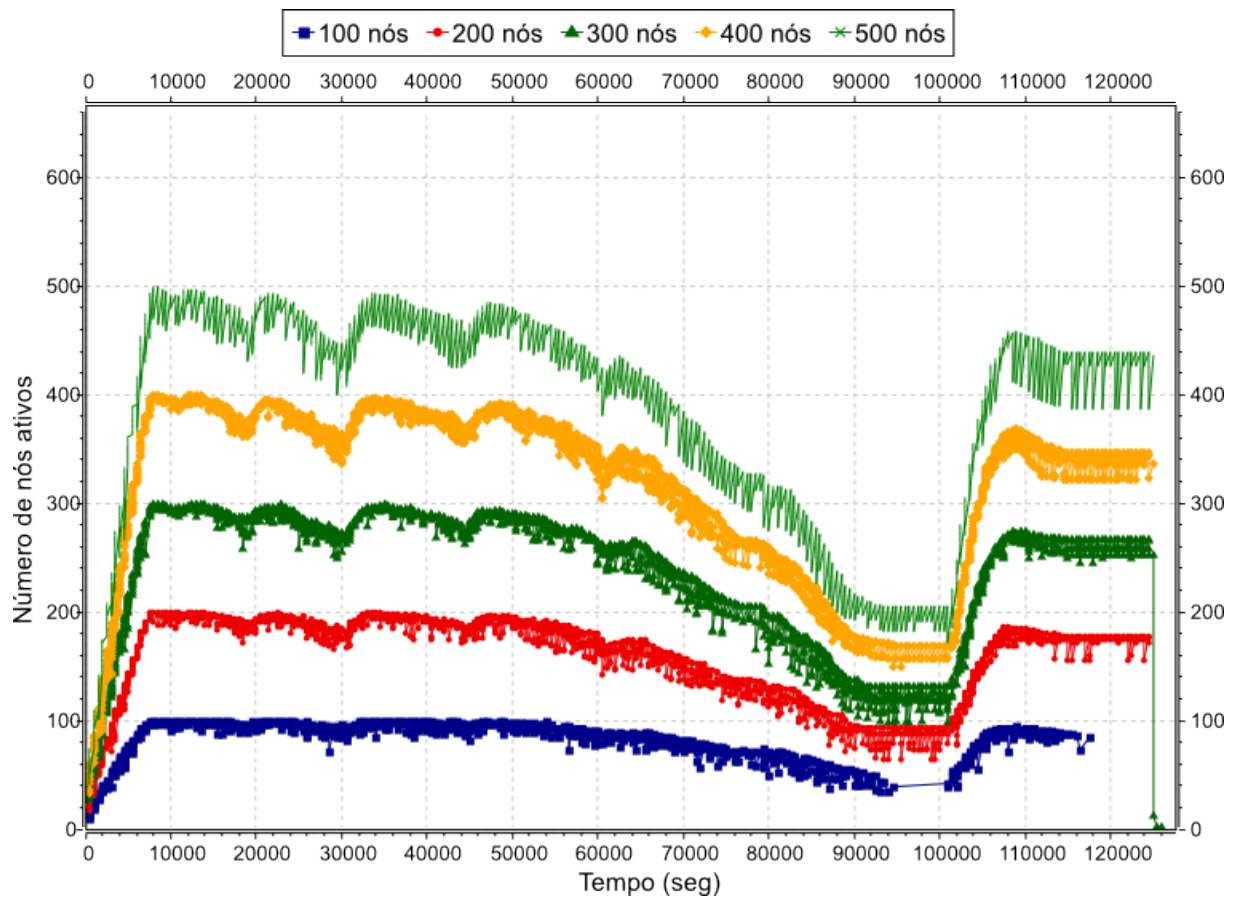
