

**Universidade Federal de São Carlos**

**Centro de Ciências Exatas e de Tecnologia**

**Departamento de Computação**

Programa de Pós-Graduação em Ciência da Computação

**Especificação e Implementação de um  
Protocolo Tolerante a Falhas e de Baixa  
Latência para Redes de Sensores Sem Fio**

ALUNO: RICHARD WERNER NELEM PAZZI  
ORIENTADORA: REGINA BORGES DE ARAÚJO  
CO-ORIENTADOR: AZZEDINE BOUKERCHE

SÃO CARLOS - SP  
DEZEMBRO - 2004

**Universidade Federal de São Carlos**

**Centro de Ciências Exatas e de Tecnologia**

**Departamento de Computação**

Programa de Pós-Graduação em Ciência da Computação

**Especificação e Implementação de um  
Protocolo Tolerante a Falhas e de Baixa  
Latência para Redes de Sensores Sem Fio**

**Richard Werner Nelem Pazzi**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Departamento de Computação, da Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Processamento de Imagens e Sinais - PIS.

SÃO CARLOS - SP

DEZEMBRO – 2004

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

P348ei

Pazzi, Richard Werner Nelem.

Especificação e implementação de um protocolo tolerante a falhas e de baixa latência para redes de sensores sem fio / Richard Werner Nelem Pazzi. -- São Carlos : UFSCar, 2004.

88 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2004.

1. Rede de computação - protocolos. 2. Sistemas de comunicação sem fio. 3. Sensor. I. Título.

CDD: 004.62 (20<sup>a</sup>)

*“My son, ask for thyself another kingdom, for that which I leave is too small for thee.”*

(King Philip II of Macedonia – 339 B.C.)

Dedico este trabalho:

À minha querida mãe, Maria Isabel, pelo incentivo e dedicação, sempre acreditando no meu sucesso.

Ao meu pai, Ronald, certamente orgulhoso por mais um importante passo na minha vida.

À minha irmã, Danielle, que sempre torceu para que tudo desse certo.

À Stela, minha namorada, pela paciência nos momentos mais difíceis.

# Agradecimentos

Primeiro, gostaria de agradecer à minha orientadora, prof<sup>a</sup> Dr<sup>a</sup> Regina Borges de Araújo, pela motivação, encorajamento, e direção durante os meus quase três anos de estudos na pós-graduação. Seu entusiasmo ilimitado pela pesquisa me fez trabalhar com muita dedicação e prazer. Gostaria de agradecer também ao prof. Dr. Azzedine Boukerche pelo seus conselhos e pelos artigos que publicamos.

À minha família que sempre me apoiou.

À minha namorada, Stela Marina Martins Braz, pela paciência durante os inúmeros finais de semana os quais passei a frente do computador.

A todos meus colegas do *LRVNet*, Goiano, Taciana, Gislaine, Marcelo, Fernando, Diego, Altieres, Nestor e Fernando, pelas discussões que contribuíram para a realização de nossos trabalhos. Agradecimentos especiais aos alunos de iniciação científica, Leonardo, Thadeu e Erlon, pela competência e principalmente pelo companheirismo.

Aos professores do Departamento de Computação da UFSCar, principalmente aos que tive a oportunidade de conhecer durante meus quase sete anos de estudos.

Às funcionárias do DC, Cristina e Mirian, pelos favores concedidos.

## Resumo

As redes de sensores sem fio estão cada vez mais sendo aplicadas para o monitoramento preciso de ambientes físicos sujeitos a situações críticas como incêndios, vazamento de gases tóxicos e explosões. Um grande desafio para essas redes é oferecer um canal rápido, confiável e tolerante a falhas para a difusão dos dados coletados, que atenda aos requisitos dos diferentes tipos de monitoramento, como periódico, baseado em consultas e dirigido a eventos, mesmo na presença de condições de emergência que podem causar falhas nos nós sensores e interrupção de caminhos de entrega de dados. Este trabalho apresenta um protocolo de roteamento de baixa latência e tolerante a falhas que atende os requisitos das aplicações de redes de sensores, especialmente as aplicações de monitoramento de condições críticas. O algoritmo de roteamento utiliza o mecanismo *publish/subscribe* e o conceito de entrega dirigida a eventos, uma técnica que seleciona o caminho mais curto para a notificação de eventos, reduzindo a latência. A tolerância a falhas é acionada localmente por nós que detectam alguma falha em um nó vizinho. O protocolo foi implementado e simulado no simulador de redes ns-2 e comparado ao conhecido paradigma Directed Diffusion. Métricas importantes, como atraso fim-a-fim, consumo de energia e taxa de sucesso no recebimento de pacotes, foram avaliadas demonstrando que o protocolo proposto pode ser uma solução apropriada para atender aos rigorosos requisitos de propagação de informações em aplicações de monitoramento em tempo real. É apresentada também uma extensão do protocolo proposto. Essa extensão está em fase de implementação e simulação e espera-se que reduza o tráfego de dados e, conseqüentemente, o consumo de energia. Essa nova solução segue uma abordagem hierárquica para a propagação de dados na rede.

# Abstract

Wireless sensor networks are increasingly being deployed for fine-grain monitoring of physical environments subjected to critical conditions such as fire, leaking of toxic gases and explosions. A great challenge to these networks is to provide a fast, reliable and fault tolerant channel for event propagation. The protocol has to meet the requirements of query-based, event-driven and periodic sensor network application scenarios, even in the presence of emergency conditions that can lead to node failures and path disruption to the sink that receives those events. This paper presents a fault tolerant and low latency routing protocol that meets the sensor networks requirements for critical condition monitoring applications. The algorithm uses the publish/subscribe mechanism and the concept of event driven delivery, a technique that selects the shortest path for the notification of events, reducing latency. Fault tolerance is achieved through the triggering of a local path repair mechanism to find new paths with minimal number of transmissions. The protocol was implemented and simulated in the ns-2 network simulator and compared to the Directed Diffusion paradigm. Relevant metrics, such as end-to-end delay, energy dissipation, and delivery ratio, were evaluated showing that the proposed algorithm can be an adequate solution to meet the strict requirements of information propagation in real-time monitoring applications. An extension of the proposed protocol is also presented. This extension is in phase of implementation and simulation and it is expected to reduce data traffic and, consequently, reduce the energy dissipation. This new solution follows a hierarchical approach to propagate data through the network.



# Lista de Figuras

---

2.1. Nós sensores espalhados em um campo de sensores .....	6
2.2. Consumo de energia dos módulos de um nó sensor .....	14
2.3. A pilha de protocolos de uma rede de sensores .....	18
2.4. Exemplos de projetos de nós sensores .....	25
3.1. O protocolo SPIN .....	29
3.2. As fases do Directed Diffusion .....	30
3.3. Plano de consulta em um nó líder .....	34
3.4. Encadeamento no PEGASIS .....	38
3.5. Agrupamento de dados em um esquema binário baseado em cadeia .....	39
3.6. Hierarquia e formação dos clusters no TEEN e APTEEN .....	40
3.7. Um típico <i>cluster</i> em uma rede de sensores .....	42
3.8. Exemplo de uma grade virtual no GAF .....	44
3.9. Encaminhamento geográfico recursivo .....	46
3.10. Modelo de filas em um nó particular .....	48
3.11. Componentes de roteamento do SPEED .....	49
4.1. Configuração inicial da rede .....	57
4.2. Estrutura de dados e algoritmo de configuração inicial da rede .....	58
4.3. (a) Envio da subscrição. (b) Entrega dos dados ao sink .....	61
4.4. Algoritmo de notificação de eventos .....	61
4.5. Envio dirigido de subscrições .....	62
4.6. Comparação do consumo de energia em um caminho .....	62
4.7. Mecanismo de reparo de caminhos do PEQ .....	64
4.8. Exemplo de cenário de simulação .....	65
4.9. Gráfico dos atrasos médios sink-fonte-sink .....	67
4.10. Gráfico do atraso médio do PEQ .....	67
4.11. Gráfico de comparação dos atrasos de DD e PEQ (30% de falhas) .....	68
4.12. Gráfico da taxa de sucesso de entrega de eventos .....	69
4.13. Gráfico de comparação das taxas de entrega do DD e PEQ (30% de falhas) .....	69
4.14. Gráfico da energia média dissipada por nó .....	69
4.15. Gráfico de comparação da energia dissipada (30% de falhas) .....	70
4.16. Gráfico do atraso médio com diferentes taxas de envio de pacotes .....	71
4.17. Gráfico da taxa de sucesso na entrega de pacotes .....	71
4.18. Seleção dos agregadores no HPEQ .....	73
4.19. Formação dos clusters .....	75
4.20. Comunicação <i>multihop</i> entre um agregador e <i>sink</i> .....	76

# Sumário

---

1	Introdução.....	1
1.1	Motivação e Objetivos do Trabalho .....	3
1.2	Organização da Dissertação.....	4
2	Redes de Sensores .....	6
2.1	Considerações Iniciais .....	6
2.2	Aplicações para Redes de Sensores.....	8
2.2.1.	Aplicações Militares .....	8
2.2.2.	Aplicações Ambientais .....	9
2.2.3.	Aplicações na Saúde.....	10
2.2.4.	Aplicações Domésticas.....	10
2.2.5.	Outras Aplicações Comerciais .....	11
2.3	Desafios no Projeto de Redes de Sensores .....	12
2.3.1.	Tolerância a Falhas .....	12
2.3.2.	Consumo de Energia.....	13
2.3.3.	Limitações de Hardware .....	14
2.3.4.	Expansibilidade .....	15
2.3.5.	Agregação de Dados.....	15
2.3.6.	Ambiente de Operação .....	16
2.3.7.	Topologia da Rede.....	17
2.3.8.	Meio de Transmissão.....	17
2.4	Arquitetura de Comunicação para Redes de Sensores .....	18
2.4.1.	A Camada de Aplicação .....	18
2.4.1.1.	Protocolo de Gerenciamento de Sensores .....	19
2.4.1.2.	Protocolo de Associação e Anúncio de Dados .....	19
2.4.1.3.	Protocolo para Consulta e Disseminação de Dados .....	20
2.4.2.	A Camada de Transporte .....	20
2.4.3.	A Camada de Rede .....	20
2.4.4.	Camada de Enlace .....	21
2.4.4.1.	Protocolos MAC para Redes de Sensores .....	22
2.4.5.	A Camada Física.....	23
2.5	Considerações Finais .....	25
3	Protocolos de Roteamento para RSSF.....	26
3.1	Considerações Iniciais .....	26
3.2	Protocolos Data-Centric .....	28
3.2.1.	Protocolo de Sensores para Informação via Negociação – SPIN.....	28
3.2.2.	Directed Diffusion – DD .....	29
3.2.3.	Energy-Aware Routing.....	31
3.2.4.	Rumor Routing .....	33
3.2.5.	COUGAR .....	34
3.2.6.	ACQUIRE .....	35
3.3	Protocolos Hierárquicos .....	36
3.3.1.	LEACH.....	37
3.3.2.	PEGASIS .....	38
3.3.3.	PEGASIS Hierárquico.....	38

