

**EFICIÊNCIA DE ENERGIA ATRAVÉS DE
COLETA PERIÓDICA
EM REDES DE SENSORES SEM FIO**

FELIPE LEITE LOBO

**EFICIÊNCIA DE ENERGIA ATRAVÉS DE
COLETA PERIÓDICA
EM REDES DE SENSORES SEM FIO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal do Amazonas como requisito para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: PROF. DR. HORÁCIO A. B. FERNANDES DE OLIVEIRA

Manaus

Março de 2012

© 2012, Felipe Leite Lobo.
Todos os direitos reservados.

Lobo, Felipe Leite
D1234p Eficiência de energia através de coleta periódica em
Redes de Sensores Sem Fio / Felipe Leite Lobo. —
Manaus, 2012
xvi, 42 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal do
Amazonas

Orientador: Prof. Dr. Horácio A. B. Fernandes de
Oliveira

1. Redes de sensores sem fio. 2. coleta periódica.
3. energia. I. Título.

CDU 519.6*82.10

[Folha de Aprovação]

Quando a secretaria do Curso fornecer esta folha,
ela deve ser digitalizada e armazenada no disco em formato gráfico.

Se você estiver usando o `pdflatex`,
armazene o arquivo preferencialmente em formato PNG
(o formato JPEG é pior neste caso).

Se você estiver usando o `latex` (não o `pdflatex`),
terá que converter o arquivo gráfico para o formato EPS.

Em seguida, acrescente a opção `approval={nome do arquivo}`
ao comando `\ppgccufmg`.

Resumo

Uma das principais aplicações existentes em Redes de Sensores Sem Fio (RSSFs) é o monitoramento ambiental, mais particularmente no acompanhamento da fauna existente em locais diversos como florestas, campos, etc. Nesses ambientes de difícil acesso, a manutenção dos sensores fica prejudicada e a restrição de energia torna-se um fator primordial para a vida útil da rede. Como na maioria desses casos os dados não são necessários em tempo real, neste trabalho é proposto e avaliado um novo algoritmo *cross-layer*, denominado PWAVE, voltado para RSSFs com coleta periódica, onde as transmissões dos dados são feitas através de intervalos predeterminados. Enquanto essas transmissões não são executadas, o sensor permanece coletando as informações do ambiente e seus rádios desligam-se, a fim de reduzir o consumo de energia. Diversos outros desafios aparecem diante deste modo de trabalho como, por exemplo, a sincronização dos relógios, que são prontamente solucionados pelo algoritmo proposto. Os resultados mostram que a solução proposta é eficiente e consegue economizar até 51.2% de energia com 99% na taxa de entrega de dados quando comparado com soluções similares.

Palavras-chave: Redes de Sensores Sem Fio, coleta periódica, consumo de energia.

Abstract

One of the main applications on Wireless Sensors Networks (WSNs) is the environmental monitoring, particularly in monitoring of the fauna in diverse places like forests, fields, etc. In these environments that are of difficult access, the maintenance of the sensors is impaired and energy restriction becomes a major factor in the life of the network. How in most such cases the data are not needed in real time, in this work is proposed and evaluated a new cross-layer algorithm, called PWAVE, turned to WSNs with periodic data collection, where data transmissions are made through a predetermined interval. While such transfers are not performed, the sensor still collecting information from the environment and turn off their radios to reduce energy consumption. Several other challenges appear before this way of working, for example, the synchronization of clocks, which are readily solved by the algorithm. The results show that the proposed solution is efficient and can save up to 51.2 % of energy with 99 % in the rate of data delivery when compared with similar solutions.

Keywords: Wireless Sensors Networks, periodic data collection, energy consumption.

Lista de Figuras

1.1	Tamanho dos Diferentes Tipos de Sensores	1
2.1	Frame: Figura adaptada [Rohde, 2009]	5
2.2	Camadas de Rede	6
2.3	Flooding: Mesma informação por mais de um caminho.	9
3.1	Funcionamento básico do esquema de sincronismo do período de dormência dos nós. Figura adaptada[Yang & Niu, 2009]	14
3.2	Funcionamento básico do T-MAC. Figura adaptada [Dam & Langendoen, 2003]	17
4.1	Árvore de Roteamento	22
4.2	Agendamento Coordenado de Coleta Periódica	22
4.3	Ajuste de Tslot	23
5.1	Energia X Perda de Pacotes.	29
5.2	Energia X Perda de Pacotes.	31
5.3	Impacto dos Rádios Transceptores.	32
5.4	Taxa de Perda de Pacotes.	32
5.5	Impacto da Densidade.	33
5.6	Impacto da imprecisão dos relógios.	34
5.7	Exemplos de Tslots dessincronizados, tornando inviável o PWAVE	34

Lista de Tabelas

5.1	Parâmetros de Simulação	28
5.2	Parâmetros Padrões de Simulação	28

Sumário

Resumo	vii
Abstract	ix
Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
1.1 Contexto e Motivações	2
1.2 Objetivos Gerais e Específicos	3
1.3 Organização do Trabalho	4
2 Fundamentos	5
2.1 Introdução	5
2.2 Economia de Energia	5
2.3 Roteamento em Redes de Sensores Sem Fio	6
2.3.1 Camada Física	7
2.3.2 Camada de Enlace	7
2.3.3 Camada de Rede	7
2.3.4 Camada de Transporte	7
2.3.5 Camada de Aplicação	8
2.3.6 Classificação de Algoritmos	8
2.4 Relógios dos Sensores	9
2.4.1 <i>Clock Skew</i>	10
2.4.2 <i>Clock Drift</i>	10
2.4.3 <i>Offset</i>	10
2.5 Comunicação Sem Fio	11
2.6 Conclusões	12

3	Trabalhos Relacionados	13
3.1	Introdução	13
3.2	Algoritmos de Roteamento	13
3.3	Algoritmos de Camada de Acesso ao Meio	16
3.4	Algoritmos Baseados em <i>Clusters</i> e Eleição de Líderes	17
3.5	Algoritmos de Coleta Periódica	18
3.6	Algoritmos de Sincronização	19
3.7	Conclusões	20
4	PWAVE - Algoritmo de roteamento para RSSFs com coleta periódica	21
4.1	Introdução	21
4.2	Descrição do Algoritmo	21
4.3	Sincronização no PWAVE	24
4.4	Colisão de Dados	25
4.5	Dinâmica da Rede	26
4.6	Conclusões	26
5	Avaliação e Resultados	27
5.1	Introdução	27
5.2	Metodologia	27
5.3	Resultados Obtidos	29
5.3.1	Impacto do Intervalo de Coleta X Eficiência em Energia	30
5.3.2	Impacto dos Estados dos Rádios	31
5.3.3	Impacto do Tamanho da Rede	31
5.4	Impacto da Densidade	33
5.5	Impacto do Erro de Sincronização	33
5.6	Conclusões	35
6	Considerações Finais	37
6.1	Aplicabilidade da Solução Proposta	38
6.2	Trabalhos Futuros	38
	Referências Bibliográficas	39

Capítulo 1

Introdução

Redes de Sensores Sem Fio (RSSFs) é um sistema distribuído composto por um grande número de nós sensores, com a finalidade de monitorar determinado fenômeno ou ambiente, sem intervenção humana direta [Bonnet et al., 2001; Akyildiz et al., 2002]. Estas redes constituem uma ferramenta com grande potencial de monitoramento com aplicações em diversas áreas, como agricultura de precisão, controle industrial, sistemas médicos e monitoramento ambiental [Callaway, 2004; Hill et al., 2004].

As RSSFs sofrem com suas limitações de *hardware* devido ao tamanho dos nós sensores (ver Figura 1.1), os quais têm baixa capacidade de armazenamento, processamento e energia para se manterem ligados por longos períodos [Akyildiz et al., 2002].

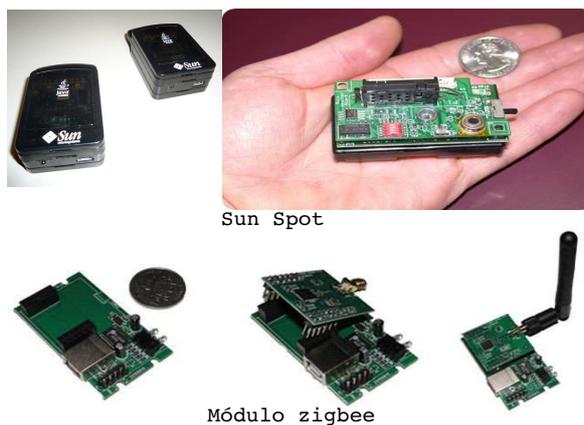


Figura 1.1. Tamanho dos Diferentes Tipos de Sensores

Dentre os desafios para os projetos de uma RSSF, a restrição de energia é um dos maiores, pois os nós sensores funcionam alimentados por baterias e a vida útil destas varia, principalmente, com a quantidade de ações executadas [Raghavendra et al., 2004]. Em geral, os nós estão dispostos em locais de difícil acesso. Desta forma, em

uma determinada rede, alguns sensores podem ficar inoperantes devido ao descarregamento da bateria e a troca desta pode ser inviável dependendo de sua localização. A maior fonte de desperdício de energia dos sensores é basicamente o rádio transceptor, responsável pela comunicação entre nós sensores.

Por sua vez, a comunicação pode ser dividida da seguinte maneira: escutando o meio, em que os rádios dos nós aguardam a comunicação de outros sensores na rede; recebendo dados, onde o nó receptor recebe dados do emissor, e por fim, a transmissão de dados, onde o emissor envia dados ao receptor.

1.1 Contexto e Motivações

Diversos trabalhos têm sido propostos com a finalidade de encontrar novas técnicas para criação de algoritmos que reduzam o consumo de energia e, conseqüentemente, aumentam o tempo de vida da rede [Anastasi et al., 2009]. Algumas técnicas de redução de consumo de energia desligam os rádios dos sensores por um determinado período de tempo [Yang & Niu, 2009], outros algoritmos criam uma árvore de roteamento de dados com o objetivo de maximizar o número de nós folha para depois desligar os rádios durante um período prolongado de tempo [Boukerche et al., 2003], existem também propostas que se apóiam na camada de acesso ao meio [Ye et al., 2004], camada de rede [Macedo et al., 2005] e até mesmo na camada de aplicação [Boukerche et al., 2003].

O protocolo de roteamento PROC (*Proactive Routing with Coordination*) [Macedo et al., 2005] tem como meta diminuir o consumo de energia e maximizar o tempo de vida da rede. Este algoritmo cria uma árvore de roteamento chamada de *backbone*, que define quais nós são os mais apropriados para rotear dados. Por sua vez, o EAD (*Energy-Aware Distributed routing*), proposto por Boukerche et al. [2003], também cria uma árvore de roteamento com o objetivo de maximizar o número de nós folha. Os nós folha não roteiam dados, e portanto podem manter o seu rádio desligado por períodos prolongados de tempo, economizando energia.

Alguns protocolos utilizam técnicas de desligamento dos rádios dos nós sensores. O S-MAC (*Sensor MAC*) [Ye et al., 2004] é um deles, proposto para a camada MAC, tem como objetivo a redução de colisões e consumo de energia, onde os nós permanecem um longo tempo inativos, rádios desligados, e quando algo é detectado, voltam ao estado ativo, rádios ligados.

A eficiência dos protocolos de roteamento é largamente dependente da aplicação da RSSF em questão. Grande parte dos algoritmos propostos são focados em RSSFs com coleta de dados reativa e poucos trabalhos focam na coleta de dados periódica.

Além disso, os trabalhos citados focam apenas em uma camada de comunicação e desprezam os muitos benefícios das soluções integradas, que trabalham em mais de uma camada. Portanto, faz-se necessário um algoritmo que faça a integração entre as camadas de comunicação, com o foco no agendamento coordenado de coleta de dados feito na camada de acesso ao meio e na sincronização dos nós sensores que acontece na camada de rede, voltado para RSSFs com coleta periódica. A nossa solução é baseada em espaço de tempo, onde as transmissões dos dados são feitas através de intervalos predeterminados pelo observador. Enquanto essas transmissões não são executadas, o sensor coleta as informações do ambiente e os rádios ficam desligados.