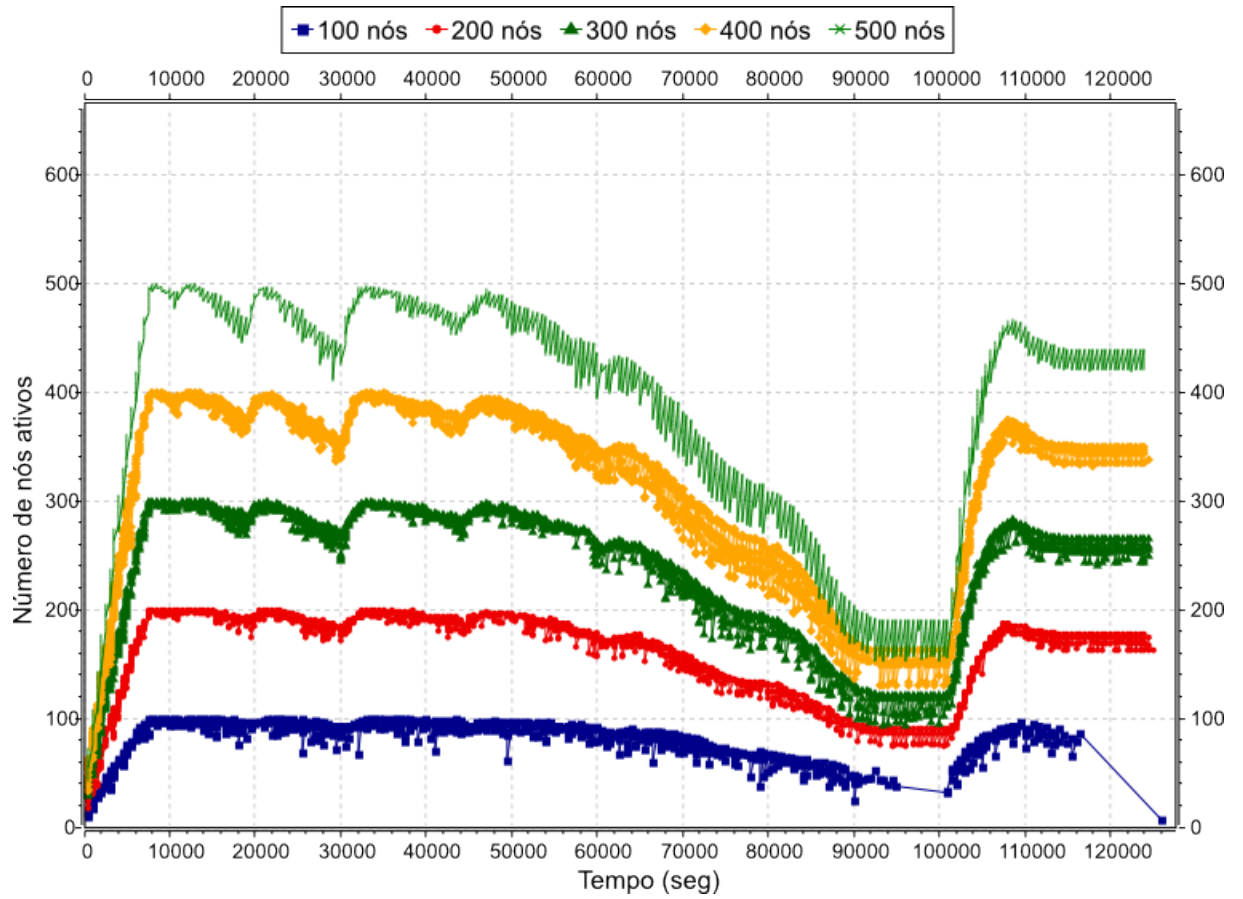