3.3.4.	TEEN .....	40
3.3.5.	APTEEN .....	41
3.3.6.	Energy-Aware Routing for Cluster-Based Sensor Networks .....	41
3.4	Protocolos Baseados em Localização .....	43
3.4.1.	GAF .....	43
3.4.2.	GEAR .....	45
3.5	Protocolos Cientes de QoS .....	46
3.5.1.	SAR .....	47
3.5.2.	Energy-Aware QoS Routing Protocol .....	47
3.5.3.	SPEED .....	48
3.6	Considerações Finais .....	50
4	Projeto do PEQ, um Protocolo de roteamento para aplicações que requerem monitoramento periódico, orientado a eventos e baseado em consultas .....	53
4.1	Considerações Iniciais .....	53
4.2	Descrição do PEQ .....	55
4.2.1.	O Paradigma Publish/Subscribe para Redes de Sensores .....	56
4.2.2.	Configuração Inicial da Rede .....	57
4.2.3.	Propagação de Mensagem de Subscrição .....	59
4.2.4.	Envio da Mensagem de Notificação .....	60
4.2.5.	Envio Dirigido de Mensagens de Subscrição .....	61
4.2.6.	Reparo de Caminhos .....	63
4.3	Experimentos de Simulação .....	64
4.3.1.	Cenário de Simulação e Métricas .....	65
4.3.2.	Resultados Obtidos .....	66
4.4	HPEQ - Uma Solução Hierárquica para reduzir o tráfego na rede .....	72
4.4.1.	Configuração Inicial da Rede .....	73
4.4.2.	Seleção Aleatória dos Agregadores .....	74
4.4.3.	Formação dos <i>Clusters</i> .....	75
4.4.4.	Transmissão de Dados para os Agregadores .....	76
4.4.5.	Transmissão de Dados para o <i>Sink</i> .....	76
4.5	Considerações Finais .....	78
5	Conclusões .....	79
5.1	Considerações Iniciais .....	79
5.2	Contribuições Geradas .....	79
5.2.1.	Artigos Publicados .....	79
5.3	Trabalhos Futuros .....	80
5.4	Conclusões Finais .....	80
6	Referências Bibliográficas .....	82

# 1 Introdução

---

As recentes evoluções nos sistemas microeletrônicos e nas comunicações sem fio resultaram no desenvolvimento de dispositivos sensores pequenos em tamanho, de baixo custo e consumo. Esses dispositivos extremamente pequenos integram módulos de sensoriamento e processamento de dados e capacidade de comunicação. Examinando cada um desses dispositivos individualmente pode parecer que possuem pouca utilidade, porém a utilização efetiva de um grande número de tais dispositivos pode ser proveitoso na execução de tarefas de monitoramento em ambientes extensos. Uma grande densidade de sensores pode ser aplicada às áreas de interesse, como terrenos inacessíveis ou áreas de desastre, e usar métodos de auto-organização e colaboração para formar uma rede de sensores sem fio. Uma grande gama de aplicações está baseada na possibilidade de uso de vários tipos de sensores, como térmicos, sísmicos, acústicos, magnéticos, etc, para monitorar uma ampla variedade de condições, como temperatura, presença e movimento de pessoas e objetos, umidade, pressão, níveis de ruído, etc. Além disso, as redes de sensores podem ser usadas para o monitoramento contínuo ou periódico, para a detecção de eventos ou rastreamento de localização. Dessa forma, as redes de sensores possuem aplicações importantes em diversas áreas:

- Militar: monitoramento de tropas e equipamentos, supervisão do campo de batalha, detecção de ataques biológicos, químicos e nucleares;
- Ambiental: detecção de incêndio em florestas, detecção de inundações;
- Agricultura de precisão: monitoramento do nível de pesticidas ou condições do solo e do ar para auxiliar na irrigação;
- Saúde: monitoramento de dados fisiológicos humanos;
- Domiciliar: ambientes inteligentes, automação domiciliar.

Uma das aplicações mais apelativas é o monitoramento de condições de emergência, auxiliando na prevenção, combate e avaliação de situações de emergência com o objetivo de minimizar situações de perigo (como incêndio, vazamento de substâncias tóxicas, inundação, explosão) que colocam em risco vidas humanas e perda de patrimônio. Nós sensores são espalhados pelo ambiente de interesse para capturar as informações relevantes e auxiliar, em tempo real, equipes de resgate na tomada de decisões durante o salvamento e combate e, posteriormente, em situações de perícia e avaliação para apuração dos fatos e causas do acidente. Em prisões, grandes complexos industriais, aeroportos e aeronaves, áreas de desastre, entre outros, é importante manter um monitoramento confiável do ambiente físico, especialmente quando surgem situações de emergência. Por exemplo, rebeliões de prisioneiros podem levar a incêndios e perda de vidas e patrimônio. Em tais situações, é importante que a informação possa ser “capturada” do ambiente físico durante um estado de emergência, pois as informações podem ser usadas pelo grupo de segurança ou de resgate para um melhor gerenciamento de operações e decisões estratégicas. Redes de sensores que monitoram ambientes físicos sujeitos a situações de emergência, em que eventos, como presença de fumaça, têm que ser entregues assim que forem detectados (baixa latência), independente de falhas nos nós (tolerância a falhas); solicitações feitas à rede têm que ser atendidas em tempo-real, como o rastreamento de pessoas e/ou objetos em um prédio em chamas; ou ainda eventos podem ter que ser enviados de forma periódica.

Entretanto, a realização eficiente dessas e outras aplicações em ambientes não convencionais, altamente dinâmicos e complexos, é uma tarefa desafiadora tanto para soluções técnicas quanto para algorítmicas. Algumas características como o enorme número de dispositivos sensores envolvidos; as restrições de energia, processamento e memória; e as eventuais falhas nos dispositivos sensores – criam novos aspectos de projeto e implementação que se diferenciam das abordagens de sistemas distribuídos e das técnicas de redes *ad hoc* atuais [BOU 04].

Para manter o fluxo de informações dos nós sensores durante uma situação de emergência, requisitos não funcionais como confiabilidade, segurança, causalidade etc., devem ser atendidos, pois uma solução de rede de sensores sem fio precisa tratar eventuais falhas nos nós sensores, que podem ocorrer devido a sensores queimados, comunicação

prejudicada por interferências, esgotamento de energia, ou até mesmo água ou fumaça densa podem prejudicar a comunicação entre nós sensores. Além de tolerante a falhas, uma rede de sensores para aplicações de monitoramento de condições críticas precisa ser confiável, oferecer baixa latência e economizar energia.

Este trabalho apresenta um novo protocolo de roteamento para redes de sensores sem fio que atende, simultaneamente, os três tipos de cenários de aplicação: periódico, dirigido a eventos, e baseado em consulta. O protocolo provê baixa latência para a notificação de eventos, rápida reconfiguração de caminhos falhos, e alta confiabilidade na entrega de pacotes com baixa dissipação de energia. A baixa latência é alcançada pelo uso do menor caminho para a entrega de pacotes. A utilização do paradigma “*publish/subscribe*” como modelo de comunicação entre nós sensores e centro de controle permite um menor tráfego de pacotes na rede. Tolerância a falhas é ativada quando um nó detecta uma falha na entrega de pacotes, que aciona um mecanismo local de reparo de caminhos. Para validar e avaliar seu desempenho, o protocolo foi implementado e simulado em um conhecido simulador de redes, o ns-2 [NS2 04], e os resultados foram comparados com um típico paradigma de propagação de dados em redes de sensores, o Directed Diffusion [INT 00][INT04].

## 1.1 Motivação e Objetivos do Trabalho

Tipicamente as atuais soluções tratam de aplicações ou de monitoramento passivo (periódico), ou orientado a eventos (como aplicação de segurança) ou de rastreamento. Uma solução para aplicações de monitoramento de ambientes sujeitos a condições de emergência pode tratar os três tipos de cenário de aplicação (periódico, baseado em consulta, e dirigido a eventos).

Em uma aplicação de redes de sensores em que a tarefa da rede de sensores é monitorar uma área sujeita a condições de emergência, é desejado que qualquer evento na região monitorada seja detectado e transmitido para um dos centros de controle. Em aplicações de tempo-real para redes de sensores, confiabilidade e latência, respectivamente a taxa de sucesso de pacotes recebidos e o tempo que a rede leva para a rede informar o centro de controle, são cruciais.

A motivação principal para o trabalho aqui apresentado é guiada pela necessidade do atendimento simultâneo de todos os seguintes requisitos: baixa latência, confiabilidade, expansibilidade, reconstrução rápida de caminhos na presença de falhas e economia de energia. Embora diversas soluções de roteamento em redes de sensores tenham sido reportadas na literatura, o atendimento a todos os requisitos ao mesmo tempo é uma tarefa difícil, pois os requisitos geralmente são conflitantes.

Este trabalho focaliza uma solução aos problemas relacionados à propagação dos dados capturados pelos nós em uma rede de sensores. Os dados precisam ser encaminhados dos nós sensores a um centro de controle, chamado na literatura de *sink*. Como resultado deste trabalho, o protocolo PEQ (Periodic, Event-driven, Query-based) foi especificado e implementado, que provê roteamento rápido e confiável para redes de sensores sem fio que pode ser empregado em aplicações de monitoramento periódico, baseado em eventos, ou baseado em consultas. A idéia básica do PEQ é utilizar nós sensores de baixo custo, realizando processamentos simples em cada nó, formando uma rede de nós homogêneos. O protocolo PEQ foi ainda estendido para prover agrupamento de dados de forma a reduzir o tráfego na rede e diminuir a latência, dando origem ao protocolo HPEQ (Hierarchical PEQ) que foi especificado e encontra-se em fase de implementação e simulação.

Apesar da motivação do trabalho ser aplicações de monitoramento de condições de emergência, a solução pode ser aplicada em diversas áreas, que variam desde saúde (monitoramento de sinais vitais do corpo, instrumentação médica, localização de objetos e pessoas nas dependências de um hospital, etc), transportes (controle de tráfego, rastreamento de veículos, etc), militar (monitoramento do campo de batalha, apoio à tomada de decisões estratégicas em tempo real, etc), resgate e combate (monitoramento de situações críticas, como incêndios, prisões, etc), entre outras.

## 1.2 Organização da Dissertação

Este trabalho está organizado da seguinte forma: o capítulo 2 descreve as redes de sensores, suas aplicações e seus principais desafios. O capítulo 3 apresenta um estudo específico sobre as soluções para a camada de rede da pilha de protocolos para redes de sensores. O capítulo 4 apresenta o projeto do protocolo PEQ, que inclui sua especificação,

implementação, simulação e avaliação de desempenho, além de sua extensão, o protocolo HPEQ e sua especificação. Finalmente, o capítulo 5 apresenta as conclusões deste trabalho e propostas futuras, seguido de Referências Bibliográficas.



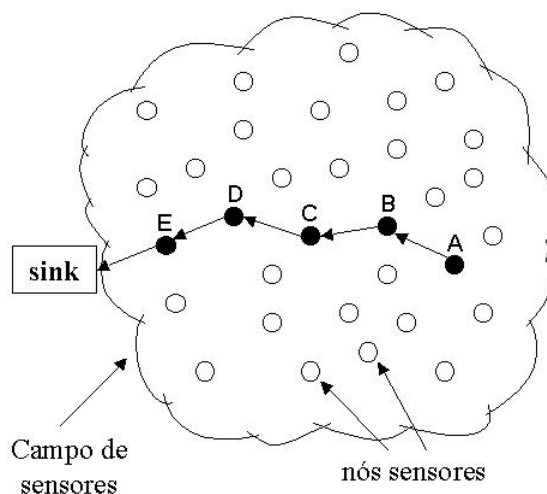
## 2 Redes de Sensores

---

### 2.1 Considerações Iniciais

Os avanços recentes nas tecnologias de comunicação, sensoriamento e processamento possibilitaram o desenvolvimento de nós sensores multifuncionais de baixo custo e consumo, que são pequenos em tamanho e que se comunicam a curtas distâncias. Constituídos de módulos de sensoriamento, comunicação e processamento de dados, esses nós sensores intensificaram a idéia de redes de sensores baseadas em esforço colaborativo de um grande número de nós.