Este trabalho apresenta e avalia uma nova solução de roteamento para RSSFs com coleta periódica, o PWAVE, que através do uso de tecnologias como arquitetura *cross-layer*, aproveitando-se de informações das camadas de acesso ao meio, juntamente com o *sleep mode* para desligar os rádios dos nós sensores por longos períodos de tempo. E informações da camada de rede para executar a sincronização entre os nós sensores, para que o agendamento coordenado de coleta de dados aconteça, o PWAVE consegue aumentar consideravelmente a economia de energia dos nós sensores e a vida útil da rede como um todo.

O PWAVE foi inicialmente desenvolvido com a finalidade de monitorar e estudar a fauna em ambientes de florestas densas. Redes de sensores neste tipo de ambiente têm sua comunicação afetada devido à degradação dos sinais dos rádios transceptores, ocasionado pelos obstáculos naturais, altas densidades e extremas variações de temperaturas [Figueiredo et al., 2009]. Diante disso, pode-se configurar o algoritmo apresentado para que a coleta periódica aconteça durante partes do dia em que a comunicação é menos afetada pelo ambiente (e.g., durante a noite), a fim de obter maior precisão na taxa de entrega de dados.

1.2 Objetivos Gerais e Específicos

O objetivo geral deste trabalho é desenvolver e avaliar uma nova solução de roteamento para RSSFs com coleta periódica em ambientes de qualidade de comunicação variável, através do uso de tecnologias como arquitetura *cross-layer* e *sleep mode*, que aumente a economia de energia dos nós sensores quando comparado as soluções existentes no estado da arte.

O algoritmo lida com problemas essenciais para o funcionamento pleno da rede, tendo como objetivos específicos:

- resolver o problema da sincronização dos relógios: uma vez que para funcionar

corretamente, os sensores de uma RSSF com coleta periódica precisam ter seus relógios sincronizados. Contudo, manter os relógios sincronizados é também um desafio, pois estes se dessincronizam devido as variações climáticas e condições do ambiente. Um fator importante é que o PWAVE permite solucionar tanto o problema de coleta de dados quanto de sincronização ao mesmo tempo, resultando em economia de energia, quando comparado com o caso em que ambos os problemas são tratados de forma separada.

- tratar a colisão de dados: necessidade de evitar que diversos nós tentem acessar o meio ao mesmo tempo para transmitir seus dados, porque quando isso acontece os pacotes colidem e se perdem sem que nenhum nó receptor os receba. O cuidado com a colisão de dados garante que os pacotes enviados sejam entregues ao nó destino. Este é um dos fatores *cross-layer* de nossa solução (dentre outros), pois estamos utilizando informações da camada MAC para otimizar o acesso ao meio.

1.3 Organização do Trabalho

Este trabalho está organizado como segue. No capítulo 2, são apresentados alguns conceitos de RSSF necessários para o melhor entendimento do trabalho proposto. O estudo do estado da arte, está em trabalhos relacionados, no capítulo 3. No capítulo 4, é proposto o algoritmo PWAVE, um protocolo de roteamento voltado para RSSF com coleta periódica. A performance do algoritmo é avaliada no capítulo 5. No capítulo 6 é descrito o método científico. E por último, serão feitas algumas conclusões e discussões para trabalhos futuros no capítulo 7.

Capítulo 2

Fundamentos

2.1 Introdução

Para termos um melhor entendimento de como funciona a solução desenvolvida neste trabalho, faz-se necessária uma base teórica e conhecer como os protocolos podem ser eficientes em termos de consumo de energia. Este capítulo faz uma análise dos tipos de roteamento existentes e uma classificação dos algoritmos de roteamento. Em seguida, serão brevemente descritos os mecanismos dos algoritmos.

2.2 Economia de Energia

É comum em aplicações de RSSFs os nós permanecerem sem atividades por um período duradouro, com baixa taxa de transmissão de dados. Portanto, não é necessário manter os nós ativos no decorrer do tempo [Polastre et al., 2004]. Em relação ao consumo de energia, há possibilidade de variar a operação cíclica dos nós da rede entre dois períodos, ativos e inativos [Dam & Langendoen, 2003; Ye et al., 2004]. Quando ativo (*listen*), o nó pode efetivar comunicações, e quando inativo (*sleep*), deve conservar seu rádio desligado, poupando energia. Os nós da rede podem alternar entre as fases de *listen* e *sleep*, quando dividido seu ciclo de operação. Como mostra a Figura 2.1, após os nós passarem pelas fases citadas, completa-se o ciclo, chamado de *frame*.



Figura 2.1. Frame: Figura adaptada [Rohde, 2009]

Entendido como funciona uma das técnicas de economia de energia, precisamos entender como funciona o roteamento e como são classificados os algoritmos atuantes em uma RSSF.

2.3 Roteamento em Redes de Sensores Sem Fio

Algumas características particulares das RSSFs tornam inviável o uso de protocolos de redes tradicionais, por necessitarem além de memória, de outros recursos inexistentes em nós sensores. Em razão disso, novos protocolos foram desenvolvidos com a finalidade de atender essas restrições, tornando possível o emprego das RSSFs [Rohde, 2009]. Os protocolos de roteamento devem possuir algumas características:

- Escalabilidade: prever e continuar funcionando bem, apesar do aumento ou diminuição do número de nós da rede;
- Simplicidade: oferecer os serviços com a quantidade mínima de processamento possível;
- Robustez: chegar a uma resposta aceitável e funcionar corretamente, sem que ocorram falhas no sistema;
- Justiça: todos os nós devem conseguir ter acesso aos recursos providos pela rede.

Segundo Campus [2003], os protocolos de comunicação em RSSF são calcados nos moldes de camadas OSI. Entretanto, devido às limitações dos nós sensores, RSSFs não utilizam todas as camadas de rede. Diante disso, os protocolos podem ser reduzidos nas seguintes camadas: Física, Enlace, Rede, Transporte e Aplicação [Rohde, 2009], ver Figura 2.2.

OSI	X	RSSF
Aplicação		Aplicação
Apresentação		Transporte
Sessão		
Transporte		Rede
Rede		
Enlace		Enlace
Física		Física

Figura 2.2. Camadas de Rede

2.3.1 Camada Física

A responsabilidade por transmitir o *bitstream* para os dois lados da comunicação é da camada física. Já em RSSFs, para esta transmissão utiliza-se dispositivos de comunicação de rádio frequência, dispostos de maneira aleatória, desenvolvida através de ondas eletromagnéticas, formando um raio de comunicação no nó sensor [Callaway, 2004]. A variação da frequência e potência do dispositivo resulta em maior alcance da comunicação. O módulo de rádio dos nós sensores atua transmitindo dados, recebendo dados, escutando o meio e o modo desligado. Essas quatro variações influenciam no consumo de energia da rede, motivando a descobrir novos meios de economizá-la.

2.3.2 Camada de Enlace

A camada de enlace detecta erros, corrigindo-os na transmissão de dados, compartilhando-os em *frames*, controlando seu fluxo entre os dois lados da comunicação, exige condições necessárias que variam de acordo com sua aplicação. Em RSSFs os nós podem desligar seus rádios transceptores durante um duradouro espaço de tempo, ligando-os subitamente quando algo é detectado pela rede [Zhao & Guibas, 2004]. Na camada de enlace os protocolos são conhecidos como protocolos de acesso ao meio ou protocolos MAC (*Medium Access Control*), um exemplo é o S-MAC [Ye et al., 2004].

2.3.3 Camada de Rede

A camada de rede é essencial para as redes de computadores, em RSSFs não é diferente [Callaway, 2004; Campus, 2003], sua principal função é fornecer o serviço de roteamento, o qual determina o processo em que a rede reconhece localiza um caminho entre a origem e o destino dos pacotes transmitidos [Rohde, 2009].

2.3.4 Camada de Transporte

A camada de transporte é responsável por fornecer um serviço de comunicação, onde mensagens são aceitas facultativamente, desconsiderando todo o princípio das camadas anteriores. Redes tradicionais permitem perda de dados, portanto uma aplicação para RSSFs que tenha como foco garantir o envio de dados deve ocasionar um desperdício de energia [Callaway, 2004].

2.3.5 Camada de Aplicação

É a camada de rede responsável por realizar a interação entre o protocolo de comunicação e o aplicativo (observador) que requisita as informações a serem coletadas. Em RSSFs temos como exemplo o protocolo SMP (*Sensor Management Protocol*).

O PWAVE trabalha com as camadas de acesso ao meio e de rede em conjunto. Assim, alcança com êxito o agendamento coordenado de coleta periódica, no qual os nós sensores desligam seus rádios transceptores por longos períodos de tempo com o objetivo de economizar energia. O algoritmo atua também na camada de rede, criando a tabela e a árvore de roteamento de forma hierárquica em diferentes alturas, encaminhando os dados coletados até o nó destino.

2.3.6 Classificação de Algoritmos

Algoritmos para RSSF podem ser classificados das seguintes formas: estáticos ou dinâmicos, distribuídos ou centralizados, pró-ativos, reativos ou híbridos, um caminho ou vários caminhos, planos ou hierárquicos, *host* inteligente ou roteador inteligente, interdomínio ou intradomínio, estado do enlace ou vetor distância [Redin, 2004].

Abordagens mais comuns utilizadas para solucionar problemas de roteamento são: inundação (*flooding*), estado do enlace e vetor distância. O entendimento dessas abordagens servem de base para a maior parte dos algoritmos de roteamento existentes.

- *Flooding*: é considerada a abordagem mais simples de todas. Um nó envia pacotes para todos os outros ao seu alcance, os nós que recebem estes pacotes, repassam para outros nós, menos para o que lhe enviou a mensagem. Essa abordagem é eficaz, porém, em algum momento, o nó destino vai receber esses pacotes mais de uma vez, como mostra a Figura 2.3. Portanto, surgem dois problemas, o de *loops* de roteamento e o de superimposição, quando um nó recebe a mesma informação de dois nós diferentes. Em ambos os casos, a redundância dos dados recebidos resulta em desperdício de energia da rede. Desta forma, o algoritmo torna o *flooding* proibitivo [Cavalcanti & Brandão, 2005]. Um protocolo que utiliza o *flooding* como base é o FLOOD apresentado por Kunz [2003].
- Estado de Enlace: grande potencial de fornecer rotas, levando em consideração vários critérios, graças a sua rápida convergência [Redin, 2004]. Em RSSFs, quando nós da vizinhança mudam de estado, um nó ao perceber esta mudança, faz um *flooding* pela rede. Assim, todos os nós da rede ficam ciente dessa mudança quando recebem o pacote, tornando possível prever *loops* na rede, pois o estado do enlace cria as rotas de maneira centralizada. Uma desvantagem do estado de

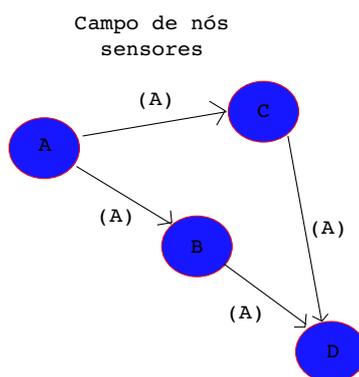


Figura 2.3. Flooding: Mesma informação por mais de um caminho.

enlace é que a rede pode falhar quando a topologia varia, fato recorrente em redes *Ad Hoc* [Redin, 2004]. Um exemplo de protocolo baseado em estado de enlace é o DSR (*Dynamic Source Routing*) [Johnson et al., 2001].

- **Vetor Distância:** esta abordagem cria e mantém a tabela de roteamento com o menor caminho entre os nós. A tabela é atualizada periodicamente, e comparada com a atual, se houverem rotas com caminhos menores, a tabela é armazenada, caso contrário, a mesma é descartada. O AODV (*Ad Hoc on-demand Distance Vector routing*) [Perkins & Royer, 1999] é um exemplo de protocolo que tem como base o vetor distância, tendo como características a eficiência computacional e a baixa convergência quando a topologia varia com uma certa frequência, com tendências a criar *loops*, devido as circunstâncias instáveis e modificações constantes de topologia, como por exemplo, redes *Ad Hoc* [Redin, 2004].

O algoritmo PWAVE é voltado para RSSFs homogêneas, hierárquicas, com distribuições irregulares e densidade variável, onde o sensoriamento do ambiente é reativo, com disseminação programada dos dados. A coleta periódica desses dados é realizada através de múltiplos saltos percorridos pelos pacotes na rede, implicando no aumento do atraso fim-a-fim. Porém no nosso trabalho esse atraso não tem importância, pois o foco é economizar energia sem que a rede perca sua funcionalidade plena. Assim podemos classificar o PWAVE como um algoritmo baseado em *flooding*.