Figura 4.26: Número de nós ativos - PDSR

Capítulo 5

Conclusões

As redes de sensores sem fio constituem uma área que possibilita diversas aplicações práticas e com questões a serem pesquisadas e melhoradas. Desde a década de 80, diversas pesquisas realizadas tornaram possível a aplicação de WSN para inúmeras funções como: segurança, monitoramento ou ainda fins militares. Com o avanço dessas pesquisas e principalmente a redução dos custos dos componentes eletrônicos que compõem os nós sensores, cada vez mais as WSN serão utilizadas.

Neste trabalho foi proposto um novo algoritmo de roteamento para redes de sensores sem fio chamado SP/WSN, com o objetivo de manter o balanceamento do nível de energia entre todos os nós, tal que a área e o tempo de cobertura da rede sejam maximizados. O SP/WSN reúne vários princípios importantes para que um algoritmo possua características de eficiência energética, sem utilizar-se de técnicas de desligamento parcial do nó ou controle da potência de transmissão. Fato este que possibilita ao SP/WSN uma maior flexibilidade de aplicação em diversos tipos de redes de sensores diferentes. Seu reduzido número de mensagens de controle e a simplicidade no processo de seleção das rotas, auxilia não apenas na eficiência energética mas facilita sua implementação em qualquer tipo de dispositivo sensor, por não necessitar de grande capacidade computacional para a execução.

Para a validação do algoritmo SP/WSN optou-se pela implementação utilizando o simulador OMNet++ com o *pluggin* MiXiM, que possui diversas vantagens na sua utilização como: simplicidade através de módulos que simulam componentes de um nó sensor, possibilidade de customização dos módulos e grande número de parâmetros que podem ser facilmente configurados através de arquivos de inicialização. O OMNet++ também possui ferramentas que possibilitam a captura de dados sobre a execução da simulação e a compilação desses dados para a geração de gráficos dentro da IDE de desenvolvimento.

Por fim, o simulador permite avaliar os algoritmos dentro de um ambiente o mais próximo de uma situação real, o que dá maior veracidade aos resultados obtidos.

Os resultados dos indicadores obtidos confirmam que o SP/WSN é uma solução melhor que o algoritmo PDSR em relação a taxa de entrega de mensagens e o número de nós ativos, e possui um resultado similar para o indicador atraso na entrega de mensagens e variação do percentual de energia restante. A taxa de entrega obteve valores próximo a 100% em praticamente todo o tempo de simulação e o número de nós ativos demonstra que a maioria dos nós permanecem em funcionamento por todo o tempo. O atraso de entrega das mensagens demonstrou que o SP/WSN permanece pela maior parte da simulação com um tempo de atraso muito próximo a zero, ocorrendo o aumento apenas no final da simulação onde vários nós da rede não estão mais funcionando. E o resultado da variação do percentual de energia restante, nos permite concluir que o SP/WSN e o PDSR realizam um consumo balanceado das fontes de energia, conforme a premissa de um algoritmo com o objetivo de realizar a utilização eficiente de energia.

Como proposta de trabalhos futuros, temos: avaliar o *overhead* causado pelo SP/WSN; aplicar modelos de movimentação dos nós, existente no plugin MiXiM, para analisar o impacto da alteração da topologia da rede; realizar a comparação do algoritmo SP/WSN com outras soluções clássicas e também como as criadas por (GHADA; LI; JI, 2012) e (CARPES, 2007); avaliar o impacto do aumento do número de tentativas de reenvio das mensagens sobre o algoritmo PDSR para alcançar uma taxa de entrega ideal; analisar o impacto da limitação de memória existente nos nós sobre o resultado da taxa de entrega do algoritmo SP/WSN; o impacto sobre o desempenho do SP/WSN com múltiplos *sinks* em uma mesma rede e por fim o desempenho do SP/WSN com variação do valor de *threshold* de envio da mensagem *Reluctance Change*.

Referências Bibliográficas

AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, v. 3, p. 325–349, 2005.

AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. *Computer Networks*, v. 38, p. 393–422, 2002.

ALBINI, L. et al. Reliable routing in wireless ad hoc networks: The virtual routing protocol. *Journal of Network and Systems Management*, Springer New York, v. 14, p. 335–358, 2006. ISSN 1064-7570. 10.1007/s10922-006-9035-8. Disponível em: <<http://dx.doi.org/10.1007/s10922-006-9035-8>>.

AVIZIENIS, A. et al. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 1, p. 11–33, January 2004. ISSN 1545-5971. Disponível em: <<http://dx.doi.org/10.1109/TDSC.2004.2>>.

BEIN, D. Self-configuring, self-organizing, and self-healing schemes in mobile ad hoc networks. In: *Guide to Wireless Ad Hoc Networks*. [S.l.]: Springer London, 2009, (Computer Communications and Networks). p. 27–41. ISBN 978-1-84800-328-6.

BEN-ASHER MORAN FELDMAN, S. F. Y.; GURFIL, P. Distributed decision and control for cooperative uavs using ad hoc communication. IEEE Computer Society, p. 511–516, 2008.