Uma rede de sensores é composta por um grande número de nós sensores, os quais são densamente instalados na área que se quer monitorar, como mostrado na Figura 2.1. A posição dos nós não precisa ser necessariamente predeterminada, o que possibilita uma disposição aleatória em locais de difícil acesso, como em áreas de desastres e incêndios. Por um outro lado, isto significa que os algoritmos e protocolos para redes de sensores devem possuir a característica de auto-organização dos nós. Outra característica única das redes de sensores é a colaboração entre os nós para tratar os dados e encaminhá-los. Ao invés de enviar o dado “bruto” para outros nós, um nó pode usar seu potencial de processamento para realizar alguma operação e transmitir o dado parcialmente processado, ou somente quando necessário.



**Figura 2.1.** Nós sensores espalhados em um campo de sensores.

Muitas aplicações para redes de sensores fazem uso dessas características citadas acima. Algumas das áreas de aplicação incluem monitoramento, segurança, saúde, e militar. Por exemplo, uma área de risco, como um presídio, pode ser monitorada remotamente para evitar incidentes. As redes de sensores também podem ser utilizadas para detectar agentes químicos no ar e na água, ajudando a identificação, concentração, e localização dos poluentes.

A realização dessas e outras aplicações para redes de sensores requer técnicas de redes *ad hoc* sem fio. Embora existam vários protocolos e algoritmos para redes sem fio tradicionais, eles não atendem bem às características das redes de sensores e aos requisitos das aplicações. Apenas para ilustrar esse ponto, as diferenças entre redes de sensores e redes *ad hoc* [PER 00] são citadas abaixo:

- O número de nós em uma rede de sensores geralmente é muito maior que o número de nós em uma rede *ad hoc*;
- Os nós sensores são aplicados densamente em uma área;
- Os nós sensores estão mais sujeitos a falhas;
- A topologia de uma rede de sensores muda freqüentemente;
- Os nós sensores usam principalmente o paradigma de comunicação *broadcast*, enquanto que a maioria das redes *ad hoc* baseiam-se na comunicação ponto-a-ponto;
- Os nós sensores são limitados quanto a energia, processamento, e memória.

Uma das mais importantes restrições nos nós sensores é o que se refere à energia. Os nós sensores possuem fontes de energia limitadas, normalmente difíceis de serem substituídas. Conseqüentemente, enquanto redes tradicionais focalizam seus esforços em oferecer qualidade de serviço (QoS), as redes de sensores precisam primeiro resolver o problema da economia de energia.

Este capítulo apresenta uma investigação das aplicações que mostram a utilidade das redes de sensores, e divulga os desafios que influenciam o projeto de uma rede de sensores.

## 2.2 Aplicações para Redes de Sensores

As redes de sensores podem ser compostas por vários tipos de sensores diferentes como térmico, sísmico, visual, infravermelho, acústico, que são capazes de monitorar uma ampla variedade de condições ambientais tais como: temperatura, umidade, luminosidade, pressão, nível de ruído, etc.

O conceito de micro-sensoriamento e a conexão sem fio dos nós prenunciam muitas novas áreas de aplicação. Algumas das áreas de aplicação podem ser divididas em militar, ambiental, saúde, doméstica, exploração espacial, processamento químico, entre outras.

### 2.2.1 Aplicações Militares

As características de uma rede de sensor sem fio fazem dela uma técnica promissora para aplicações militares. Como essas redes são baseadas em grandes densidades de nós sensores de baixo custo, a destruição de alguns desses nós por ações hostis não deve afetar uma operação militar tão quanto a destruição de um sensor tradicional, que faz do conceito de redes de sensores uma melhor abordagem para campos de batalhas. Algumas das aplicações militares para redes de sensores são: monitoramento dos soldados, equipamentos e munição; supervisão do campo de batalha; e detecção de ataques nucleares, biológicos e químicos.

- *Monitoramento dos soldados, equipamentos e munição:* Os comandantes podem constantemente monitorar o estado de suas tropas, a condição e a disponibilidade de equipamentos e munição em um campo de batalha com o uso de redes de sensores. Cada tropa, veículo, equipamento e munição crítica pode carregar pequenos sensores que reportam seu estado. Essas informações são coletadas por estações base (*sinks*) e enviadas para os comandantes.
- *Supervisão do campo de batalha:* Territórios críticos, rotas de aproximação, caminhos podem ser rapidamente cobertos por redes de sensores e observados para detecção e localização de forças inimigas.
- *Detecção de ataques nucleares, biológicos e químicos:* Redes de sensores aplicadas a uma região podem ser usadas como um sistema de alerta sobre agentes químicos e

biológicos, fornecendo às tropas tempo hábil para reação, diminuindo as baixas drasticamente. As redes de sensores também podem ser úteis para um reconhecimento detalhado de um ataque nuclear, biológico ou químico após sua detecção. Por exemplo, uma área atingida por um ataque nuclear pode ser estudada sem a necessidade de expor a tropa de reconhecimento à radiação nuclear.

### 2.2.2 Aplicações Ambientais

Algumas aplicações ambientais para redes de sensores incluem: agricultura de precisão; detecção de incêndio em florestas; detecção de enchentes; rastreamento de animais; monitoramento de condições ambientais que afetem plantações; detecção química e biológica; monitoramento de mares e oceanos, solos e da atmosfera; pesquisas meteorológicas ou geofísicas; e estudos sobre poluição [BUL 01][CER 01][HAL 01][KAH 99][WAR 01][WIT 01].

- *Agricultura de precisão*: Alguns dos benefícios são: a habilidade de monitorar o nível de pesticida na água; e monitorar condições do ar e do solo em tempo real. Essas informações podem ser usadas, por exemplo, para aplicar a irrigação de forma mais eficiente, evitando desperdícios.
- *Detecção de incêndio em florestas*: Os nós sensores podem ser espalhados estratégica ou aleatoriamente em uma floresta, relatando aos usuários a origem exata do fogo antes que se espalhe. Esses nós devem ser aplicados em grande número e precisam estar equipados com algum tipo de fonte de energia alternativa [CHA 99], como células solares, devido ao fato de que os sensores devem funcionar por meses e até anos, pois a troca de suas baterias pode ser inviável. Os nós vão colaborar uns com os outros para realizar o monitoramento distribuído e para superar obstáculos, como árvores e rochas, que bloqueiem a comunicação entre nós.
- *Detecção de enchentes*: Um exemplo de detecção de enchentes é o sistema ALERT [ALE 01] desenvolvido nos EUA. Diversos tipos de sensores utilizados no sistema ALERT são detectores de chuva, nível de água e condição climática. Esses sensores fornecem informação para um sistema de banco de dados centralizado, que toma a devida decisão.

### 2.2.3 Aplicações na Saúde

Algumas das aplicações de redes de sensores na área de saúde são: monitoramento integrado de pacientes; diagnósticos; administração de drogas em hospitais; monitoramento de dados fisiológicos humanos; e rastreamento de doutores e pacientes em um hospital [BUL 01][KAH 99][NOU 00][RAB 00][WAR 01][AKY 02].

- *Administração de drogas em hospitais:* Se nós sensores puderem ser instalados em medicamentos, a chance de prescrever uma medicação errada aos pacientes pode ser minimizada. Pacientes podem carregar nós sensores que identificam suas alergias e medicamentos necessários. Sistemas computadorizados como descrito em [SIB 01] têm mostrado que podem ajudar a minimizar os eventos indesejáveis relacionados aos medicamentos.
- *Monitoramento de dados fisiológicos humanos:* Os dados fisiológicos coletados pela rede de sensores podem ser armazenados por um longo período, e podem ser usados para investigação médica [OGA 98]. A rede de sensores instalada pode também monitorar e detectar o comportamento de pessoas idosas. Esses pequenos sensores permitem à pessoa uma maior liberdade de movimento e permite a doutores identificar sintomas mais cedo [AKY 02]. Eles também oferecem uma melhor qualidade de vida aos pacientes comparada aos centros de tratamento [BAU 00]. O projeto “*Health Smart Home*”, em desenvolvimento na Faculdade de Medicina em Grenoble – França, visa validar a viabilidade de tal sistema [NOU 00].
- *Rastreamento de médicos e pacientes em um hospital:* Cada paciente possui nós sensores leves e pequenos acoplados a eles. Cada nó tem sua tarefa específica. Por exemplo, um nó sensor detecta o ritmo cardíaco enquanto um outro detecta a pressão sanguínea. Médicos podem também carregar um nó sensor, que permite a outros doutores localizar seus colegas de trabalho em um hospital [AKY 02].

### 2.2.4 Aplicações Domésticas

- *Automação em casas:* Com o avanço tecnológico, nós sensores inteligentes e atuadores podem ser acoplados a aparelhos, tais como aspiradores de pó, fornos de

microondas, refrigeradores, videocassetes, etc [PET 01]. Esses nós sensores embutidos nos aparelhos domésticos podem interagir uns com os outros e com redes externas via Internet ou satélite. Eles permitem aos usuários gerenciar local e remotamente seus aparatos domésticos com maior facilidade.

- *Ambiente Inteligente*: O desenvolvimento de ambientes inteligentes pode ter duas perspectivas diferentes: centrado no ser humano e centrado na tecnologia [ABO 00]. Para o centrado no humano, um ambiente inteligente tem que se adaptar às necessidades de seus usuários em termos de entrada e saída. Para o centrado na tecnologia, novas tecnologias de hardware, soluções em rede, e serviços de *middleware* precisam ser desenvolvidos. Um cenário que mostra como nós sensores podem ser usados para criar um ambiente inteligente é explorado em [HER 00]. Os nós sensores podem ser embutidos em móveis e aparelhos domésticos, e podem comunicar-se uns com os outros e com o servidor da sala. O servidor da sala também pode se comunicar com outros servidores de outras salas para aprender sobre os serviços que eles oferecem, por exemplo, impressão, fax, sistema de som, etc. Outro exemplo de ambiente inteligente é o “*Residential Laboratory – Aware Home*” do Instituto de Tecnologia da Geórgia [ESS 00]. As tarefas de computação e monitoramento nesse ambiente precisam ser confiáveis, persistentes e transparentes.

### 2.2.5 Outras Aplicações Comerciais

Algumas das aplicações comerciais incluem: monitoramento de desgaste de material; monitoramento de qualidade de produtos; escritório inteligente; controle de robôs em fábricas; brinquedos interativos; museus interativos, monitoramento de áreas de desastre; detecção e monitoramento de furtos de carros; e rastreamento de veículos [RAB 00][POT 00][BUL 01][WAR 01].

- *Escritório inteligente*: O condicionamento de ar da maioria dos prédios de escritórios é controlado centralmente. Conseqüentemente, a temperatura dentro de uma sala pode variar em alguns graus; um lado da sala pode estar mais quente do que o outro devido ao único controle presente na sala, e o fluxo de ar do sistema central normalmente não é distribuído. Um sistema distribuído de redes de sensores

sem fio pode ser instalado para controlar o fluxo de ar e a temperatura em diferentes partes da sala. Estima-se que tal tecnologia distribuída pode reduzir o consumo de energia nos EUA em dois quadrilhões de unidades térmicas (BTU's), que significa uma economia de 55 bilhões de dólares por ano e uma redução de 35 milhões de toneladas de emissão de carbono [RAB 00].