2.4 Relógios dos Sensores

Algumas aplicações de RSSFs necessitam de informações exatas do momento em que os eventos ocorrem, como por exemplo, detecção de incêndios em florestas. Diante disso, os nós sensores responsáveis por essas informações devem ter seus relógios corretamente

ajustados. Entretanto, manter os relógios sincronizados não é uma tarefa simples, pois os nós sofrem com problemas denominados de *clock drift* e *clock skew*, causando diferença entre seus relógios. Esta diferença é referida como erro de *offset*.

Basicamente existem três razões para a diferença entre os relógios: nós iniciarem seus respectivos relógios em momentos diferentes; cristais de quartzo funcionarem em frequências diferentes, o que resulta em uma divergência gradual (*clock skew*); frequência variar de acordo com as condições ambientais (*clock drift*) [Ganeriwal et al., 2005].

2.4.1 *Clock Skew*

Os cristais de quartzo possuem osciladores típicos que podem variar 100 partes por milhão, ou seja, aproximadamente 0,6 milissegundos a cada minuto [Vig, 1992]. Essas oscilações são denominadas de *clock skew*.

2.4.2 *Clock Drift*

Os relógios dos sensores têm seu tempo gerado pelo cristal de quartzo, que embora trabalhem na mesma frequência, existe uma margem de erro. Assim, com o passar do tempo e levando em consideração que os sensores foram iniciados no mesmo instante, os relógios podem não marcar o mesmo tempo. Este problema é causado pela variação da frequência do cristal devido a ruídos, temperaturas, envelhecimento, entre outros [Tjoa et al., 2004].

2.4.3 *Offset*

O *offset* nada mais é que a diferença de tempo entre um relógio e outro, considerando que os mesmos não sofrem interferência de *clock drift* [Ganeriwal et al., 2005]. Por exemplo, um nó começa em um momento diferente de outro na rede, modelado na Equação 2.1:

nó 1 começa no tempo 0s.

nó 2 começa no tempo 3s.

$$\Delta(t_2 - t_1) = 3T + (t_2 - t_1), \quad (2.1)$$

se os nós começam no mesmo instante, então o *offset* é modelado como a Equação 2.2.

$$\Delta(t_2 - t_1) = t_2 - t_1, \quad (2.2)$$

Apesar do cenário deste trabalho se tratar de coleta periódica e não precisar, a princípio, de respostas imediatas aos eventos, como citado acima, mesmo assim a sincronização é necessária para gerar uma linha de tempo dos eventos que ocorreram bem como permitir que os nós sensores se liguem em tempos coordenados que possibilite o envio dos dados coletados ao nó *sink*.

2.5 Comunicação Sem Fio

Observado que as variações de temperaturas e umidades extremas influenciam no funcionamento dos relógios, percebeu-se que estes fenômenos naturais também têm efeitos sobre a comunicação entre os nós em RSSFs.

A comunicação sem fio é feita através de propagação de ondas eletromagnéticas, onde o transmissor tem sua energia elétrica convertida em energia eletromagnética que é irradiada. Por sua vez, o receptor capta essa energia eletromagnética transmitida e a converte em energia elétrica.

Existem três mecanismos básicos de propagação que influenciam em um sistema de comunicação móvel [Rappaport, 2001]:

- Reflexão: é quando a onda se choca com um obstáculo de dimensões maiores que seu comprimento de onda. Esse mecanismo influencia num sistema de comunicação no sentido de atrapalhar a propagação da onda por prédios e construções grandes, além do próprio solo. Outro problema que surge com a reflexão é a interferência por múltiplos caminhos, em que um mesmo sinal pode chegar ao receptor por visada direta ou através de reflexão, causando redundância na informação transmitida e assim degradando a qualidade do sinal;
- Difração: ocorre quando a onda encontra uma aresta de um objeto fazendo com que a onda sofra curvaturas, viabilizando assim comunicação entre transmissor e receptor mesmo quando não há linha de visada;
- Dispersão: quando a onda encontra muitos objetos de comprimento inferior ao seu ou quando encontra uma superfície áspera, fazendo com que o sinal se espalhe para todas as direções induzindo ao mesmo problema abordado na reflexão, o de interferência causada por múltiplos caminhos. Alguns exemplos de objetos que causam esse tipo de fenômeno são as folhagens de árvores, pingos de chuva, postes e fios.

O PWAVE é aplicado em ambientes de florestas densas, portanto, a comunicação é afetada principalmente pelo o mecanismo da dispersão, devido aos obstáculos naturais

como as árvores e folhagens, além das temperaturas elevadas e a umidade extrema. Em razão disso, procurou-se agendar as comunicações em partes do dia, por exemplo, durante as noites, em que tais fenômenos naturais não influenciem na propagação das ondas eletromagnéticas.

2.6 Conclusões

Neste capítulo, vimos as características principais e desejáveis para os algoritmos de roteamento e discutimos também três das principais abordagens. Foram apresentadas as camadas de rede atuantes e a classificação das RSSFs. Mostrou-se como o meio influencia na sincronização e na comunicação dos nós sensores.

A partir desses fundamentos teóricos, juntamente com a pesquisa pelo estado da arte, descrito no próximo capítulo, obtivemos a base para a elaboração do nosso algoritmo (PWAVE). Um algoritmo que atua no seguinte cenário: coleta de dados em ambientes de florestas densas, em que não precisamos dos dados coletados continuamente e, principalmente em partes do dia que a comunicação é prejudicada devido aos obstáculos naturais, temperaturas e umidades extremas, onde soluções de RSSFs semi-baseadas em hierarquia e baseadas em coleta periódica são recomendadas.

Capítulo 3

Trabalhos Relacionados

3.1 Introdução

Neste capítulo, serão apresentados alguns trabalhos que abordam algoritmos que buscam reduzir o consumo de energia, que atuam na camada de acesso ao meio, protocolos baseados em *clusters*, de coleta periódica, sincronização e técnicas de organização dos nós aplicados em redes de sensores.

3.2 Algoritmos de Roteamento

Observa-se que técnicas e algoritmos vêm sendo propostos com o foco em eficiência de energia. No trabalho de Yang & Niu [2009], é mostrado um esquema de sincronização apropriado para desligar e ligar os rádios dos sensores de uma rede. Todos os nós são separados entre nós que encaminham dados e nós que escutam. Em seguida, é escolhido um nó como coordenador, o qual é responsável por mandar um *beacon* contendo o comando para desligar os rádios dos nós via *broadcast*, alcançando todos através de múltiplos saltos usando *flooding*. Através do mecanismo de sincronização, os rádios permanecem desligados e após determinado tempo se ligam quase que simultaneamente. Entretanto, a escolha de um coordenador gera um tráfego extra na rede e o conseqüente consumo maior de energia, como mostra a Figura 3.1. Desta forma, no presente trabalho, não existe um nó coordenador para enviar o *beacon*, o desligamento dos rádios é agendado através de um contador *timer*, que, ao expirar, desliga os rádios dos sensores e, após um período de tempo, liga-os novamente.

Um modelo que utiliza um agendamento de ativação periódica dos nós para conservar energia e diminuir a interferência na camada de acesso ao meio, é apresentado em

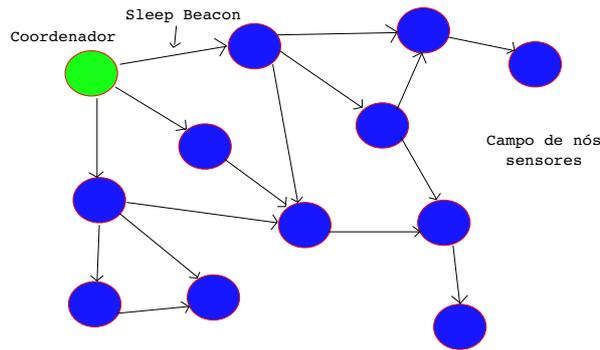


Figura 3.1. Funcionamento básico do esquema de sincronismo do período de dormência dos nós. Figura adaptada[Yang & Niu, 2009]

Trigoni et al. [2007]. Quando os nós não estão transmitindo e nem recebendo dados, têm seus rádios desligados. O agendamento é feito através da localização dos nós, onde cada nó tem o mesmo agendamento em células diferentes para reduzir a interferência e a coleta de dados é realizada de maneira periódica. Nesse modelo, cada nó só consegue transmitir os dados diretamente para os seus quatro vizinhos, o que não é adequado devido ao desperdício de energia, pois esses vizinhos receberão informações redundantes, divergindo da proposta do PWAVE, onde os nós transmitem os dados diretamente para seus respectivos pais na árvore de roteamento, evitando tal desperdício.

O protocolo STORM/AD (*Self-organizing Topology Discovery and Maintenance/Adaptive Diffusion*) [Nakamura et al., 2004], é um algoritmo de super imposição de difusão adaptativa, que opera sobre a topologia de rede criada e mantida pelo algoritmo de auto-organização. O funcionamento do par STORM/AD ocorre da seguinte forma, o algoritmo de auto-organização é executado continuamente, enquanto o algoritmo de roteamento é aplicado sobre a estrutura de rede disponível. O STORM é um algoritmo distribuído para descoberta e manutenção de topologia para RSSFs provendo toda a estrutura necessária para a atividade de disseminação de dados. A topologia resultante é um grafo acíclico direcionado de múltiplos caminhos, onde o fluxo de dados segue dos nós fontes para o nó sorvedouro. Assim, quando um nó precisa enviar dados, ele avalia seus pais, levando em conta métricas como menor caminho e caminho de maior energia, escolhendo para qual será enviado o pacote de dados. No PWAVE, não são levadas em consideração essas métricas utilizadas pelo o STORM, os dados trafegam pela árvore de roteamento a partir dos nós folhas até o *sink*.

O algoritmo PROC (*PRO-active routing protocol for Continuous data dissemination*) [Macedo et al., 2005], é um protocolo de roteamento específico para redes de disseminação contínua de dados, que interage com a aplicação para estabelecer rotas, este mecanismo auxilia na formação de rotas, atribuindo uma pontuação ao nó. Quanto

maior o número de pontos acumulados em um nó, maior a chance do nó ser escolhido como roteador de dados.

Conforme Macedo et al. [2005], o uso de regras confere ao PROC um mecanismo de adaptação e economia de energia, que permite o ajuste da organização de estabelecimento de rotas em tempo de execução, de acordo com as necessidades da aplicação. Então, a aplicação pode interagir com o roteamento, utilizando informações da posição geográfica, informação sensoriada, topologia da rede, estado atual do ambiente e do sensor. Informações estas, que ajudam no estabelecimento de rotas. O algoritmo possibilita que protocolos de controle de topologia desliguem nós que não estão roteando dados, sem que isso influencie no fluxo de dados na rede e ainda prevê mecanismos de tolerância a falhas. O PROC cria uma árvore de roteamento (*backbone*), reconstruído periodicamente por um processo iniciado pelo *sink*, que define quais nós são mais apropriados para rotar dados. Este algoritmo possui um mecanismo de reconstrução local de rotas, que utiliza as mensagens de confirmação de recepção da camada de controle de acesso ao meio (MAC) para identificar nós que não respondem as recepções, assim o nó recalcula a sua rota [Macedo et al., 2005]. Diferentemente do PROC, o algoritmo proposto neste trabalho, utiliza o sistema de disseminação de dados *broadcast*, que envia a mensagem para todos os nós ao seu alcance, desta forma, rotas secundárias são automaticamente utilizadas quando as primárias não estão mais disponíveis, ou seja, não há a necessidade de se reconstruir rotas.