CALSAVARA, A.; LIMA, L. A. P. Routing based on message attraction. In: *Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*. Washington, DC, USA: IEEE Computer Society, 2010. (WAINA '10), p. 189–194. ISBN 978-0-7695-4019-1. Disponível em: <<http://dx.doi.org/10.1109/WAINA.2010.111>>.

CARPES, L. M. Ad*: Algoritmo de roteamento em redes de sensores baseado em inteligência artificial distribuída. Pontifícia Universidade Católica do Paraná, 2007.

CHONG, C.-Y.; KUMAR, S. P. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, v. 91, n. 8, p. 1247–1256, ago. 2003. ISSN 0018-9219. Disponível em: <<http://dx.doi.org/10.1109/JPROC.2003.814918>>.

ELBATT, T.; EPHREMIDES, A. Joint scheduling and power control for wireless ad-hoc networks. 2002.

ELBATT, T.; EPHREMIDES, A. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, IEEE Communications Society, Piscataway, NJ, USA, v. 3, n. 1, p. 74–85, jan. 2004. ISSN 1536-1276. Disponível em: <<http://dx.doi.org/10.1109/TWC.2003.819032>>.

FREITAS, E. P. de et al. Experimental analysis of coordination strategies to support wireless sensor networks composed by static ground sensors and uav-carried sensors. IEEE Computer Society, Washington, DC, USA, p. 152–161, 2010. Disponível em: <<http://dx.doi.org/10.1109/ISPA.2010.72>>.

GEHRKE, J.; MADDEN, S. Query processing in sensor networks. *IEEE Pervasive Computing*, v. 3, n. 1, p. 46–55, 2004.

GEORGE DOLLIMORE JEAN, K. T. C. Sistemas distribuídos conceitos e projeto. Bookman, 2007.

GHADA, K.; LI, J.; JI, Y. Cross-layer design for topology control and routing in manets. *Wireless Communications and Mobile Computing*, John Wiley and Sons, Ltd., v. 12, n. 3, p. 257–267, 2012. ISSN 1530-8677. Disponível em: <<http://dx.doi.org/10.1002/wcm.957>>.

HEIDEMANN, J. et al. Building efficient wireless sensor networks with low-level naming. *SIGOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 35, p. 146–159, October 2001. ISSN 0163-5980. Disponível em: <<http://doi.acm.org/10.1145/502059.502049>>.

JAGANNATHAN, e. a. S. Wireless ad hoc and sensor networks: Protocols, performance and control. CRC Press, 2007.

JOHNSON, D. B.; MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. Kluwer Academic Publishers, p. 153–181, 1996.

JUN, J. A. Z. Wireless sensor networks a networking perspective. Wiley-IEEE Press, 2009.

KAMAT, U.; ZHANG, Y.; OZTURK, C. Enhancing source-location privacy in sensor network routing. In: *In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS*. [S.l.: s.n.], 2005. p. 599–608.

KöPKKE, A. et al. Simulating wireless and mobile networks in omnet++ the mixim vision. In: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, (Simutools '08). p. 71:1–71:8. ISBN 978-963-9799-20-2. Disponível em: <<http://portal.acm.org/citation.cfm?id=1416222.1416302>>.

LEVIS, P. et al. Tossim: accurate and scalable simulation of entire tinyos applications. ACM, New York, NY, USA, p. 126–137, 2003. Disponível em: <<http://doi.acm.org/10.1145/958491.958506>>.

LI, N. et al. Privacy preservation in wireless sensor networks: A state-of-the-art survey. *Ad Hoc Networks*, v. 7, n. 8, p. 1501 – 1514, 2009. ISSN 1570-8705. <ce:title>Privacy and Security in Wireless Sensor and Ad Hoc Networks</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870509000407>>.

LI, Y.; REN, J. Preserving source-location privacy in wireless sensor networks. In: *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*. Piscataway, NJ, USA: IEEE Press, 2009. (SECON'09), p. 493–501. ISBN 978-1-4244-2907-3. Disponível em: <<http://dl.acm.org/citation.cfm?id=1687299.1687355>>.