- *Museu interativo*: No futuro, crianças poderão ser capazes de interagir com objetos em museus e aprender mais sobre eles. Esses objetos serão capazes de responder ao seu toque e à fala das pessoas. As crianças poderão participar em tempo real em experimentos de causa e efeito, que podem ensiná-las sobre ciência. As redes de sensores sem fio também podem prover localização dentro do museu. Um exemplo de museu interativo é o *San Francisco Exploratorium*, que oferece uma combinação de dados e experimentos de causa e efeito [RAB 00].
- *Detecção e monitoramento de furtos de carros*: Nós sensores estão sendo utilizados para detectar e identificar incidentes relacionados a furto de veículos dentro de uma região geográfica. As informações são reportadas para usuários remotos pela Internet para análise [POT 00].

## 2.3 Desafios no Projeto de Redes de Sensores

O projeto de uma rede de sensores deve considerar e tentar superar vários desafios, que incluem: tolerância a falhas; baixo consumo de energia; limitações de hardware; expansibilidade; agregação de dados; ambiente de operação; topologia da rede; e meio de transmissão. Esses requisitos são, na maioria das vezes, atendidos separadamente nos trabalhos encontrados na literatura. Poucos são os estudos que possuem uma visão integrada de todos esses desafios, que servem como guia de projeto para redes de sensores.

### 2.3.1 Tolerância a Falhas

Os nós de uma rede de sensores estão sujeitos a falhas devido a alguns fatores como: falta de energia; danos físicos; ou interferência do ambiente. As falhas dos nós sensores não devem afetar a tarefa global da rede de sensores, que é a de fornecer informações sobre o ambiente monitorado. Tolerância a falhas, ou confiabilidade da rede, é

a habilidade de manter as funcionalidades da rede sem qualquer interrupção devido às falhas dos nós [HOB 00][SHE 01]. Protocolos e algoritmos devem ser implementados para alcançar os níveis de tolerância a falhas requisitados pelas redes de sensores. Se uma rede de sensores for utilizada em um ambiente onde há pouca interferência, o protocolo pode ser um pouco mais simples. Por exemplo, se os nós sensores forem utilizados em uma casa para monitorar o nível de umidade ou temperatura, o mecanismo de tolerância a falhas pode ser menos exigido, pois esse tipo de rede de sensores é dificilmente afetado pelos ruídos do ambiente ou por danos físicos. No entanto, se uma rede for aplicada para monitoramento e detecção em um campo de batalhas, há uma grande necessidade de um algoritmo mais robusto de tolerância a falhas. Portanto, o nível de tolerância a falhas depende da aplicação da rede de sensores [AKY 02].

### 2.3.2 Consumo de Energia

Um nó sensor normalmente é equipado com uma fonte de energia limitada. Em alguns cenários de aplicação, a substituição de fontes de energia pode ser inviável. Portanto, o tempo de vida de um nó é altamente dependente do tempo de vida de sua bateria. Em uma rede de sensores onde os nós atuam de forma colaborativa, cada nó tem o duplo papel de fonte e encaminhador de dados. O mau funcionamento de alguns nós pode causar mudanças na topologia e, conseqüentemente, a necessidade de reorganização de rotas para entrega de dados. A importância da conservação de energia induz os pesquisadores ao desenvolvimento de protocolos e algoritmos cientes de energia (*energy-aware*) para redes de sensores.

O consumo de energia não é o principal foco de atenção em algumas outras redes sem fio, pois as fontes de energia podem ser substituídas facilmente. Para essas redes a atenção e os esforços estão voltados para o fornecimento de qualidade de serviço (QoS), diferentemente das redes de sensores, onde a eficiência no consumo de energia é uma importante métrica de desempenho, influenciando diretamente no tempo de vida da rede [AKY 02].



A tarefa de um nó da rede pode ser dividida em três categorias: sensoriamento, processamento de dados, e comunicação. O módulo de sensoriamento de um nó é a unidade responsável por detectar eventos no ambiente e convertê-los para sinais digitais. O consumo de energia de um módulo de sensoriamento varia de acordo com a natureza da aplicação. A detecção esporádica de eventos consome menos energia que um monitoramento constante. Dentre os três módulos, o de comunicação é o que mais consome energia em um nó sensor, como mostrado na Figura 2.2 [BOU 03][EST 02]. O exemplo descrito em [POT 00] ilustra essa disparidade. Devido a essa disparidade, a conservação de energia através do desligamento dos módulos de comunicação dos nós domina as pesquisas na área de redes de sensores.

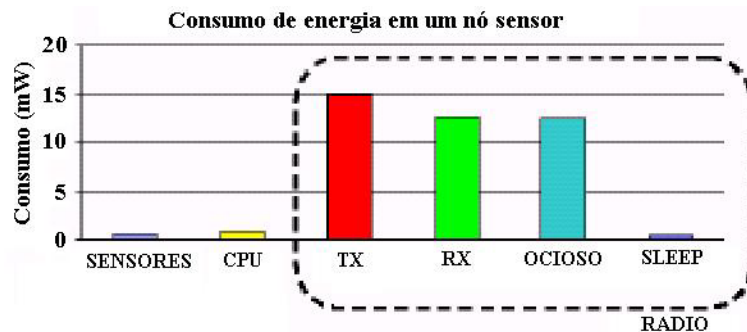


Figura 2.2. Consumo de energia dos módulos de um nó sensor.

### 2.3.3 Limitações de Hardware

Para garantir a viabilidade em algumas aplicações, muitos sensores precisam ser extremamente pequenos, às vezes menores que um centímetro cúbico [POT 00]. Além do tamanho, existem outras restrições para os nós sensores [KAH 99], que devem:

- Consumir pouca energia;
- Operar em altas densidades;
- Possuir baixo custo de produção e ser descartáveis;
- Ser autônomos;
- Ser adaptáveis ao ambiente.

Como os nós sensores normalmente são inacessíveis, o tempo de vida da rede depende do tempo de vida das fontes de energia dos nós. A energia é também um recurso escasso devido à limitação de tamanho [MIN 01]. É possível estender o tempo de vida da rede de sensores através da extração de energia do ambiente. Um exemplo é a célula solar, que transforma a energia térmica em energia elétrica.

Embora um maior poder computacional esteja sendo disponibilizado em processadores cada vez menores, os módulos de processamento e de memória dos nós sensores ainda são recursos escassos. Por exemplo, a unidade de processamento de um nó “*smart dust*” [POT 00] é um micro-controlador de 4 MHz com 8 KB de memória flash para instruções, 512 bytes de memória RAM e 512 bytes de EEPROM [PEA 01]. O TinyOS [TOS 04] é o sistema operacional utilizado neste processador, que possui 3500 bytes para o sistema operacional e 4500 bytes disponíveis para código.

#### 2.3.4 Expansibilidade

Dependendo da aplicação, uma rede de sensores pode possuir centenas ou até milhares de nós. Novos esquemas devem ser capazes de funcionar com estes números de nós. Devem também utilizar a alta densidade das redes de sensores. A densidade pode variar de poucos nós a centenas de nós em uma região. Essa densidade depende da aplicação na qual a rede de sensores está sendo usada. Para a aplicação de diagnose de máquinas, a densidade está em torno de 300 nós em uma região de  $5 \times 5 \text{ m}^2$ , e a densidade para a aplicação de rastreamento de veículos está em torno de 10 nós por região [SHI 01]. Para aplicações domésticas, o número de nós sensores varia de 25 a 100 por região [CER 01]. A densidade será extremamente alta quando uma pessoa carregando centenas de sensores, que estão embutidos em óculos, roupas, sapatos, relógio, jóias, e no corpo humano, estiver transitando dentro de sua casa ou estabelecimento que utiliza os serviços de uma rede de sensores.

#### 2.3.5 Agregação de Dados

Como os nós sensores normalmente podem gerar dados redundantes, pacotes similares de múltiplos nós podem ser agregados para que o número de transmissões seja reduzido. A *agregação de dados*, ou  *fusão*, é a combinação de dados de fontes diferentes

através do uso de funções como *supressão* (eliminação de pacotes duplicados), *mínimo*, *máximo*, *média*, etc [EST 99]. Algumas dessas funções podem ser utilizadas parcial ou integralmente em cada nó sensor, permitindo que os nós reduzam os dados na rede [ELS 01][EST 00]. Ao reconhecer que o processamento consome muito menos energia que a comunicação [CHO 00], surge a possibilidade de economizar energia através da agregação de dados. Essa técnica tem sido usada para alcançar a eficiência no consumo e a otimização de tráfego em alguns protocolos de roteamento [ELS 01][EST 00][GIR 01]. Em algumas arquiteturas de rede, todas as funções de agregação são atribuídas a nós especializados, com mais recursos de energia, processamento e memória [CER 01]. A agregação de dados também se faz viável através de técnicas de processamento de sinais. Neste caso, é referido como  *fusão de dados*  quando um nó é capaz de produzir um sinal mais preciso através da redução de ruídos e do uso de técnicas como o *beamforming* para combinar os sinais [CHO 00]

### 2.3.6 Ambiente de Operação

Os nós de uma rede de sensores normalmente têm que funcionar sob as mais diversas condições. A lista a seguir mostra alguns locais onde as redes de sensores podem ser utilizadas [AKY 02]. Os nós devem funcionar:

- No interior de grandes maquinários;
- No fundo de um oceano;
- Dentro de um ciclone;
- Na superfície de um oceano durante um tornado;
- Em um campo contaminado;
- Em um campo de batalha;
- Em uma casa ou grandes empresas ou fábricas;
- Acoplados a animais;
- Acoplados a veículos, etc.

Os nós trabalham sob alta pressão no fundo de um oceano, sob calor ou frio extremo como dentro de máquinas, e em ambientes com muito ruído, que interfere na comunicação dos sensores.

### 2.3.7 Topologia da Rede

Devido às freqüentes falhas que os nós estão sujeitos, o gerenciamento de topologia de uma rede de sensores é uma tarefa desafiante. Centenas a milhares de nós são utilizados em uma rede, que podem ser lançados ou colocados um a um no local que se quer monitorar. Por exemplo, os nós podem ser lançados de um avião ou colocados em uma fábrica por um humano ou robô. Mudanças na topologia da rede acontecem devido a diversos fatores, tais como: posição dos nós, energia disponível, e mau funcionamento.

### 2.3.8 Meio de Transmissão

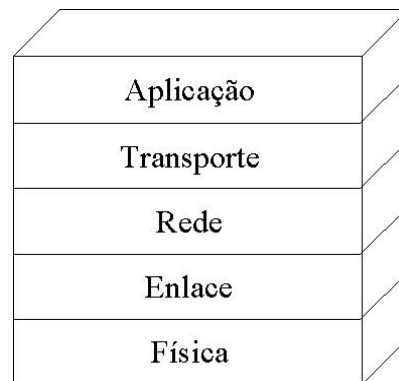
Em uma rede de sensores sem fio, os nós comunicam-se entre si através de rádio ou infravermelho. Uma opção para comunicação por rádio é o uso da banda industrial, científica e médica (ISM), que oferece licença livre na maioria dos países. De acordo com os autores de [POR 00], as limitações de hardware e o equilíbrio entre eficiência da antena e consumo de energia limitam a escolha de uma freqüência portadora para os pequenos e baratos módulos de comunicação dos nós de uma rede de sensores.

A maioria dos projetos de hardware para nós sensores é baseada em circuitos de rádio freqüência. O nó sensor  $\mu$ AMPS, descrito em [SHI 01], usa um transceptor Bluetooth de 2.4 GHz. O dispositivo sensor descrito em [WOO 01], usa um canal simples de rádio freqüência operando a 916 MHz. A arquitetura do WINS [POT 00] também utiliza rádio freqüência para comunicação.