Similar ao PROC, o EAD (*Energy Aware Data centric routing protocol*) [Boukerche et al., 2003], também cria uma árvore de roteamento com o objetivo de maximizar o número de nós folha. Os nós folha não roteiam dados, portanto podem manter o seu rádio desligado por períodos prolongados de tempo, economizando energia. O EAD atrasa a transmissão de dados em um intervalo proporcional a energia residual, que diminui significativamente a quantidade de colisões. O algoritmo depende da reconstrução periódica de rotas para identificar rotas falhas, o fluxo de dados é concentrado em um número reduzido de nós, logo a ocorrência de falhas nestes nós é mais grave que em nós folhas [Boukerche et al., 2003]. O algoritmo EAD utiliza a propagação de uma mensagem de *beacon* para construir as suas rotas. Esta mensagem carrega o identificador do emissor da mensagem, sua energia residual, seu pai na árvore de roteamento e seu estado atual, que pode ser nó folha ou nó não-folha. Depois de construída a rota, a mensagem de *beacon* é enviada primeiro pelos nós com maior energia residual [Boukerche et al., 2003]. O trabalho proposto no presente projeto, utiliza um *flooding* para construir uma árvore de roteamento. Esta mensagem, assim como no EAD, carrega o identificador do emissor e quantos saltos o nó está distante do nó *sink*. Porém o EAD não consegue diminuir uma das fontes de desperdício de energia,

conhecida por escuta ociosa, que ocorre quando o nó sensor mantém o seu rádio ligado mesmo quando não há eventos na rede. O PWAVE soluciona a escuta ociosa através do agendamento sincronizado de coleta periódica.

3.3 Algoritmos de Camada de Acesso ao Meio

Na literatura, existem diversos algoritmos propostos para RSSF na camada de acesso ao meio, um deles é o S-MAC, que procura reduzir a escuta ociosa fazendo com que o ciclo de funcionamento do nó sensor seja dividido em períodos de escuta e dormência, os quais tem como finalidade reduzir o consumo de energia durante os longos tempos de ociosidade, enquanto não são detectados eventos. Durante o período de dormência o nó desliga seu rádio e ajusta um temporizador para acordá-lo mais tarde. Nos períodos de escuta, o nó liga o seu rádio e realiza as devidas transmissões e recepções. Esta técnica de funcionamento de rádio é utilizada neste trabalho, apresentando algumas alterações para evitar o atraso na comunicação devido aos períodos de dormência [Ye et al., 2004].

Para diminuir colisões de dados, o S-MAC trabalha da seguinte forma, quando o nó sensor não consegue acesso ao meio, este entra no estado ocioso, e quando o receptor estiver livre, volta ao estado ativo. Cada nó sabe quanto tempo pode ficar em repouso, pois cada pacote possui a informação de quanto tempo sua transmissão dura, assim o componente evita transmissões desnecessárias e coloca o nó em repouso enquanto seus vizinhos trocam mensagens [Ye et al., 2004]. O S-MAC não é direcionado para coleta periódica de dados, atua de forma reativa, quando detecta um evento os nós passam ao estado ativo para a transmissão de dados.

Similar ao S-MAC, o protocolo T-MAC (*Timeout - MAC*) [Dam & Langendoen, 2003], voltado para coleta periódica e contínua, visa reduzir o tempo de escuta ociosa, utilizando ciclos dinâmicos de atividade e dormência, para diminuir o consumo de energia do nó. O controle do tempo ativo é feito por um temporizador, que ao seu término desliga o rádio, este temporizador não utiliza um intervalo fixo, o nó escuta a rede, transmite e recebe dados durante o seu tempo ativo. A transmissão ou recepção de todos os pacotes é feita em rajadas de tamanho variável. Os nós transmitem suas informações para os seus vizinhos e a recepção de pacotes é suficiente para renovar o temporizador (TA). O funcionamento básico do T-MAC pode ser visto na Figura 3.2.

O intervalo do temporizador deve ser suficiente para receber pelo menos o início de um pacote [Luz, 2004]. Em nosso trabalho, o temporizador é fixo, pois o intervalo de tempo é predeterminado de acordo com a necessidade da aplicação.

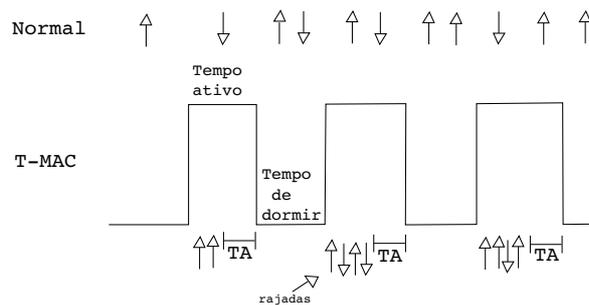


Figura 3.2. Funcionamento básico do T-MAC. Figura adaptada [Dam & Langendoen, 2003]

3.4 Algoritmos Baseados em *Clusters* e Eleição de Líderes

Assim como na camada de acesso, há diversos algoritmos propostos baseados em *cluster* de nós sensores. Os algoritmos baseados em *clusters* em RSSFs surgiram a partir do roteamento hierárquico, o qual envolve formação de *clusters*, onde os nós com menos energia apenas sensoriam o ambiente, enquanto a transmissão é feita pelos nós com mais energia residual. A ideia é de um roteamento eficiente em energia. Existe um líder do *cluster*, que pode ser um nó sensor especial com maior capacidade de energia ou um nó normal, dependendo do algoritmo ou aplicação. O líder também tem funções como a de agregação e compressão de dados para reduzir o número de transmissões até o nó *sink*, economizando energia. Uma das vantagens destes algoritmos é que a latência é minimizada.

O GAF (*Geographic Adaptive Fidelity*) [Xu et al., 2001], apesar de ser um protocolo que utiliza técnicas de controle de topologia (controle de densidade) baseado em localização, foi colocado nesta seção devido aos nós ficarem divididos em *clusters* virtuais e cientes de sua localização, através desta informação cada nó associa-se a um *cluster*. Nós dentro de um mesmo *cluster* são considerados equivalentes, característica esta que permite que alguns nós de uma determinada área sejam completamente desligados, enquanto outros ficam responsáveis pela comunicação e monitoramento do ambiente. Em nossa proposta, os nós têm apenas os rádios transceptores desligados e continuam coletando os dados do ambiente.

Em Israr & Awan [2006], foi apresentado um algoritmo de roteamento de múltiplos saltos, em múltiplas camadas para comunicação entre *clusters*. O algoritmo trabalha com o princípio de divisão e conquista. Traz consigo características de balanceamento de carga e eficiência energética. Explora a propriedade de redundância das RSSFs, selecionando alguns nós da rede e os elege como líderes temporários dos

clusters. Utiliza esses nós para fazer a comunicação *multihop* entre *clusters*. A desvantagem deste algoritmo é que o líder do *cluster* é selecionado aleatoriamente o que ocasionalmente pode afetar uma área de cobertura da rede, se por exemplo, o mesmo nó for eleito várias vezes o líder, este terá a energia residual menor que os demais nós do *cluster* [Israr & Awan, 2006].

Um algoritmo que também utiliza eleição de líder é o RSSR (*Received Signal Strength Routing*) [Boukerche et al., 2008]. Diferente de outras propostas que coletam informações de localização do nó para a eleição de líderes, o RSSR utiliza o indicador de potência de sinal recebido para eleger e selecionar suas rotas. Sua ideia principal é tirar proveito da maior capacidade do nó *sink* equipado de maneira diferenciada dos demais nós da rede, com equipamento de comunicação mais potente, capaz de fazer consultas a todos os nós da rede em apenas um salto, enquanto todos os outros nós são capazes de responder a essas consultas através de múltiplos saltos. Em nosso projeto, utilizamos o nó *sink* para tratar os dados coletados e não para fazer consultas a rede em questão.

3.5 Algoritmos de Coleta Periódica

O protocolo TA-PDC (*Traffic Adaptive-Periodic Data Collection*) [Vahabi et al., 2008] também atua na camada de acesso ao meio, especializado em coleta de dados periódica, consegue reduzir o consumo de energia através da redução dos ciclos dos nós, diminuindo o tempo de escuta ociosa dos mesmos. Os tempos de todos os nós da rede com relação à sua taxa de geração são calculados pelo *sink*. Os nós sensores seguem seus próprios agendamentos que determinam seu tempo de *listen/sleep*. Utiliza um método simples de sincronização centralizada, na qual o *sink* é responsável por iniciar e manter a sincronização, enquanto os outros nós exclusivamente transmitem seus pacotes. O método de adaptação proposto divide a rede em dois grupos: rota principal, onde os nós próximos ao *sink* têm seus períodos de transmissão e recepção maiores que os outros nós; rota de tráfego pesado, na qual os nós são responsáveis por repassar apenas um pacote de dados com *duty cycle* maior. O tempo em que os nós desligam seus rádios é variável e depende da sua localização e do tipo de rota, a qual pertencem.

O TA-PDC reduz a latência e a taxa de perda de pacotes, porém em seus resultados obtidos através das simulações indicam que seu uso é adequado apenas em RSSFs com baixo número de nós sensores. Quando este número ultrapassa a quantidade de dez nós com grandes *duty cycles* o consumo de energia aumenta consideravelmente, o que contraria a proposta do PWAVE.

Outro trabalho que foca a coleta de dados periódica é apresentado em Haque et al. [2010]. É um protocolo baseado na camada de acesso ao meio que utiliza o CSMA (*Carrier Sense Multiple Access*), o qual divide a rede em *clusters* e o líder é equipado com uma fonte de energia maior que os demais nós. Estes, estão apenas a um salto de distância do nó líder. Os nós líderes, são responsáveis por coletar os dados dos nós dentro do seu *cluster* e encaminhá-los até a estação base. Este algoritmo utiliza sucessivas janelas de contenções para manter a taxa de transmissão de dados constante e diminuir a colisão de dados, e consequentemente, o desperdício de energia na rede.

A técnica de utilizar nós líderes com *hardware* diferenciado com maiores recursos é interessante, inclusive em ambientes de florestas densas, onde geralmente o acesso para a manutenção é inviável. Contudo, não o utilizamos como *baseline* para o PWAVE devido em Haque et al. [2010] não ficar claro se os rádios transceptores dos nós são desligados ou não durante o tempo de simulação, focando apenas na comparação das janelas de contenções, colisões de dados e o desperdício de tempo nas sobreposições de janelas de contenções dos modelos IEEE 802.11 e SSCW.

3.6 Algoritmos de Sincronização

Para a solução do problema de sincronização entre os relógios dos nós sensores, pesquisas no estado da arte foram fundamentais para a solução proposta no PWAVE.

O RBS (*Reference Broadcast Synchronization*) [Elson et al., 2002] é um protocolo de sincronização onde nós difundem *beacons* aos seus vizinhos, que utilizam como referência o seu tempo de chegada. Os nós compartilham seus *timestamps* para calcular as oscilações dos relógios e realizar as sincronizações mútuas. Porém o RBS não realiza sincronização através de múltiplos saltos e aumenta o fluxo de pacotes devido a constante troca de mensagens entre os vizinhos.

Um outro protocolo é o TPSN (*Timing-sync Protocol for Sensor Networks*) [Ganerwal et al., 2003], que cria uma estrutura hierárquica e realiza a sincronização par a par, onde cada nó troca duas mensagens com o nó referência, localizado um nível acima na estrutura hierárquica. Seu desempenho é melhor que o do RBS, porém não estima a variação do *clock* e não suporta topologias dinâmicas.

O FTSP (*Flooding Time Synchronization Protocol*) [Maroti et al., 2004] é um protocolo de sincronização que utiliza uma largura de banda de comunicação baixa o que o torna robusto contra falha de nós e *links*. O nó *sink* faz um *broadcast* na rede, os nós recebem o pacote utilizando como referência o tempo do *sink*, fazem o *timestamp* na camada MAC, calculam o seu atraso de transmissão e repassam o pacote com as

devidas correções. Assim todos os nós alcançáveis da rede estarão sincronizados com uma certa precisão e terão realizado o *broadcast*.

Em Werner-Allen et al. [2005] é apresentado o RFA (*Reachback Firefly Algorithm*), um algoritmo de sincronicidade descentralizada, implementado no *TinyOs*. Baseado em um modelo matemático que descreve como os vagalumes e os neurônios do cérebro humano sincronizam seus pulsos espontaneamente. Aplicando o modelo em RSSFs, um nó pisca e seus vizinhos percebem este evento, zerando seu contador de tempo. Permite ainda que nós utilizem informações de atraso do evento passado para o próximo evento. Porém Werner-Allen et al. [2005] não demonstra como o *clock skew*, falhas de *links* e outros parâmetros da topologia afetam este modelo.

3.7 Conclusões

Os projetos de protocolos de RSSFs para aplicações gerais, requerem análises e implementações de hipóteses bastante genéricas, onde raramente são utilizadas em uma aplicação específica.