LIMA, L.; CALSAVARA, A. Autonomic application-level message delivery using virtual magnetic fields. *Journal of Network and Systems Management*, Springer New York, v. 18, p. 97–116, 2010. ISSN 1064-7570. 10.1007/s10922-009-9145-1. Disponível em: <<http://dx.doi.org/10.1007/s10922-009-9145-1>>.

MACEDO, M.; GRILO, A.; NUNES, M. Distributed latency-energy minimization and interference avoidance in tdma wireless sensor networks. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 53, p. 569–582, April 2009. ISSN 1389-1286. Disponível em: <<http://dl.acm.org/citation.cfm?id=1514443.1514688>>.

- MEHTA, K.; LIU, D.; WRIGHT, M. Location privacy in sensor networks against a global eavesdropper. In: *ICNP'07*. [S.l.: s.n.], 2007. p. 314–323.
- MIN, C.-H.; KIM, S. On-demand utility-based power control routing for energy-aware optimization in mobile ad hoc networks. *J. Netw. Comput. Appl.*, Academic Press Ltd., London, UK, UK, v. 30, p. 706–727, April 2007. ISSN 1084-8045. Disponível em: <<http://portal.acm.org/citation.cfm?id=1224250.1224468>>.
- MISRA, S.; WOUNGANG, I.; MISRA, S. C. *Guide to Wireless Ad Hoc Networks*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2009. ISBN 1848003277, 9781848003279.
- NIKOLAOS, A. P.; DIMITRIOS, D. V. Energy efficiency and power control. In: ZHENG, J.; JAMALIPOUR, A. (Ed.). *Wireless Sensor Networks: A Networking Perspective*. [S.l.]: WILEY, 2008. p. 67–143.
- PAN, M.-S.; TSENG, Y.-C. Quick convergecast in zigbee beacon-enabled tree-based wireless sensor networks. *Comput. Commun.*, Butterworth-Heinemann, Newton, MA, USA, v. 31, p. 999–1011, March 2008. ISSN 0140-3664. Disponível em: <<http://dl.acm.org/citation.cfm?id=1349885.1349988>>.
- Energy Aware Routing Protocol for Heterogeneous Wireless Sensor Networks*. [S.l.]: IEEE Computer Society, 2005. 133-137 p. ISBN 0-7695-2424-9.
- POTTIE, G. J.; KAISER, W. J. Embedding the Internet: wireless integrated network sensors. *Communications of the ACM*, v. 43, n. 5, p. 51–51, maio 2000. ISSN 0001-0782. Disponível em: <<http://www.acm.org/pubs/citations/journals/cacm/2000-43-5/p51-pottie/>>.
- RUBINSTEIN, M. et al. A survey on wireless ad hoc networks. In: *Mobile and Wireless Communication Networks*. Springer Verlag, 2006. p. 1–33. Disponível em: <<http://www.gta.ufrj.br/ftp/gta/TechReports/RMCC06.pdf>>.
- SAJAL, K. D.; HABIB, M. A. Routing and data dissemination. In: ZHENG, J.; JAMALIPOUR, A. (Ed.). *Wireless Sensor Networks: A Networking Perspective*. [S.l.]: WILEY, 2008. p. 67–143.
- SCHMIDT-EISENLOHR, F. et al. Selected system models. In: WEHRLE, K.; GÜNES, M.; GROSS, J. (Ed.). *Modeling and Tools for Network Simulation*. [S.l.]: Springer, 2010. p. 235–303. ISBN 978-3-642-12330-6.

- SHAO, M. et al. Towards statistically strong source anonymity for sensor networks. In: *In IEEE INFOCOM*. [S.l.: s.n.], 2008.
- SHIH, E. et al. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In: . [S.l.: s.n.], 2001. p. 272–287.
- VARGA, A. Omnet++. In: WEHRLE, K.; GÜNE?, M.; GROSS, J. (Ed.). *Modeling and Tools for Network Simulation*. [S.l.]: Springer Berlin Heidelberg, 2010. p. 35–59. ISBN 978-3-642-12331-3.
- WEI, X. et al. Optimized priority based energy efficient routing algorithm for mobile ad hoc networks. *Ad Hoc Networks*, v. 2, n. 3, p. 231 – 239, 2004. ISSN 1570-8705. <ce:title>Quality of service in ad hoc networks</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870504000277>>.