Outro modo de comunicação entre nós em uma rede de sensores é por infravermelho, que é livre de licença e sofre pouca interferência de dispositivos eletrônicos. Os transceptores de infravermelho são baratos e fáceis de serem construídos. Porém, a grande desvantagem do infravermelho é a necessidade de uma linha de visão entre os pares comunicantes [AKY 02].

## 2.4 Arquitetura de Comunicação para Redes de Sensores

Os nós sensores geralmente são espalhados por um campo de sensores. Cada um desses nós possui a capacidade de coletar e encaminhar dados a uma estação base (*sink*). A pilha de protocolos usada pelos nós é mostrada na figura 2.3, e combina roteamento com economia de energia, comunica eficientemente através do meio sem fio, e promove esforços colaborativos entre os nós sensores. A pilha de protocolos pode consistir de: *camada de aplicação*, *camada de transporte*, *camada de rede*, *camada de enlace*, e *camada física*. Dependendo da tarefa de sensoriamento, diferentes tipos de aplicações podem ser implementados e usados na camada de aplicação. A camada de transporte ajuda a manter o fluxo de dados à necessidade da aplicação. A camada de rede cuida do roteamento dos dados. Como o ambiente é ruidoso e os nós sensores podem ser móveis, o protocolo MAC deve economizar energia e minimizar as colisões de pacotes. A camada física trata a necessidade de técnicas de modulação, transmissão e recepção.



**Figura 2.3.** A pilha de protocolos de uma rede de sensores.

### 2.4.1 A Camada de Aplicação

Embora muitas áreas de aplicação para redes de sensores foram propostas e definidas [AKY 02][BUL 01][CER 01][HAL 01][WAR 01][WIT 01][NOU 00][RAB 00], protocolos para a camada de aplicação ainda constituem uma vasta região inexplorada. Nesta seção serão discutidos três possíveis protocolos para a camada de aplicação: protocolo de gerenciamento de sensores (SMP), protocolo de associação e anúncio de dados (TADAP), e protocolo para consulta e disseminação de dados (SQDDP) [AKY 02].

#### 2.4.1.1 Protocolo de Gerenciamento de Sensores

Desenvolver um protocolo de gerenciamento para a camada de aplicação apresenta várias vantagens. As redes de sensores possuem muitas áreas de aplicação diferentes, e o acesso a elas através de outras redes como a Internet é objeto de estudo em alguns projetos recentes [POT 00]. Um protocolo de gerenciamento para a camada de aplicação faz com que o *hardware* e o *software* das camadas inferiores fiquem transparentes para as aplicações.

O SMP (*Sensor Management Protocol*) é um protocolo de gerenciamento que oferece as operações de software necessárias para a realização das seguintes tarefas administrativas:

- Introdução das regras relacionadas à agregação de dados e nomeação baseada em atributos;
- Sincronização dos nós sensores;
- Ligar e desligar os nós;
- Consultar a configuração da rede de sensores e o estado de cada nó;
- Reconfiguração da rede;
- Autenticação, distribuição de chaves e segurança na comunicação de dados.

Algumas dessas tarefas são descritas em [ELS 01][EST 99][PEA 01][SAV 01][SHE 01].

#### 2.4.1.2 Protocolo de Associação e Anúncio de Dados

Outra operação importante nas redes de sensores é a disseminação de consultas (interesses). Os usuários enviam seus interesses para a rede toda ou para um subconjunto de nós. O interesse pode ser sobre um certo atributo de um fenômeno ou um evento [INT 00]. Outra abordagem é o anúncio de dados disponíveis onde os nós sensores divulgam os dados para os usuários, que podem então consultar o dado de interesse [HEI 99]. Um protocolo da camada de aplicação que ofereça ao usuário uma interface eficiente para a disseminação de interesses é benéfico para as operações das camadas inferiores, como o roteamento.

### 2.4.1.3 Protocolo para Consulta e Disseminação de Dados

O SQDDP (*Sensor Query and Data Dissemination Protocol*) provê aplicações com interfaces para realizar consultas, responder a consultas e coletar as respostas dos sensores. As consultas normalmente não são direcionadas a nós em particular. Ao invés disso, esquemas de nomeação de dados baseados em atributo ou localização são utilizados. Por exemplo, consultas do tipo “quais locais apresentam temperatura acima de 60°C” são baseadas em atributo. Como exemplo, a (SCTL) (*Sensor Query and Tasking Language*) [SHE 01] é uma linguagem para consultas em sensores proposta como uma aplicação que fornece um conjunto de serviços para consulta em redes de sensores.

Diferentes tipos de protocolos para consulta podem ser desenvolvidos para várias aplicações. Porém, o uso desses protocolos pode ser único para cada aplicação.

## 2.4.2 A Camada de Transporte

A necessidade da camada de transporte é apontada na literatura em [POT 00][RAB 00]. Essa camada é necessária especialmente quando o sistema precisa ser acessado pela Internet ou outras redes externas. Entretanto, não há tentativa alguma até o momento de proposta de um esquema ou discussão das questões relacionadas à camada de transporte de uma rede de sensores na literatura [AKY 02]. Diferentemente de protocolos como TCP, os esquemas de comunicação fim-a-fim nas redes de sensores não são baseados em endereçamento global. Esses esquemas devem considerar que uma nomeação de dados baseada em atributos é utilizada para indicar os destinatários dos pacotes. Além disso, fatores como consumo de energia e expansibilidade, e características como roteamento centrado em dados (*data-centric*) exigem um tratamento diferente na camada de transporte de uma rede de sensores [POT 00].

## 2.4.3 A Camada de Rede

Como os nós de uma rede de sensores estão espalhados tipicamente de forma densa pela área que se quer monitorar e precisam atuar em conjunto para entregar dados para um determinado ponto, há a necessidade de protocolos de roteamento entre os nós sensores e o sink. As técnicas de roteamento *ad hoc* propostas na literatura normalmente não se adaptam

aos requisitos das redes de sensores devido às razões citadas em [PER 00] e na seção 2.1. A camada de rede deve ser projetada de acordo com os seguintes princípios:

- Eficiência no consumo de energia é sempre uma consideração importante;
- A maioria das redes de sensores é *data-centric*;
- Uma rede de sensores ideal possui endereçamento baseado em atributos e é ciente de localização.

A maioria das pesquisas em redes de sensores está concentrada na camada de rede [BOU 04][INT 03]. Esta camada é abordada em detalhes no capítulo 3, onde serão estudados os protocolos de roteamento para redes de sensores, pois o foco deste trabalho é a especificação e a simulação de um protocolo de roteamento.

#### 2.4.4 Camada de Enlace

A camada de enlace é responsável pelo controle de acesso ao meio e tratamento de erros, e garante comunicações confiáveis em uma rede de comunicação. Um protocolo de controle de acesso ao meio (MAC) para redes de sensores sem fio deve atingir dois objetivos [AKY 02]. O primeiro é a criação de uma infra-estrutura de rede. Como milhares de nós sensores estão densamente espalhados por um campo de sensores, o esquema MAC deve estabelecer enlaces de comunicação para transferência de dados. Isso forma a infra-estrutura básica necessária para a comunicação sem fio nó a nó e oferece a habilidade de auto-organização para a rede de sensores. O segundo objetivo é compartilhar eficientemente os recursos de comunicação entre os nós sensores [WUS 00].

Os protocolos MAC existentes não podem ser aplicados em redes de sensores por diversos fatores. Para melhor ilustrar o impacto das características de redes de sensores no projeto de um protocolo MAC, serão mostrados alguns esquemas MAC para outras redes sem fio.

Em um sistema celular, as estações base formam um *backbone* com fio. Um nó móvel está a apenas um *hop* de distância da estação base mais próxima. Esse tipo de rede é também referenciado na literatura como baseado em infraestrutura [AKY 02]. O objetivo principal dos protocolos MAC nesses sistemas é a garantia de uma melhor qualidade de



serviço (QoS) e eficiência na largura de banda. A economia de energia assume apenas uma importância secundária, pois as estações base possuem recursos de energia ilimitados e os usuários podem substituir uma bateria fraca em seus nós móveis. Nas redes de sensores, o consumo de energia ocupa o lugar de maior importância. Em muitas aplicações de redes de sensores, os nós são espalhados em locais de difícil acesso, inviabilizando a substituição de suas baterias. Por isso o tempo de vida da rede depende da eficiência no consumo de energia.

O *Bluetooth* [BLU 04] é um sistema sem infra-estrutura, de baixo alcance e sem fio planejado para substituir os cabos entre terminais eletrônicos com rádio frequência. A topologia do Bluetooth é uma rede em estrela onde um nó mestre pode ter até sete nós escravos conectados a ele para formar uma *piconet*. O alcance de transmissão é da ordem de algumas dezenas de metros. O protocolo *MAC* em uma *MANET* tem a tarefa de formar a infra-estrutura de rede e garantir a mobilidade dos nós. Além disso, o objetivo principal é garantir QoS sob condições móveis [WUS 00]. Os nós são dispositivos portáteis alimentados por bateria que o usuário pode substituir e, por isso, o consumo de energia tem uma importância secundária.

Em contraste com esses sistemas, uma rede de sensores pode abrigar um número muito maior de nós. O alcance de transmissão de um nó sensor é muito menor do que no Bluetooth ou na MANET. As mudanças de topologia são mais frequentes em uma rede de sensores, devido às falhas dos nós. Assim sendo, a importância da economia de energia para prolongar o tempo de vida de uma rede de sensores significa que nenhum dos protocolos MAC existentes para redes sem fio pode ser usado diretamente [TSE 01].

#### 2.4.4.1 Protocolos MAC para Redes de Sensores

Ficou evidente das discussões anteriores que um protocolo MAC para redes de sensores deve possuir estratégias para economizar energia. Embora vários esquemas tenham sido propostos para acesso ao meio em MANETs [TSE 01][WUS 00], o desenvolvimento de um esquema eficiente para MAC em redes de sensores ainda é uma questão aberta para pesquisa. Alguns dos protocolos MAC propostos são discutidos a seguir.

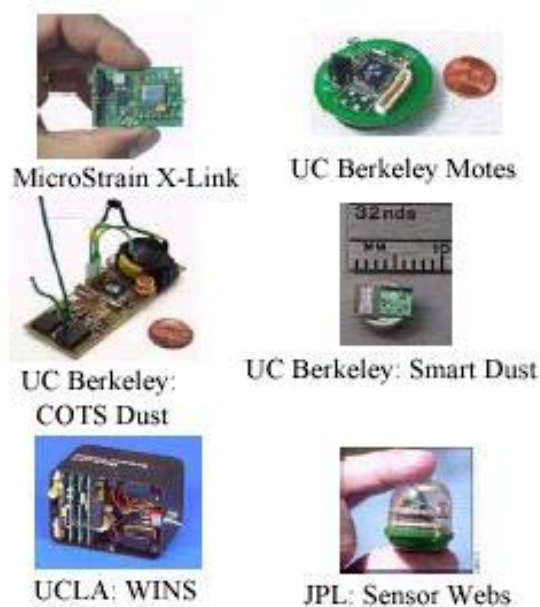
- SMACS – O SMACS [SOH 00] é um protocolo que realiza a organização da topologia e garante a comunicação dos nós em uma rede de sensores. Possibilita aos nós descobrirem seus vizinhos e estabelece um escalonamento para transmissão e recepção sem a necessidade de um nó mestre. A economia de energia é obtida ligando-se os nós durante a fase de comunicação e desligando-os durante os períodos ociosos.
- Acesso ao meio baseado em CSMA – Um esquema baseado em CSMA para redes de sensores é apresentado em [WOO 01]. Suas simulações mostram que os períodos de escuta constante presentes no CSMA são eficientes em energia e a introdução de atrasos aleatórios oferece robustez contra colisões repetidas. É sugerido neste trabalho que a energia consumida por unidade de comunicação bem sucedida pode servir como um bom indicador de eficiência em energia.
- Baseado em TDMA/FDMA (híbrido) – Este esquema controlado centralmente é introduzido em [SHI 01]. É assumido que o sistema é composto por nós sensores limitados em energia e que se comunicam com uma estação base central, próxima (<10 m) e com bons recursos de energia. É abordada no trabalho uma aplicação de monitoramento de máquinas, com requisitos rigorosos de latência dos dados. Enquanto um esquema TDMA puro dedica toda a largura de banda para um único nó, um esquema FDMA puro aloca uma largura de banda mínima para cada nó. Uma fórmula analítica é derivada em [SHI 01] para encontrar o número ótimo de canais que resulta no menor consumo de energia do sistema. Isso determina o esquema híbrido TDMA-FDMA a ser usado. O número ótimo de canais depende da razão entre os consumos de energia do transmissor e do receptor. Se o transmissor consome mais energia, um esquema TDMA é favorecido. O esquema se inclina para o FDMA quando o receptor consome mais energia.