Em sua maioria, os protocolos para RSSFs aqui apresentados são desenvolvidos para aplicações gerais, coleta de dados reativa, ou periódica em um pequeno intervalo de tempo, focados em apenas uma das camadas de rede, onde a comunicação entre estas é feita através de protocolos adjacentes.

Com o objetivo de economizar energia, explorando características de uma aplicação específica, de monitoramento ambiental, em ambiente de floresta densa, onde a comunicação é afetada diretamente pelo meio em algumas partes do dia, é proposto, neste trabalho, um algoritmo *cross-layer* para RSSFs voltado para coleta periódica. Onde compartilhar informações de energia residual, segurança, tolerância a falhas, desligar os rádios transceptores, em um só algoritmo traz os benefícios de economia de energia, garantia na taxa de entrega de dados, sincronização dos relógios e evita colisões de pacotes. No capítulo a seguir é descrito o protocolo proposto.

Capítulo 4

PWAVE - Algoritmo de roteamento para RSSFs com coleta periódica

4.1 Introdução

Neste capítulo será proposto o algoritmo PWAVE, um protocolo voltado para RSSFs com coleta periódica.

O PWAVE é um protocolo de roteamento integrado baseado em ciclos de tempo, onde as transmissões dos dados são feitas através de intervalos predeterminados. Assim, diferentemente do EAD, o PWAVE consegue diminuir uma das fontes de desperdício de energia, a escuta ociosa. Ao invés de manter seu rádio ligado mesmo quando não há eventos, mantém seus rádios desligados mesmo quando os eventos existem, com o objetivo de reduzir o consumo de energia. Enquanto as transmissões não são executadas, os nós sensores permanecem coletando e processando as informações do ambiente.

4.2 Descrição do Algoritmo

O PWAVE organiza os nós sensores em uma árvore de roteamento de dados. No início do algoritmo, os nós estão todos com seus rádios ligados e então é executado um *flooding* controlado na rede, onde o nó *sink* envia um pacote de dados que alcança todos os nós da rede. Este pacote contém o número de saltos percorridos do *sink* até o nó destino. Cada nó sensor ao receber o pacote, incrementa a variável de número de saltos, fica ciente de qual é a sua altura, quem é seu respectivo pai na árvore e cria sua tabela de roteamento. A solução para montar a árvore de roteamento usada foi escolher como o nó pai o primeiro a receber o pacote. Tal algoritmo é ilustrado na Figura 4.1(a).

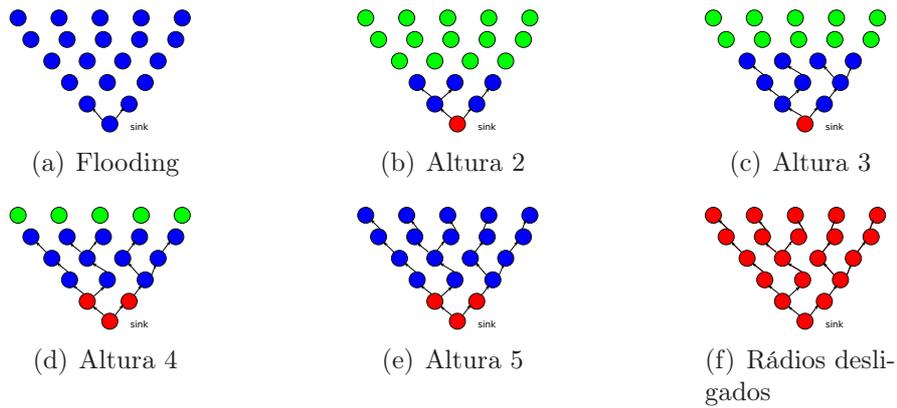


Figura 4.1. Árvore de Roteamento

Em seguida, o nó que acabou de receber o pacote, o repassa para a altura seguinte até que o pacote alcance a rede toda.

Pode acontecer de um nó receber mais de um pacote, pois como vimos anteriormente o *broadcast* envia o pacote para todos os nós ao seu alcance, então ao receber o pacote repetido o nó faz a verificação e o descarta. Após o repasse do pacote, cada nó aciona o seu *timersleep* para o desligamento dos rádios, visando a economia de energia, ilustrado na Figura 4.1(f). Ao expirar este *timer* os nós ligam os rádios durante um intervalo de tempo para que a comunicação seja feita. Entretanto, os nós não se ligam todos ao mesmo tempo, mas de forma ordenada, conforme será explicado a seguir.

A coleta periódica, ilustrada na Figura 4.2, ocorre da seguinte forma: os nós folhas da altura H acordam durante T_{slot} de tempo que é dividido em dois, sendo respectivamente, tempo de receber (TR) e o tempo de enviar dados (TE). Os nós da altura H aguardam o TR. Como não há nenhuma outra altura acima com os nós ligados, estes não irão receber dados. No momento TE da altura H, os nós da altura H-1 ligam seus rádios devendo o TR desta altura coincidir no tempo com o TE da altura acima (H) para que ocorra a comunicação (ver Figura 4.2).

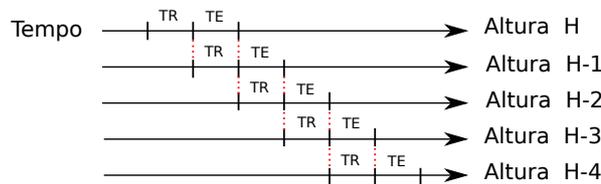


Figura 4.2. Agendamento Coordenado de Coleta Periódica

No PWAVE todos os nós da rede encaminham os dados coletados. Logo, os nós da altura H enviam seus pacotes de dados para seus nós pais da altura H-1, enquanto isso

todos os outros nós, que não pertencem às respectivas alturas da árvore de roteamento permanecem com seus rádios desligados. Assim que o TE dos nós da altura H termina, os sensores desligam seus rádios. Enquanto os nós da altura H-2 ligam seus rádios e seu TR coincide com o TE da sua altura acima (H-1). Então, os nós da altura H-1 agregam seus dados coletados com os dados recebidos da altura anterior e encaminham para altura seguinte (H-2). Este processo se repete até que os dados percorram todas as alturas da árvore de roteamento até chegarem ao destino, o nó *sink*. Desta forma, o PWAVE mantém somente duas alturas por vez ligadas, evitando a escuta ociosa e, principalmente, colisões de pacotes. Esta rotina, a qual se assemelha a uma onda se repete a cada coleta de dados dando origem ao nome do algoritmo, PWAVE.

Como já citado anteriormente, um ponto crucial para que a solução seja aplicada é a sincronização, conforme Vig [1992], no pior caso os relógios de quartzo sofrem uma interferência de 100 ppm para mais ou para menos. Portanto o Tslot de cada altura na árvore é modificado e ajustado de acordo com o período de coleta dos dados. Por exemplo, se a coleta é feita uma vez por semana então os nós podem dessincronizar no pior caso $(7 \cdot 86400) / 10^4 = 60.48$ segundos, para mais ou para menos. Este cálculo feito é baseado conforme mostra a seção 2.4.1 deste trabalho. Então este tempo máximo de dessincronização é adicionado antes do t_i (tempo inicial do Tslot) se esta for para menos, caso contrário é adicionado após o t_f (tempo final do Tslot). Não tem como precisar se esta dessincronização que os relógios de cada nó sofrem é para mais ou para menos.

Assim o Tslot é ajustado de maneira não acumulativa, levando em consideração o pior caso da dessincronização dos relógios tanto para mais (soma-se ao t_f o tempo máximo de dessincronização), quanto para menos (subtrai-se ao t_i o tempo máximo de dessincronização), como demonstrado na Figura 4.3. A cada nova onda de coleta de dados é feita a sincronização, os relógios se ajustam e os nós transmitem seus dados até o *sink*.

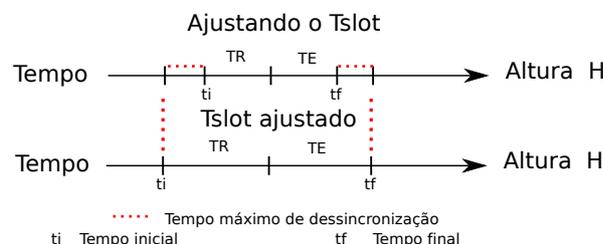


Figura 4.3. Ajuste de Tslot

A sincronização é um dos requisitos necessários para o agendamento coordenado de comunicação entre os nós, para a organização dos *duty cycles* para que nós da

mesma altura na rede possam acordar ao mesmo tempo e para permitir que os dados sejam entregues através de múltiplos saltos até o nó *sink*. Para isso, na próxima seção será explicada a solução de sincronização utilizada pelo PWAVE para que não seja necessária a execução de um algoritmo específico para isso.

4.3 Sincronização no PWAVE

Em cenários onde os sensores são dispostos em locais com grande variação de temperaturas (e.g., florestas e campos), interfere no cristal de quartzo utilizado pelos relógios, podendo variar sua frequência de acordo com a temperatura ou condições do ambiente. Então, faz-se necessária a sincronização dos relógios dos sensores, onde todos os nós da rede devem possuir uma escala de tempo comum entre si, levando em consideração um tempo como referência.

Neste trabalho, é apresentada uma técnica de sincronização em RSSF, que permite um agendamento coordenado de comunicação entre os nós. A solução proposta no PWAVE é fazer com que os relógios dos nós sincronizem com seus vizinhos, mesmo sem um nó de referência. Tal solução se mostrou suficiente para a execução do PWAVE e ao mesmo tempo capaz de economizar energia. Para realizar esta sincronização, utilizamos os pacotes de dados da própria coleta periódica, evitando assim que mais pacotes trafeguem pela rede. No momento em que os pacotes são enviados, os nós receptores armazenam o tempo do relógio do emissor, calcula a média de tempo, levando em consideração o atraso do salto do pacote e o erro de sincronização, atualizam os seus *offsets* e ajustam os seus relógios. Desta forma a cada novo ciclo de coleta de dados os nós sincronizam-se e evitam um atraso acumulativo. Tal solução é detalhada no

Algoritmo 1

```

input : Msg
1  Inicialização das variáveis
2   $TE \leftarrow getMyTime()$ ;
3   $TR \leftarrow 0$ ;
4   $TA \leftarrow 0$ ;
5   $CompOffset \leftarrow 0$ ;
6  if  $Msg = PacotePeriodico$  then
7  Ajuste dos relógios
8   $TE \leftarrow TE + oneHopError()$ ;
9   $TR \leftarrow getMyTime()$ ;
10  $TA \leftarrow (TE + TR)/2$ ;
11  $CompOffset \leftarrow (TR - TA)$ ;
output: Tempo Ajustado

```

Algoritmo 1: Sincronização dos relógios.

onde a variável TE é o tempo de envio do pacote pelo emissor, TR é o tempo de recebimento do pacote pelo receptor, TA é o tempo ajustado dos relógios, $CompOffset$ é quanto tempo o relógio está dessincronizado. A função $getMyTime()$ retorna o tempo marcado no exato momento pelo relógio do sensor e $oneHopError()$ retorna o tempo de atraso de um salto do pacote.

4.4 Colisão de Dados

A colisão de dados é evitada ao manter-se o mínimo de nós ligados ao mesmo tempo. Entretanto, como todos os nós da mesma altura irão ligar seus rádios e transmitir ao mesmo tempo, é ainda possível que mais de um nó transmita simultaneamente seus dados para outro, o que poderia resultar em colisão de dados. Para evitar ainda mais essas colisões, os nós ligam-se ao mesmo tempo, porém no momento da transmissão, cria-se uma janela de contenção de dados, respeitando as características de uma distribuição normal de tempo aleatório dentro de um intervalo pequeno de tempo de poucos milissegundos. Desta maneira, cada nó que tentar acessar o meio, tem uma probabilidade menor de colidir seus dados com os de outro nó.

Além disso, outra característica do PWAVE que contribui para diminuir o problema da colisão, operando em conjunto com a janela de contenção, é o $Tslot$, que por ser dividido em transmissão e recepção, distribuído em tempos diferentes para cada altura da árvore de roteamento, resulta em uma menor chance dos dados colidirem.

Um fator que pode implicar em colisão de dados é o intervalo de coleta de da-

dos. Em intervalos maiores de coleta, provavelmente a quantidade de dados coletados durante este período de tempo será maior, portanto o `Tslot` deve ser ajustado com um intervalo maior para garantir que não haja colisão e que esse tempo seja suficiente para conseguir a transmissão dos dados coletados entre as alturas na árvore. Sendo assim, garantimos que a conexão entre as alturas existe de forma apropriada para a disseminação dos dados na rede.