#### 2.4.5 A Camada Física

A camada física é responsável por: seleção de frequência, geração da frequência portadora, detecção de sinal, modulação e encriptação de dados. A geração de frequência e detecção de sinal estão relacionadas mais ao projeto de *hardware* do transceptor.

Sabe-se que a comunicação sem fio de longa distância pode ser muito custosa em termos de energia e complexidade de implementação. Durante o planejamento de uma camada física para redes de sensores, o consumo de energia assume um papel importante, assim como os efeitos de reflexão, difração e degradação que o sinal pode sofrer. Em geral, a energia mínima requerida para transmitir um sinal sobre uma distância  $d$  é proporcional a  $d^n$ , onde  $2 \leq n < 4$  [RAP 96]. A escolha de um bom esquema de modulação também é um ponto crítico para a comunicação confiável em uma rede de sensores. Uma arquitetura de baixo consumo de espalhamento do espectro para redes de sensores é apresentada em [CHI 01].

A figura 2.4 mostra alguns dos projetos de nós sensores para redes de sensores sem fio. Suas características comuns são: muito pequenos, baixo consumo de energia, possuem sensores para luminosidade, temperatura, umidade, pressão, aceleração, etc; possuem módulos de rádio e processamento.



**Figura 2.4.** Exemplos de projetos de nós sensores.

A camada física para redes de sensores é uma grande área a ser explorada. As questões abertas de pesquisa variam do desenvolvimento de transceptores de baixo consumo a esquemas de modulação. O desenvolvimento de nós sensores extremamente pequenos e de baixo consumo é crucial para muitas aplicações em redes de sensores.

## 2.5 Considerações Finais

As características das redes de sensores como flexibilidade, tolerância a falhas, alta fidelidade no sensoriamento e baixo custo criam novas e excitantes áreas de aplicação para o sensoriamento remoto. No futuro, esta gama de aplicações levará as redes de sensores para o cotidiano das pessoas. Entretanto, a efetivação das redes de sensores precisa satisfazer as restrições impostas por fatores como tolerância a falhas, expansibilidade, custo, hardware, variações na topologia, e consumo de energia. O fato de que essas restrições são altamente rigorosas e específicas para as redes de sensores faz com que novas técnicas para redes sem fio *ad hoc* sejam necessárias. Vários pesquisadores estão atualmente engajados no desenvolvimento de tecnologias para os diferentes protocolos para redes de sensores. Uma lista dos projetos de pesquisa é dada na tabela 2.1. Os projetos na área de roteamento serão investigados no capítulo 3.

**Tabela 2.1.** Projetos de pesquisa.

Nome do projeto	Área de pesquisa	Localização http
SensorNet [AKY 02]	Camadas de transporte, rede, enlace e física.	<a href="http://www.ece.gatech.edu/research/labs/bwn/">http://www.ece.gatech.edu/research/labs/bwn/</a>
WINS [EST 01][POT 00]	Redes distribuídas e acesso a sensores pela Internet.	<a href="http://www.janet.ucla.edu/WINS/">http://www.janet.ucla.edu/WINS/</a>
SPINS [PEA 01]	Protocolos de segurança	<a href="http://paris.cs.berkeley.edu/~perrig/projects.html">http://paris.cs.berkeley.edu/~perrig/projects.html</a>
SINA [SHE 01]	Arquitetura de rede	<a href="http://www.eecis.udel.edu/~cshen/">http://www.eecis.udel.edu/~cshen/</a>
$\mu$ AMPS [SHI 01]	<i>Framework</i> para implementação de micro-sensores	<a href="http://www-mtl.mil.edu/research/icsystems/uamps">http://www-mtl.mil.edu/research/icsystems/uamps</a>
Smart dust [KAH 99]	Hardware para nós micro-sensores de baixo consumo	<a href="http://robotics.eecs.berkeley.edu/~pister/SmartDust/">http://robotics.eecs.berkeley.edu/~pister/SmartDust/</a>
DataSpace [IMI 99]	Processamento de consultas distribuído	<a href="http://www.cs.rutgers.edu/dataman">http://www.cs.rutgers.edu/dataman</a>

## 3 Protocolos de Roteamento para RSSF

---

### 3.1 Considerações Iniciais

É esperado que o uso de redes de sensores sem fio (RSSF) tenha um impacto significativo na eficiência de muitas aplicações militares e civis como supervisão de campo de batalha, monitoramento de desastres, e segurança. Esses sistemas processam dados coletados de múltiplos sensores para monitorar eventos em uma área de interesse. Por exemplo, um grande número de sensores pode ser lançado por um helicóptero em uma área de desastre. Trabalhando em rede, esses sensores podem auxiliar operações de resgate através da localização de sobreviventes, identificar áreas de risco e tornar os soldados mais informados sobre a situação. Tais aplicações de redes de sensores podem, além de aumentar a eficiência das operações de resgate, garantir uma maior segurança ao grupo de resgate.

Porém, os nós sensores são muito limitados quanto à energia e largura de banda. Tais limitações, combinadas com o uso de um grande número de nós, impõem diversos desafios no desenvolvimento e gerenciamento de redes de sensores. Esses desafios necessitam de soluções para economia de energia em todas as camadas a arquitetura. As questões relacionadas às camadas física e de enlace são geralmente comuns para todos os tipos de aplicações de sensoriamento, e as pesquisas nessas áreas focam em escalonamento dinâmico de tensão, hardware para comunicação por rádio, particionamento do sistema, e protocolos MAC que economizam energia [HEI 00][MIN 00][WOO 01][HED 02][SHI 01]. Na camada de rede, o objetivo principal é achar meios de roteamento eficientes em economia de energia e, simultaneamente, encaminhar dados dos sensores ao sink de forma confiável, de modo que o tempo de vida da rede seja maximizado.

O roteamento em RSSF é uma questão desafiadora devido às diversas características que as distingue das redes sem fio *ad-hoc* [PER 00]. Primeiro, protocolos clássicos baseados em IP não são adequados para serem aplicados em redes de sensores. Segundo, ao contrário das redes de comunicação típicas, quase todas as aplicações para RSSF requerem o encaminhamento de dados coletados de múltiplas regiões (fontes) para um determinado *sink*. Terceiro, o tráfego gerado pode ter uma redundância significativa,

pois múltiplos sensores podem gerar o mesmo dado para um fenômeno. Essa redundância de nós precisa ser adotada pelos protocolos de roteamento para melhorar o uso da energia e da largura de banda. Quarto, os nós sensores são altamente limitados pela energia de transmissão, capacidade de processamento e armazenamento, o que requer um cuidadoso gerenciamento de recursos.

Devido a essas diferenças, vários algoritmos foram propostos para resolver o problema do roteamento em RSSF [HEI 99][INT 00][INT 03][BRA 02][SCH 01][CHU 02][MAN 01][YAO 02][SHA 02][SAD 03]. Esses mecanismos de roteamento têm considerado as características dos nós sensores junto com os requisitos das aplicações e arquiteturas. Quase todos os protocolos de roteamento podem ser classificados nas seguintes categorias:

- centrado no dado (*data-centric*);
- hierárquico;
- baseado em localização;
- e ciente de *QoS*.

Um protocolo centrado no dado é baseado em consultas e depende do método de nomeação dos dados. Esta abordagem ajuda na eliminação de muitas transmissões redundantes. Protocolos hierárquicos focam a formação de clusters onde o nó central (*cluster head*) é capaz de realizar agregação e redução de dados para economizar energia.

Protocolos baseados em localização utilizam informação de posição para despachar dados para regiões específicas ao invés de difundir para toda a rede. A última categoria inclui as abordagens de roteamento que tentam alcançar alguns dos requisitos de qualidade de serviço e economia de energia simultaneamente.

Este capítulo tem como objetivo sumarizar os mecanismos de roteamento existentes divididos em suas categorias apropriadas.

## 3.2 Protocolos Centrados no Dado

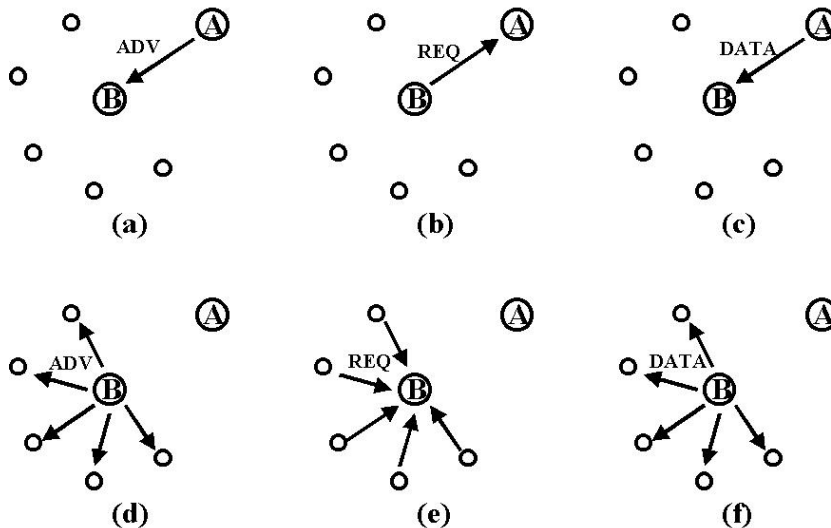
Em muitas das aplicações de redes de sensores sem fio, não é possível associar identificadores globais a cada nó devido ao grande número de nós envolvidos. Essa falta de identificação global junto com o posicionamento aleatório dos nós sensores dificulta a seleção de um conjunto específico de nós para ser consultado. Subseqüentemente, os dados normalmente são transmitidos de cada nó de uma certa região, gerando uma redundância significativa. Como esta abordagem é muito ineficiente em termos de consumo de energia, há inúmeras pesquisas em agregação de dados e capacidade de seleção de nós específicos.