4.5 Dinâmica da Rede

Com o decorrer do tempo a vida útil da rede diminui e alguns nós podem ter sua energia esgotada antes de outros, ou até mesmo a possibilidade de falha ou inclusão de alguns nós na rede. Quando isto ocorre, faz-se necessária a reconstrução da árvore de roteamento de dados. Esta pode ocorrer de maneira adaptativa de acordo com o intervalo de tempo de coleta de dados, já que quanto maior o tempo de coleta, maior a probabilidade de aumento do tamanho dos dados coletados.

Então a reconstrução da árvore de roteamento pode ser feita uma vez por dia, ou a cada ciclo de coleta de dados. Este é um parâmetro que pode ser ajustado conforme a necessidade da aplicação ou do observador.

4.6 Conclusões

Neste capítulo apresentamos um novo algoritmo de roteamento voltado para RSSFs com coleta periódica, com a finalidade de ser eficiente em energia sem que a rede perca sua funcionalidade. Para verificar sua eficácia, no próximo capítulo é feita a avaliação de desempenho do protocolo proposto.

Capítulo 5

Avaliação e Resultados

5.1 Introdução

Neste capítulo iremos avaliar o desempenho do algoritmo proposto, o PWAVE. Para isso, apresentamos a metodologia de avaliação utilizada, os resultados obtidos, e discutimos suas vantagens e desvantagens.

5.2 Metodologia

Esta seção descreve os processos do trabalho científico realizado, classificação e etapas de seu planejamento.

Neste trabalho utilizamos a aplicação da pesquisa pelo estado da arte, gerando conhecimentos para aplicação do protocolo PWAVE, dirigido à solução de problemas de consumo de energia em Redes de Sensores sem Fio com coleta periódica em ambientes de florestas densas.

A pesquisa teve um procedimento reflexivo e crítico, a partir do desenvolvimento do protocolo e simulações, buscando soluções para o problema descrito no parágrafo anterior. Seu planejamento e a execução fizeram parte de um processo sistematizado compreendendo suas etapas detalhadas como segue.

A avaliação é feita através do simulador de redes NS-2 (*Network Simulator-2*) na versão 2.34 [Ns-2, 2011]. Para a escolha do algoritmo para comparação do PWAVE, um *baseline*, foram feitos experimentos com um conjunto de diversas combinações de protocolos, como descrito a seguir.

Os parâmetros de simulação com relação ao consumo de energia são baseados em Yupho & Kabara [2007] e seus valores estão definidos de acordo com a tabela 5.1.

Tabela 5.1. Parâmetros de Simulação

Parâmetros	Valores
Energia na transmissão	342×10^{-7} joule/byte
Energia na recepção	1888×10^{-8} joules/s
Energia na escuta	1000×10^{-8} joules/s
Energia inicial	50 joules

Os parâmetros padrões utilizados nas simulações estão de acordo com a tabela 5.2.

Tabela 5.2. Parâmetros Padrões de Simulação

Parâmetros	Valores
Campo de sensoriamento	130x130 m
Raio de Comunicação	15 m
Número de nós	128
Densidade	0.02
Intervalo de coleta	86400 s (1 dia)
Tempo de simulação	2592000 s (30 dias)

Os resultados obtidos através dos experimentos, são as médias dos valores após executadas 33 diferentes instâncias (*seeds*). Em todos os gráficos, curvas representam a média dos valores, enquanto as barras de erro, o intervalo de confiança de 95%. Estes resultados obtidos estão descritos na próxima seção.

Para uma avaliação mais justa do desempenho do PWAVE, comparamos sua utilização com o protocolo S-MAC que serve como referência para os protocolos atuais do estado da arte. Porém, foi feita uma adaptação do S-MAC para coleta de dados periódica com longos períodos de dormência dos rádios, fazendo seus rádios permanecerem desligados um período de tempo igual ao do PWAVE. O agendamento programado de coleta de dados também foi igual.

Para comprovar a escolha do *baseline*, comparamos quatro combinações diferentes descritas a seguir. Na primeira combinação, empregamos o PWAVE em conjunto com o protocolo MAC 802.11, padrão do NS-2. Nesta combinação que chamaremos de PWAVE + MAC, inicialmente é feito um *flooding* na rede para a descoberta das rotas que serão utilizadas na coleta periódica. Em seguida os rádios dos sensores são desligados e agendados para ligá-los novamente no momento da coleta. Neste instante, os nós ligam seus rádios e transmitem seus dados coletados para seus respectivos pais na árvore de roteamento.

Posteriormente, foi combinado o PWAVE com o S-MAC (denominada de PWAVE + S-MAC), onde acontece o mesmo processo da combinação anterior, exatamente igual a maneira da descoberta das rotas e de como os rádios são desligados.

Combinamos também, o protocolo de coleta periódica básica que chamamos de algoritmo de árvore, com o MAC 802.11 que denominamos de *Árvore + MAC*, na qual, os rádios dos nós permaneceram ligados durante todo o tempo de simulação e a coleta de dados periódica foi feita através de *flooding*, no qual o nó manda os dados coletados para seu pai na árvore.

Uma outra combinação foi utilizar o protocolo de coleta periódica básica com o S-MAC, que chamamos de *Árvore + S-MAC*, onde é feito um *flooding* para a descoberta das rotas, os nós são desligados pelo protocolo S-MAC e permanecem desligados até que a coleta de dados ocorra.

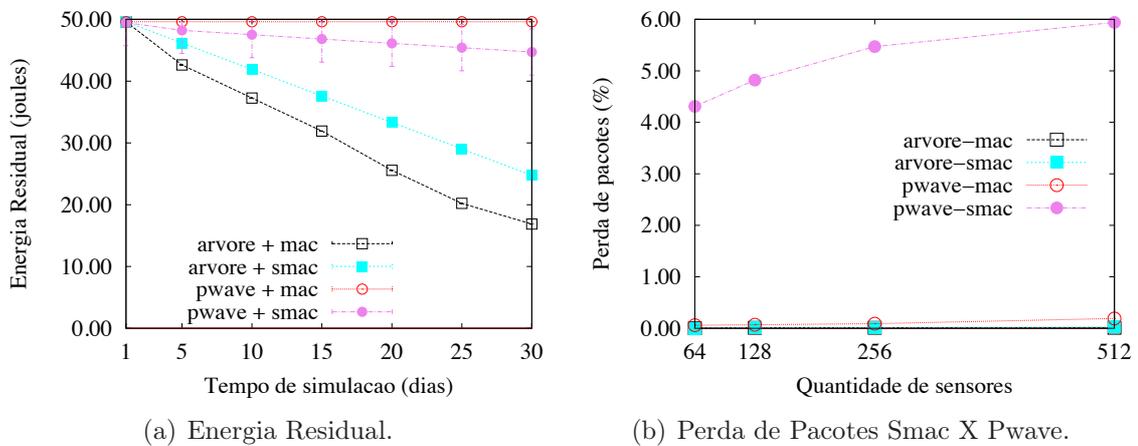


Figura 5.1. Energia X Perda de Pacotes.

5.3 Resultados Obtidos

Conforme ilustrado na Figura 5.1(a) a combinação *Árvore + MAC* obteve o pior desempenho em eficiência de energia, pois nesta combinação os rádios permaneceram ligados durante todo o tempo de simulação. A combinação *Árvore + S-MAC*, já alcançou um desempenho melhor, porque o S-MAC desliga os rádios dos sensores. Observa-se que o *PWAVE + MAC* é a combinação que apresenta maior economia de energia, isto porque o S-MAC utiliza o estado de rádio *idle*, o qual implica em gasto de energia, enquanto o *PWAVE* só emprega três estado: transmissão, recepção e desligado.

Na Figura 5.1(b), pode-se notar que a taxa de perda de pacotes aumenta quando combinado o *PWAVE* com S-MAC, justamente devido ao *PWAVE* possuir seus próprios agendamentos dos períodos de dormência e de transmissão de pacotes, que conflitam quando combinados com S-MAC, que por sua vez também possui seus agendamentos próprios. Nas demais combinações a taxa de perda de pacotes permaneceu estável.

Após feita a análise inicial de acordo com os resultados acima, levando em consideração as métricas de energia residual e taxa de perda de pacotes, comparamos para fins de resultados o PWAVE + MAC *versus* Árvore + S-MAC, que a partir de agora chamaremos respectivamente de PWAVE e SMAC.

O motivo pelo qual escolheu-se o S-MAC para comparação, é porque além dele servir como base para o PWAVE, ele também é base da maioria dos algoritmos propostos na literatura até os dias de hoje. Além disso, o SMAC não mostrou-se compatível com o PWAVE, gerando as perdas de pacotes e o mal funcionamento do algoritmo.

5.3.1 Impacto do Intervalo de Coleta X Eficiência em Energia

Com relação à eficiência em energia, o presente trabalho leva em consideração que um nó pode estar em quatro estados: *sleep mode*, transmissão, recepção e escuta. A principal diferença entre os algoritmos avaliados, é que no PWAVE o nó liga seu rádio somente no momento da coleta periódica, enquanto no S-MAC o rádio é ligado para identificar se algum outro nó deseja se comunicar.

Diante disso, foi criado um cenário, no qual varia-se o intervalo de coleta periódica entre 1, 2, 4 e 7 dias com os valores padrões constantes de nós e densidade. Foi possível notar que, conforme esperado, o intervalo de coleta periódica influencia diretamente no consumo de energia. Quanto maior é o intervalo, mais os algoritmos conseguem economizar energia Figura 5.2(a), e que o aumento desta economia implica em um pequeno aumento da taxa de perda de pacotes. Vale ressaltar a importância da capacidade de armazenamento dos nós sensores neste cenário, pois quando o intervalo de coleta aumenta, consequentemente a quantidade de dados coletados provavelmente será maior. Sendo assim os nós sensores devem ter a capacidade de armazenamento adequada para manterem-se coletando dados do ambiente.

Percebe-se que apesar do aumento da taxa de perda de pacotes, conforme ilustrado na Figura 5.2(b), o PWAVE não superou a marca de 1%. O aumento deve-se ao período de sincronização, pois a mesma é feita junto com a coleta periódica, então quanto maior é o espaço de coleta, maior é o erro de sincronização.

No mesmo cenário comparamos o PWAVE com o S-MAC. O algoritmo proposto PWAVE mostrou-se bem mais eficiente em termos de energia, devido à sua maneira de organização da rede, criando a árvore de roteamento, desligando os rádios e mantendo apenas os rádios de duas alturas ligados por vez. Como mostra a Figura 5.2(a) o PWAVE foi capaz de economizar 51.2% de energia quando comparado ao S-MAC.

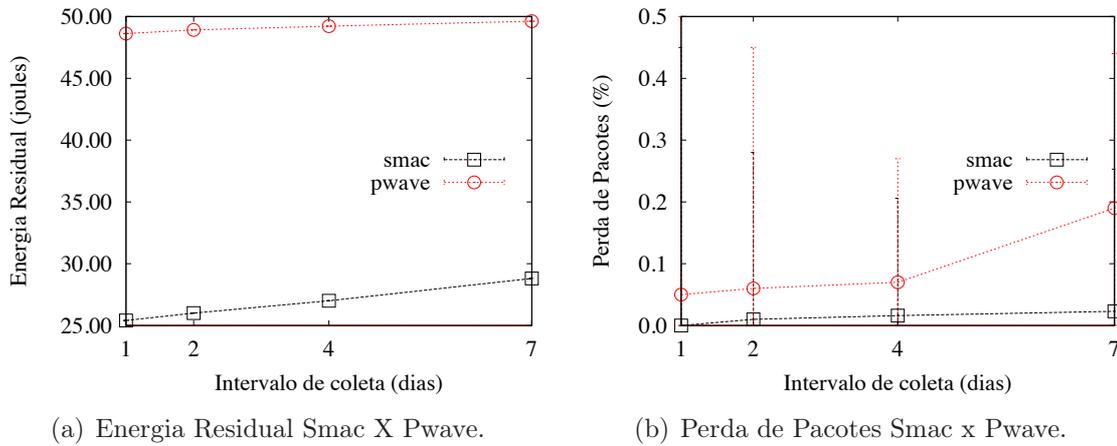


Figura 5.2. Energia X Perda de Pacotes.