No roteamento centrado no dado, o *sink* envia consultas a certas regiões e espera pelos dados dos sensores localizados na região selecionada. Como os dados são requisitados por meio de consultas, um sistema de nomeação se faz necessário para especificar as propriedades desses dados. O primeiro protocolo centrado em dados proposto foi o SPIN [HEI 99], que considera a negociação de dados entre os nós para eliminar a redundância e economizar energia. Depois, o paradigma Directed Diffusion [INT 00][INT 03] foi desenvolvido e tornou-se uma referência para os protocolos centrados em dados. Foram propostos alguns protocolos [BRA 02][SCH 01][CHU 02] baseados no Directed Diffusion, e outros seguindo conceitos similares [MAN 01][YAO 02][SHA 02][SAD 03]. Nesta seção serão discutidos os protocolos centrados em dados e suas idéias principais.

### 3.2.1 Protocolo de Sensores para Informação via Negociação – SPIN

O SPIN [HEI 99] é o trabalho precursor a perseguir o mecanismo de roteamento centrado no dado. A idéia por trás do SPIN é nomear os dados usando descritores de alto nível ou meta-dados. Antes de uma transmissão, meta-dados são trocados entre os sensores via um mecanismo de anúncio de dados, que é o aspecto principal do SPIN. Cada nó, ao receber um novo dado, anuncia esse dado aos seus vizinhos e, aqueles que não tiverem o dado, enviam uma mensagem de requisição para recebê-lo. A negociação dos meta-dados do SPIN resolve os problemas clássicos de *inundamento* (*flooding*) como a transmissão de dados redundantes, alcançando uma boa economia de energia. Não há um formato padrão para os meta-dados e é assumido como específico da aplicação. São três as mensagens

definidas no SPIN para trocar dados entre os nós: a mensagem ADV para um sensor anunciar um meta-dado particular, a mensagem REQ para requisitar um dado, e DATA que transporta o dado. A figura 3.1 resume as etapas do protocolo SPIN.



**Figura 3.1.** O protocolo SPIN. O nó A inicia anunciando seu dado para o nó B (a). O nó B responde enviando uma requisição (b). O nó A envia o dado (c) para o nó B que anuncia este dado para seus vizinhos (d). Os vizinhos de B respondem com requisições (e), que são atendidas por B, enviando o dado (f). (Adaptado de [HEI 99])

Comparando com o inundamento da rede, o SPIN é mais eficiente em consumo de energia e a negociação de meta-dados praticamente reduz pela metade os dados redundantes [HEI 99]. Porém, o mecanismo de anúncio de meta-dados não pode garantir a entrega dos dados. Por exemplo, se os nós interessados no dado estão longe do nó fonte e os nós entre fonte e destino não estiverem interessados neste dado, tal dado não será entregue ao destino. Conseqüentemente, o SPIN não é uma boa escolha para aplicações de monitoramento de condições críticas, como desastres e incêndios, por exemplo, que requerem uma entrega confiável de pacotes.

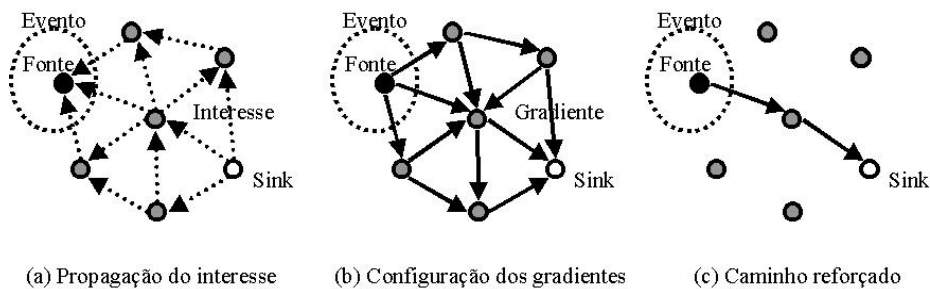
### 3.2.2 Directed Diffusion – DD

O Directed Diffusion [INT 00][INT 03] é um marco importante na pesquisa de roteamento data-centric em redes de sensores. A idéia focaliza a difusão de dados pelos nós sensores usando um esquema de nomeação de dados. A razão principal por trás do uso de tal esquema é acabar com as operações desnecessárias do roteamento na camada de rede,



economizando energia. O Directed Diffusion sugere o uso de pares atributo-valor para os dados, e consultas nos sensores, sob demanda, usando esses pares. Para criar uma consulta, um interesse é definido usando uma lista de pares atributo-valor como nome de objetos, intervalo, área geográfica, etc. O interesse é difundido (broadcast) na rede pelo sink. Cada nó que recebe o interesse pode armazená-lo para uso futuro. Os nós também possuem a habilidade de agregar dados. Os interesses armazenados nos nós são usados para comparar os dados recebidos com os valores nos interesses. O interesse também possui diversos gradientes. Um gradiente é um ponteiro para o vizinho do qual o interesse foi recebido. Utilizando interesses e gradientes, caminhos são estabelecidos entre o sink e os nós fontes.

Vários caminhos podem ser formados e apenas um é selecionado pelo mecanismo de reforço. O sink reenvia o interesse original, porém com um intervalo menor, através do caminho selecionado, forçando o nó fonte neste caminho a aumentar sua taxa de envio de



**Figura 3.2.** As fases do Directed Diffusion. (Adaptado de [INT 00])

dados. A figura 3.2 ilustra o protocolo Directed Diffusion.

O reparo de caminhos quebrados também é possível no Directed Diffusion. Quando um caminho entre um nó fonte e o sink falha, um caminho alternativo deve ser estabelecido. Para isso, o DD basicamente reinicia o mecanismo de reforço para procurar outros caminhos, que estão enviando dados em taxas baixas. Entretanto, o esquema de reparo de caminhos do DD possui um alto custo, pois exige um inundamento da rede para reforçar outros caminhos.

O DD difere do SPIN no mecanismo de consulta de dados sob demanda que ele possui. No DD o sink pergunta aos nós sensores se um dado específico está disponível, através do inundamento da rede com alguns interesses. No SPIN, os nós anunciam a disponibilidade de um dado permitindo que os nós interessados consultem aquele dado. O DD tem várias vantagens. Como ele é data-centric, toda comunicação é de vizinho para vizinho sem a necessidade de um mecanismo de endereçamento de nós. Cada nó pode realizar agregação e armazenamento temporário, em adição ao sensoriamento. Outra vantagem é a eficiência no consumo de energia, garantida pelo esquema sob demanda de consulta e por não necessitar de gerenciamento global da topologia da rede.

Entretanto, o DD não pode ser aplicado a todas as aplicações de redes de sensores, devido ao seu modelo de entrega de dados baseado em consultas. As aplicações que demandam entrega de dados contínua ao sink não funcionarão eficientemente neste modelo. Conseqüentemente, o DD não é recomendado como um protocolo de roteamento para aplicações como monitoramento ambiental e agricultura de precisão. Outro problema do DD é o esquema de nomeação de dados utilizado, que é dependente de aplicação e deve ser definido a priori. O processo de comparação de dados e consultas é outro entrave no DD, pois requer um custo extra nos nós.

### 3.2.3 Energy-Aware Routing

Shah *et al.* [SHA 02] propuseram o uso ocasional de um conjunto de caminhos para aumentar o tempo de vida da rede. Esses caminhos são escolhidos através de uma função de probabilidade, que depende do consumo de energia de cada caminho. A sobrevivência da rede é a métrica principal nesta abordagem. Esta abordagem alega que usando sempre um mesmo caminho, a energia dos nós deste caminho se esgotará rapidamente. Alternativamente, o protocolo escolhe um dos múltiplos caminhos com uma certa probabilidade, aumentando o tempo de vida da rede. É assumido que cada nó possui um endereço e informação sobre sua localização e seu tipo. O protocolo está dividido em três fases [SHA 02]: configuração, comunicação de dados, e manutenção de rotas.

Na fase de configuração ocorre uma *inundação* localizada para encontrar rotas e criar as tabelas de rotas. Durante este processo, o custo total de energia é calculado em cada

nó. Por exemplo, se uma requisição é enviada do nó  $N_i$  para o nó  $N_j$ , o nó  $N_j$  calcula o custo do caminho da seguinte forma:

$$C_{N_j, N_i} = \text{Custo}(N_i) + \text{Métrica}(N_j, N_i)$$

Aqui, a métrica de energia usada captura os custos de transmissão e recepção e também a energia residual dos nós. Os caminhos que tiverem um alto custo serão descartados. A seleção de um nó é feita de acordo com sua proximidade ao destino. O nó atribui uma probabilidade a cada um de seus vizinhos na sua tabela de roteamento ( $RT$ ) correspondente aos caminhos formados. A probabilidade é inversamente proporcional ao custo, que é:

$$P_{N_j, N_i} = \frac{1}{\sum_{k \in RT} \frac{1}{C_{N_j, N_k}}}$$

$N_j$  calcula então o custo médio para alcançar o destino através dos vizinhos na tabela de roteamento ( $RT_j$ ) usando a fórmula:

$$\text{Custo}(N_i) = \sum_{i \in RT_j} P_{N_j, N_i} C_{N_j, N_i}$$

Esse custo médio para  $N_j$  é atribuído ao campo custo da requisição.

Na fase de comunicação de dados cada nó encaminha o pacote escolhendo aleatoriamente um nó na sua tabela de roteamento usando as probabilidades.

Na fase de manutenção de rotas é realizado um inundamento localizado da rede para reparar qualquer caminho quebrado.

A abordagem descrita é similar ao Directed Diffusion no modo como os caminhos dos nós fontes ao sink são descobertos. No DD, o dado é enviado através de múltiplos caminhos, e apenas um deles é reforçado para enviar dados a taxas mais altas. Por outro lado, Shah *et al.* selecionam aleatoriamente um único caminho, dentre as múltiplas

alternativas, para economizar energia. Comparada ao DD, esta abordagem alcança uma melhoria de 21.5% na economia de energia e um aumento de 44% no tempo de vida da rede [SHA 02]. Porém, o uso de um único caminho impossibilita a recuperação do caminho quando um nó falha, ao contrário do DD. Outra desvantagem em relação ao DD é a necessidade de coletar informação de localização e configurar o mecanismo de endereçamento para os nós, complicando a fase de configuração de rotas.

### 3.2.4 Rumor Routing

O *Rumor Routing* [BRA 02] é outra variação do Directed Diffusion e é utilizado principalmente em aplicações onde critérios de roteamento geográfico não são aplicáveis. Normalmente o DD inunda a rede toda com interesses quando não há critérios geográficos para difundir tarefas. Porém, em alguns casos existe apenas uma pequena quantidade de requisições de dados e o uso do inundamento é desnecessário. Uma abordagem alternativa é inundar os eventos se o número de eventos for pequeno e o número de consultas for grande. O Rumor Routing está entre o inundamento de eventos e o inundamento de consultas. A idéia é guiar as perguntas aos nós que observaram um evento particular ao invés de inundar a rede toda para conseguir a informação sobre os eventos.

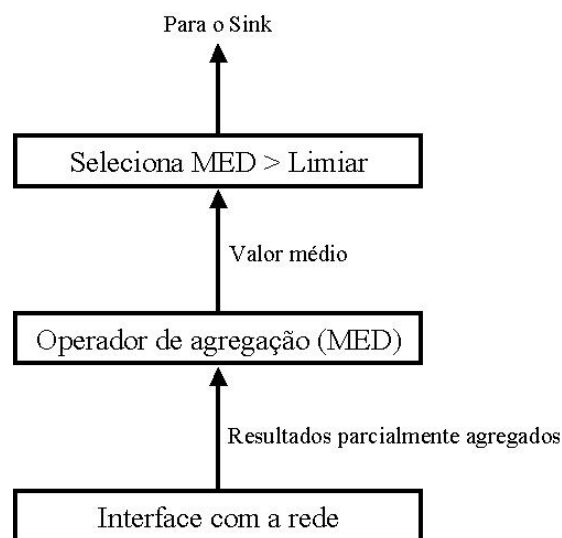
Para inundar a rede com eventos, o algoritmo utiliza pacotes de vida longa, chamados agentes. Quando um nó detecta um evento, ele adiciona tal evento em sua tabela local e gera um agente. O agente viaja pela rede com o objetivo de propagar a informação sobre os eventos locais para nós distantes. Quando um nó gera uma consulta para um evento, os nós que conhecem a rota podem responder à consulta. Por isso, o custo de inundar toda a rede é evitado. O rumor routing mantém apenas um caminho entre o nó fonte e o destino.