5.3.2 Impacto dos Estados dos Rádios

Nesta seção foi avaliado o impacto que os estados dos rádios transceptores exercem na energia residual da rede, utilizando o cenário padrão.

Para tal avaliação, comparou-se o PWAVE com ele próprio, porém feita uma adaptação para não desligar os rádios dos nós, ou seja, os nós mantiveram seus rádios ligados durante todo o tempo de simulação. A Figura 5.3(a) mostra que a solução do PWAVE a energia é visivelmente economizada. Isto acontece porque os rádios transceptores no PWAVE se mantêm ligados apenas T_{slot} de tempo a cada ciclo de coleta no decorrer da simulação, como mostra a Figura 5.3(b), na qual observa-se também que para 30 dias de operação, cada nó ficou ligado durante cinco horas, ficando, em média, ligados dez minutos por dia. Dividido em cinco minutos para a recepção de dados e os outros cinco minutos para a transmissão. Portanto, o PWAVE soluciona prontamente o problema de escuta ociosa.

5.3.3 Impacto do Tamanho da Rede

O impacto do tamanho da rede foi avaliado através do aumento do número de nós sensores de 64 até 512 com uma densidade constante de 0.02 nós/m². O campo de sensoriamento foi redimensionado de acordo com o número de nós. Em todos os casos o fluxo de pacotes transmitidos pela rede aumentou, pois cada nó que sensoria o ambiente e detecta o evento, gera uma mensagem, que é agendada para ser encaminhada até o *sink*. O custo da comunicação é modelado conforme a Equação 5.1, onde n é o número de pacotes. É feito um *flooding* de custo n , depois um ciclo de coleta também de custo n e este se repete todos os dias durante o período de simulação, ou seja, 30 ciclos.

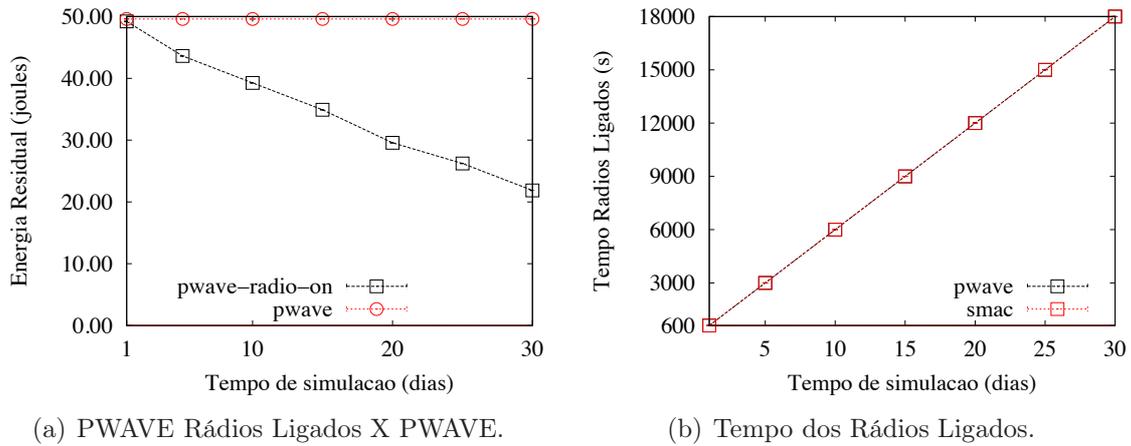


Figura 5.3. Impacto dos Rádios Transceptores.

$$Total_Pacotes = n + n * 30, \quad (5.1)$$

Por isso, quando a quantidade de nós aumenta a tendência é que mais mensagens sejam geradas na rede. A Figura 5.4(a) mostra que o tráfego de pacotes na rede dobra proporcionalmente ao aumento da quantidade de nós.

Apesar do aumento do tráfego, o algoritmo PWAVE atingiu uma taxa de perda de pacotes de 0.19%, já o S-MAC para as quantidades menores de nós sensores (64, 128 e 256 nós) manteve sua média 0.01%, em contrapartida, quando o número de nós foi para 512, sua média de taxa de perda foi de 0.23%. Nota-se na Figura 5.4(b) que o PWAVE somente supera o S-MAC na métrica taxa de entrega de dados, quando o cenário tem a maior quantidade de nós. Ou seja, os dois algoritmos se equiparam neste ponto avaliado.

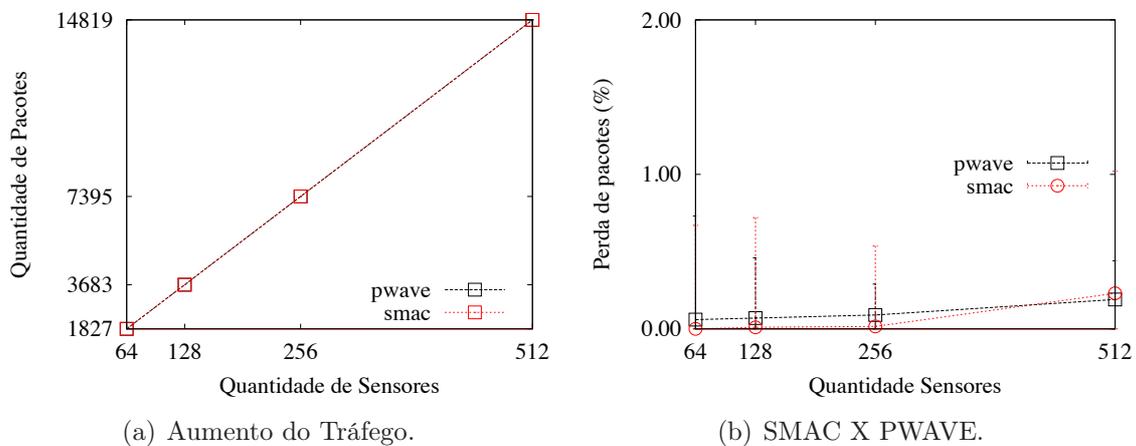


Figura 5.4. Taxa de Perda de Pacotes.

5.4 Impacto da Densidade

Avaliamos o impacto da densidade da rede aumentando a quantidade de nós no mesmo campo de sensores, de modo que a densidade da rede variou de 0.02 , 0.03, 0.04 e 0.05 nós/m². A Figura 5.5(a) mostra que a média de energia residual da rede que diminuiu aproximadamente 0.3% com o aumento da densidade. A densidade impactou na métrica de perda de pacotes, devido a maior concentração do número de nós sensores no espaço. De acordo com a Figura 5.5(b) o aumento na taxa de perda de pacotes foi de 0.4%.

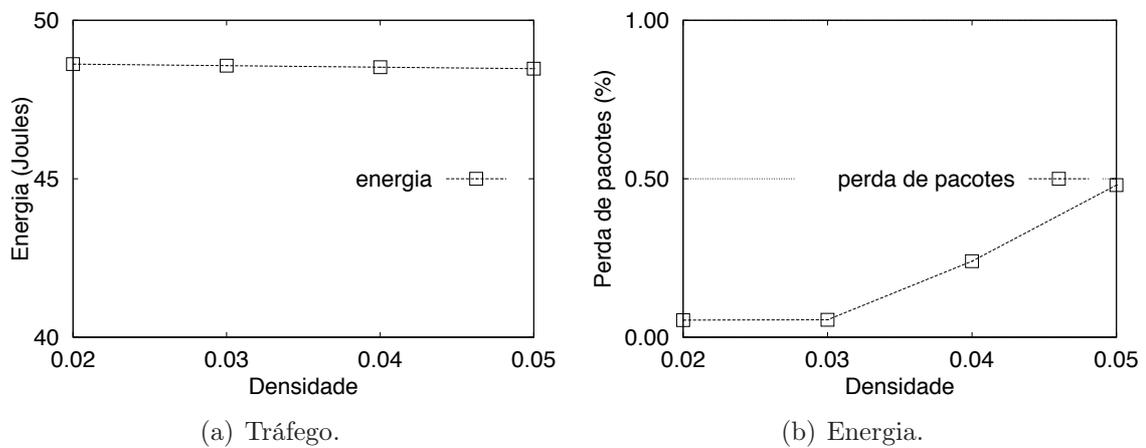


Figura 5.5. Impacto da Densidade.

5.5 Impacto do Erro de Sincronização

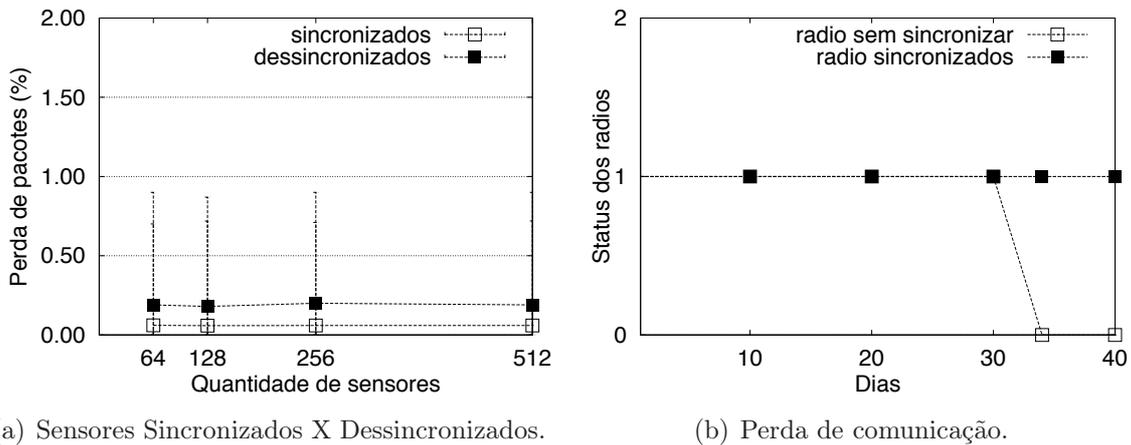
Como citado anteriormente, sabe-se que a sincronização dos relógios é essencial para que a solução proposta pelo PWAVE funcione. Então nesta seção avaliou-se o impacto do erro de sincronização no cenário padrão. Com os relógios sincronizados, a taxa de perda de pacotes atingiu apenas 0.06%. Com os relógios dessincronizados, no pior caso conforme Vig [1992], o quartzo sofre uma interferência de 100 ppm para mais ou para menos. Portanto no cenário padrão, com o intervalo de coleta de 7 dias, temos a Equação 5.2

$$Total_dessincronizacao = 7 * 86400 / 10^4 \quad (5.2)$$

com o erro máximo de dessincronização, a taxa de perda de pacotes alcançou 0.19%. Uma das vantagens do PWAVE é que a taxa de perda de dados é constante mesmo com a variação do número de nós, como nota-se na Figura 5.6(a) em ambos os

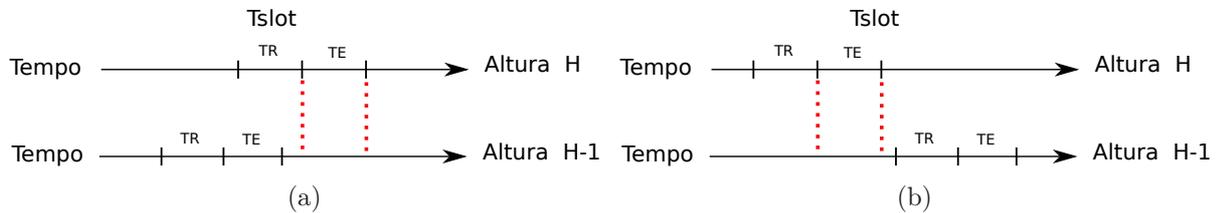
casos a taxa se manteve constante. Isto ocorre devido ao agendamento coordenado de coleta em conjunto com a sincronização dos relógios a cada ciclo de coleta. Caso não houvesse, a dessincronização seria acumulativa e em algum momento no tempo alguns nós da rede ficariam incapazes de se comunicar com o restante da rede.

Para mensurar quando os nós perdem a comunicação, o cenário padrão foi modificado, fixando o número de nós sensores em 256, modificando o PWAVE para que ele deixasse de sincronizar os relógios e aumentando o tempo de simulação para 60 dias. Resultados mostram que foram precisos aproximadamente 34 dias, 17 horas e 16 minutos, de acordo com a Figura 5.6(b), onde 1 é onde os rádios conseguem se comunicar e 0 os rádios perdem a comunicação, tornando inviável a aplicação do PWAVE. Pois o TE da altura anterior deixa de coincidir com o TR da altura seguinte abaixo na árvore de roteamento do PWAVE, ver as Figuras 5.7(a) e 5.7(b).



(a) Sensores Sincronizados X Dessincronizados.

(b) Perda de comunicação.

Figura 5.6. Impacto da imprecisão dos relógios.**Figura 5.7.** Exemplos de Tslots dessincronizados, tornando inviável o PWAVE

Tais resultados mostram que o algoritmo de sincronização descentralizado proposto pelo PWAVE é suficiente para permitir a comunicação organizada dos nós mesmo sem a presença de um nó raiz (*root*) para sincronizar todos os nós.

5.6 Conclusões

Este capítulo apresenta o desempenho do PWAVE, que através da interação de informações entre as camadas de acesso ao meio (MAC) e aplicação, consegue organizar a rede em uma árvore de roteamento de dados e agendar de maneira coordenada a coleta periódica dos dados coletados.

Para a avaliação de desempenho, foram criadas quatro combinações entre os algoritmos PWAVE, S-MAC, 802.11 e o algoritmo de Árvore. Foram avaliadas as métricas de energia residual da rede, perda de pacotes e o tempo que os rádios permaneceram ligados. Avaliou-se também o impacto do intervalo de coleta periódica, dos estados dos rádios sensores (ativos ou inativos) e o erro de sincronização sobre essas métricas.

Os resultados obtidos mostram que a combinação PWAVE + 802.11 consegue ser eficiente em energia quando comparado com as outras combinações. Com relação a taxa de perda de pacote, a mesma combinação não superou a média de 1%, índice este que se equipara com as outras combinações testadas. Estes resultados apontam ainda que o método de desligamento dos rádios é um fator fundamental para a economia de energia e que o tamanho da rede provoca um aumento no fluxo de pacotes que trafegam pela mesma. E por fim, nota-se que a sincronização é imprescindível para a aplicação da solução proposta, pois após determinado tempo sem a sincronização dos relógios o PWAVE deixa de funcionar.

O método de sincronização apresentado no PWAVE atende aos *deadlines* mesmo levando em consideração o atraso que as RSSFs sofrem neste cenário, portanto decidimos não usar a tecnologia TDMA (*Time Division Multiple Access*) nos *slots*, e sim mostrar que o agendamento coordenado pode garantir a não colisão de dados, uma vez que já tem a sincronização.

Capítulo 6

Considerações Finais

No presente trabalho tratamos do problema de energia em redes de sensores com coleta periódica. Inicialmente, apresentamos um embasamento teórico sobre redes de sensores sem fio, classificações, roteamento e algoritmos. Cujo o entendimento é fundamental para a compreensão do funcionamento da proposta aqui apresentada e dos resultados parciais obtidos. Em seguida, mostramos algumas técnicas e algoritmos que visam a economia de energia existentes na literatura. Então, apresentamos o PWAVE, uma nova solução para reduzir o consumo de energia em RSSFs com coleta periódica. Avaliamos seu desempenho e mostramos seus resultados.

A principal contribuição deste trabalho foi mostrar que a coleta periódica é uma técnica que pode ser aplicada com o objetivo de minimizar o consumo de energia em uma RSSF, utilizada para aplicações em ambientes de florestas densas, em que não precisamos dos dados coletados continuamente e, principalmente em partes do dia onde a comunicação fica prejudicada devido as temperaturas elevadas e umidades extremas. O algoritmo trata do problema de sincronização dos relógios, com uma proposta inovadora, que sincroniza os nós de maneira descentralizada dispensando a necessidade de um nós raiz. Diminui o problema da colisão de dados, devido seu `Tslot` ser dividido em transmissão e recepção, distribuído em tempos diferentes para cada altura da árvore de roteamento, que atua em conjunto com as janelas de contenções de dados, resultando em uma menor probabilidade de colisão dos dados.

Resultados obtidos mostram que o PWAVE é um algoritmo candidato a solução de RSSFs baseadas em árvores e coleta periódica, contribuindo para modelagem e desenvolvimento dessas aplicações.

6.1 Aplicabilidade da Solução Proposta

O PWAVE pode ser utilizado como solução de coleta periódica em redes de sensores e servir como base para outras aplicações com características semelhantes. Como por exemplo, em cenários de ambientes de florestas, os quais não necessitam de emergência na entrega dos dados, como controle da fauna, previsões meteorológicas e também para monitoração de umidade de plantações.

Em contrapartida, devido aos longos períodos que os rádios ficam desligados, o atraso na comunicação existe, ou seja, a latência aumenta. Portanto o protocolo não deve ser utilizado em aplicações que necessitam de baixa latência, em aplicações de coleta em tempo real ou que necessitem de informações críticas (emergências e controle de queimadas).

6.2 Trabalhos Futuros

Como trabalhos futuros, pretende-se trabalhar as limitações do tamanho máximo da árvore de roteamento de dados, com diversas ondas, sem a necessidade da primeira onda de dados ter terminado para que a próxima comece a comunicação entre as alturas da árvore.

Um outro trabalho a ser realizado é uma técnica adaptativa para utilizar a coleta de dados forçada, que ocorre fora do agendamento programado pela coleta periódica, quando por exemplo, um nó da rede está prestes a estourar sua memória de armazenamento, este deve transmitir seus dados imediatamente. Esta transmissão deve levar em consideração o momento do dia mais adequado (condições climáticas), para realizá-la sem que sofra muita interferência do meio. Outro ponto importante a se trabalhar é tratar da perda de pacotes, já que o PWAVE não retransmite o pacote perdido.

Referências Bibliográficas

- Akyildiz, I.; W. Su, Y. S. & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114.
- Anastasi, G.; Conti, M.; Francesco, M. D. & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537 – 568.
- Bonnet, P.; Gehrke, J. & Seshadri, P. (2001). Towards sensor database systems. In *Second Int Conf. Mobile Data Management (MDM 2001)*, pp. 3–14.
- Boukerche, A.; Cheng, X. & Linus, J. (2003). Energy-aware data-centric routing in microsensor networks. In *In Proceedings of the 6th international workshop on Modeling analysis and simulation of wireless and mobile systems*, pp. 42–49. ACM Press.
- Boukerche, A.; de Oliveira, H. A. B. F.; Nakamura, E. F. & Loureiro, A. A. F. (2008). A novel location-free greedy forward algorithm for wireless sensor networks. In *ICC 2008*, pp. 2096–2101.
- Callaway, E. H. J. (2004). Wireless sensor networks architectures and protocols. In *Auerbach Publications*.
- Campus, E., editor (2003). *Redes de Computadores*.
- Cavalcanti, G. & Brandão, L. (2005). Roteamento de fontes dinâmicas em redes ad-hoc. Master's thesis, FACID.
- Dam, T. V. & Langendoen, K. (2003). An adaptive energy- efficient mac protocol for wireless sensor networks. In *Proceedings of the first international conference on embedded networked sensor systems.*, pp. 171–180. ACM Press, ACM.
- Elson, J.; Girod, L. & Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. In *Proc. OSDI '02*.

- Figueiredo, C. M. S.; Nakamura, E. F.; Ribas, A. D.; de Souza, T. R. B. & Barreto, R. S. (2009). Assessing the communication performance of wireless sensor networks in rainforests. In *Proceedings of the 2nd IFIP conference on Wireless days, WD'09*, pp. 226–231, Piscataway, NJ, USA. IEEE Press.
- Ganeriwal, S.; Capkun, S.; Han, C.-C. & Srivastava, M. B. (2005). Secure time synchronization service for sensor networks. In Jakobsson, M. & Poovendran, R., editores, *Workshop on wireless security*, pp. 97–106. ACM.
- Ganeriwal, S.; Kumar, R. & Srivastava, M. B. (2003). Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems, SenSys '03*, pp. 138-149, New York, NY, USA. ACM.
- Haque, A.; Murshed, M. & Ali, M. (2010). Efficient contention resolution in mac protocol for periodic data collection in wsns. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, IWCMC 2010*, pp. 437-441, New York, NY, USA. ACM.
- Hill, J.; M.Horton; R.Kling & Krishnamurthy, L. (2004). The platforms enabling wireless sensor networks. *ACM Communication*, 47(6):41–46.
- Israr, N. & Awan, I. (2006). Multihop routing algorithm for inter cluster head communication. In *Performance Engineering Workshop Bournemouth*, pp. 24–31. 22nd UK Performance Engineering Workshop Bournemouth.
- Johnson, D. B.; Maltz, D. A. & Broch, J. (2001). *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks Ad Hoc Networking*, chapter 5, pp. 139–172. Addison-Wesley, Charles E. Perkins edição.
- Kunz, T. (2003). Multicasting in mobile ad-hoc networks: achieving high packet delivery ratios. In *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research, CASCON 03*, pp. 156–170. IBM Press.
- Luz, G. D. (2004). Roteamento em redes de sensores. Master's thesis, USP.
- Macedo, D. F.; Correia, L. H. A.; dos Santos, A. L.; Loureiro, A. A. & e Nogueira., J. M. (2005). A pro-active routing protocol for continuous data dissemination wireless sensor networks. In *In 10th IEEE Symposium on Computer and Communications (ISCC)*,.

- Maroti, M.; Kusy, B.; Simon, G. & Ledeczi, A. (2004). The flooding time synchronization protocol. In *In SenSys04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. ACM Press.
- Nakamura, E. F.; Figueiredo, C. M. S. & Loureiro, A. A. (2004). Disseminação de dados adaptativa em redes de sensores sem fio auto-organizáveis. In *RSSF*.
- Ns-2 (2011). *Ns-2. The network simulator - ns-2*. <http://www.isu.edu./nslam/ns>. Online Available.
- Perkins, C. E. & Royer, E. M. (1999). Ad hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing System and Applications*, pp. 90–100, New Orleans, LA.
- Polastre, J.; Hill, J. & Culler., D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 95–107, New York, NY, USA. ACM, SenSys 04.
- Raghavendra, C. S.; Sivalingam, K. M. & Znati, T., editores (2004). *Wireless sensor networks*. Kluwer Academic Publishers, Norwell, MA, USA.
- Rappaport, T. (2001). *Wireless Communications Principles and Practice*. Prentice Hall, 2nd edition edição.
- Redin, J. C. (2004). Avaliação do impacto da implementação de segurança em protocolos de roteamento em redes ad-hod. Master's thesis, UFSC.
- Rohde, G. F. (2009). R-mac: Protocolo de acesso ao meio para consumo eficiente de energia em redes de sensores sem fio. Master's thesis, PUCRS.
- Tjoa, R.; Chee, K.; Sivaprasad, P.; Rao, S. & Lim., J. (2004). Clock drift reduction for relative time slot tdma-based sensor networks. In 2004, P., editor, *Personal, Indoor and Mobile Radio Communications*, volume 2, pp. 1042–1047. 15th IEEE International Symposium.
- Trigoni, N.; Yao, Y.; Demers, A.; Gehrke, J. & Rajaraman, R. (2007). Wave scheduling and routing in sensor networks. In *Article 2*, volume 3.
- Vahabi, M.; Rasid, M. F. A.; Abdullah, R. S. A. R. & Ghazvini, M. H. F. (2008). Adaptive data collection algorithm for wireless sensor networks. In *International Journal*, volume 8. Computer Science and Network Security.

- Vig, J. R. (1992). Introduction to quartz frequency standards. *IEEE Ultrasonics, Ferroelectrics, and Frequency Control Society*, 1:14–23.
- Werner-Allen, G.; Tewari, G.; Patel, A.; Welsh, M. & Nagpal, R. (2005). Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the 3rd international conference on Embedded networked sensor systems, SenSys '05*, pp. 142–153, New York, NY, USA. ACM.
- Xu, Y.; Heidemann, J. & Estrin, D. (2001). Geography-informed energy conservation for ad-hoc routing. In *7th Annual ACM/IEEE International Conference on Mobile Computing and Networking*.
- Yang, O. & Niu, D. (2009). Synchronous sleep and wake in ip-enabled wireless sensor networks. In *IEEE*.
- Ye, W.; Heidemann, J. & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12:493–506.
- Yupho, D. & Kabara, J. (2007). The effect of physical topology on wireless sensor network lifetime. *Journal of Networks*, 2(5):14–23.
- Zhao, F. & Guibas, L. (2004). Wireless sensor networks: An information processing approach. In 1.