Resultados de simulação têm mostrado que o rumor routing alcança uma economia de energia significativa em relação ao inundamento e também pode tratar falhas em nós [BRA 02]. Porém, o rumor routing obtém resultados bons apenas quando o número de eventos é pequeno. Para um grande número de eventos, o custo de manutenção dos agentes e tabelas de eventos em cada nó não pode ser evitado, quando não houver interesses suficientes do sink nesses eventos.

### 3.2.5 COUGAR

Um protocolo *centrado no dado* que considera a rede como um grande sistema de banco de dados distribuído é proposto em [YAO 02]. A idéia principal é usar consultas declarativas para abstrair o processamento de consulta das funções da camada de rede como seleção de sensores relevantes, e utilizar agregação de dados para economizar energia. A abstração é apoiada por uma nova camada de consulta entre as camadas de rede e de aplicação.

O COUGAR propõe uma arquitetura para o sistema de banco de dados de sensores onde os nós sensores selecionam um nó líder para realizar a agregação e transmitir os dados ao *sink*. A arquitetura é mostrada na figura 3.3. O *sink* é responsável por gerar um plano de consulta, o qual especifica a informação necessária sobre o fluxo de dados e o envia para os nós relevantes. O plano de consulta também descreve como selecionar um nó líder para a consulta.



**Figura 3.3.** Plano de consulta em um nó líder: o nó líder captura todas as leituras, calcula a média e, se for maior que um limiar, envia para o sink [YAO 02].

Embora o COUGAR ofereça uma solução independente da camada de rede para consulta a nós sensores, ele possui alguns problemas. Primeiro, a introdução de uma nova camada em cada nó gera uma sobrecarga extra em termos de consumo de energia e

armazenamento. Segundo, os nós líderes devem ser gerenciados dinamicamente para preveni-los contra falhas.

### 3.2.6 ACQUIRE

Um mecanismo data-centric recente para aquisição de dados de uma rede de sensores é o *ACtive QUery forwarding In sensoR nEtworks* (ACQUIRE) [SAD 03]. Como no COUGAR [YAO 02], a abordagem considera a rede de sensores como um banco de dados distribuído. O mecanismo de consulta funciona da seguinte maneira: Uma pergunta é encaminhada pelo sink e cada nó que a recebe, tenta responder parcialmente usando informação pré-armazenada e a encaminha para outro sensor. Se a informação pré-armazenada estiver desatualizada, os nós coletam dados de seus vizinhos dentro de um tempo de vida (*time to live*) de  $d$  nós (*hops*). Assim que a pergunta é resolvida completamente, ela é enviada de volta pelo caminho reverso ou menor até o sink.

Uma das principais motivações para a proposta do ACQUIRE é tratar consultas complexas para dados onde a resposta pode ser providenciada por vários nós. As abordagens *data-centric*, como o Directed Diffusion, usam mecanismos de consulta baseados em inundamento da rede; não faria sentido usar o mesmo mecanismo para consultas complexas devido a considerações de energia. O mecanismo do ACQUIRE fornece uma consulta eficiente através do ajuste do valor do parâmetro  $d$ . Se  $d$  for igual ao tamanho da rede, então o algoritmo se comporta como um inundamento. Por outro lado, se  $d$  for muito pequeno, a consulta tem que atravessar mais nós.

Uma modelagem matemática foi derivada para o custo de energia do ACQUIRE e comparada com ambos inundamento e procura em anel. Um valor ótimo para o parâmetro  $d$  é calculado para uma matriz de sensores onde cada sensor possui quatro vizinhos. Porém, não há uma validação para os resultados através de simulação e os custos de recepção não foram levados em consideração durante os cálculos.

O problema de selecionar o próximo nó para encaminhar a consulta, que o ACQUIRE resolve, foi estudado no Rumor Routing [BRA 02]. O rumor routing tenta encaminhar a consulta para um nó, que sabe o caminho para o evento procurado. Como os

nós tornam-se cientes dos eventos através dos agentes, a heurística para definir a rota de um agente afeta negativamente o desempenho da seleção do próximo nó. No ACQUIRE, o próximo nó que encaminhará a consulta é selecionado aleatoriamente ou baseado no potencial máximo de satisfação da consulta [SAD 03].

### 3.3 Protocolos Hierárquicos

Similar às outras redes de comunicação, a expansibilidade é um dos atributos mais importantes de um projeto de redes de sensores. Uma rede com apenas um nível hierárquico pode gerar uma sobrecarga no sink com o aumento da densidade de sensores na rede. Essa sobrecarga pode causar atrasos na comunicação e levar a um rastreamento inadequado dos eventos. Além disso, uma arquitetura com apenas um único *gateway* não é expansível para um conjunto maior de sensores, cobrindo uma área extensa, pois os sensores são tipicamente incapazes de se comunicar a longas distâncias. Para permitir que o sistema suporte uma carga adicional e seja capaz de cobrir uma vasta área de interesse sem degradar o serviço, mecanismos de *clustering* em redes vêm sendo perseguidos em algumas abordagens de roteamento.

O objetivo principal do roteamento hierárquico é manter, de forma eficiente, um consumo de energia mais homogêneo entre os nós sensores, fazendo com que eles se comuniquem com um *cluster* particular de forma *multi-hop* e realizando agregação e fusão de dados para diminuir o número de mensagens transmitidas ao sink. A formação de *clusters* é tipicamente baseada na reserva de energia dos sensores e na proximidade dos sensores com o *cluster head* [BUC 98][LIC 97]. O LEACH [HEI 00a] é uma das primeiras abordagens de roteamento hierárquico para redes de sensores. A idéia proposta no LEACH serviu de inspiração para vários protocolos de roteamento hierárquico [MAN 01][LIN 02][LIN 01][MAN 02], embora alguns protocolos tenham sido desenvolvidos independentemente [SUB 00][YOU 02]. Alguns protocolos hierárquicos são detalhados nesta subsecção.

### 3.3.1 LEACH

Um dos algoritmos mais populares de roteamento hierárquico em redes de sensores é o *Low-Energy Adaptive Clustering Hierarchy* (LEACH) [HEI 00a]. A idéia é formar *clusters* de nós sensores baseados na força do sinal recebido e no uso de *cluster heads* como roteadores para o sink. A comunicação com o sink é feita apenas pelos *cluster heads*, ao invés de todos os sensores, economizando energia. O número ótimo de *cluster heads* é estimado em 5% do número total de nós [HEI 00a].

Todo o processamento de dados como fusão e agregação são locais ao *cluster*. A função de *cluster head* de um nó muda para outro nó aleatoriamente para balancear a dissipação de energia entre os nós. Essa decisão é tomada pelo nó, que escolhe um número aleatório entre 0 e 1. O nó torna-se um *cluster head* se o número for menor que o seguinte limiar:

$$T(n) = \begin{cases} \frac{p}{1 - p * (r \bmod \frac{1}{p})} & \text{se } n \in G \\ 0 & \text{caso contrário} \end{cases}$$

Aqui,  $p$  é a porcentagem desejada de *cluster heads* (por exemplo, 0.05),  $r$  é a rodada atual, ou seja, o período que um nó atua como *cluster head*, e  $G$  é o conjunto de nós que não foram *cluster heads* nas últimas  $1/p$  rodadas.

O LEACH é sete vezes mais eficiente em dissipação de energia se comparado com a comunicação direta [AKK 04]. Os nós “morrem” aleatoriamente e o mecanismo de *clustering* dinâmico aumenta o tempo de vida do sistema. O LEACH é completamente distribuído e não requer um conhecimento global da rede [HEI 00a]. No entanto, os *cluster heads* comunicam-se com o sink diretamente e, conseqüentemente, o LEACH não pode ser aplicado em redes que monitoram grandes regiões. Além disso, a idéia de *clustering* dinâmico carrega um custo extra, como as mudanças de *heads*, anúncio do estado de um *head*, etc., que podem atenuar o ganho em economia de energia.



### 3.3.2 PEGASIS

O *Power-Efficient Gathering in Sensor Information Systems* (PEGASIS) [LIN 02] é um melhoramento do protocolo LEACH. Ao invés de formar múltiplos *clusters*, o PEGASIS forma cadeias de nós sensores de tal forma que cada nó transmite e recebe de um vizinho e apenas um nó da cadeia é escolhido para transmitir para o sink. Os dados coletados passam de nó em nó, que são agregados e, eventualmente, enviados para o sink, como mostrado na figura 3.4, onde o nó c0 passa seu dado para o nó c1, que agrega o dado e entrega para o líder. Após o nó c2 passar a permissão de transmissão para o nó c4, este transmite seu dado para o nó c3, onde o dado é agregado e transmitido para o líder. O nó c2 aguarda para receber os dados de ambos vizinhos e então agrega seu dado com os dados dos vizinhos. Finalmente, o nó c2 transmite apenas uma mensagem para o sink.

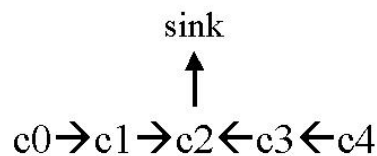


Figura 3.4. Encadeamento no PEGASIS.

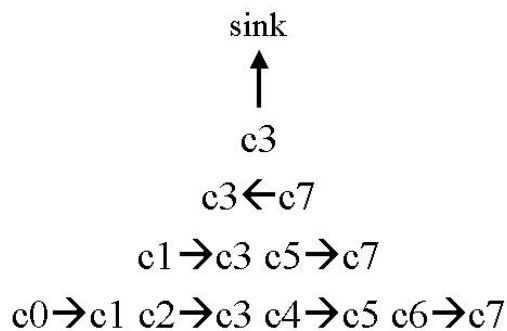
A diferença para o LEACH está no uso do roteamento de múltiplos *hops*, formando cadeias e selecionando apenas um nó para transmitir para o *sink* ao invés de usar múltiplos nós. O PEGASIS superou o LEACH nas simulações em torno de 100 a 300% para tamanhos e topologias diferentes de redes. O ganho de desempenho é alcançado através da eliminação da sobrecarga causada pela formação dinâmica de *clusters* no LEACH e através da diminuição do número de transmissões e recepções usando agregação de dados [LIN 02].

### 3.3.3 PEGASIS Hierárquico

O PEGASIS hierárquico [LIN 01] é uma extensão do PEGASIS, que visa diminuir o atraso gerado durante a transmissão de pacotes para o sink e propõe uma solução para o problema de agrupamento de dados considerando a métrica energia versus atraso. Para reduzir o atraso no PEGASIS, transmissões simultâneas de mensagens de dados são realizadas. Duas abordagens são investigadas para evitar colisões e possíveis interferências

de sinal entre os sensores. A primeira incorpora codificação do sinal, por exemplo, CDMA. Na segunda abordagem, apenas os nós separados espacialmente têm permissão de transmitir ao mesmo tempo.

O protocolo baseado em cadeia e com CDMA constrói uma cadeia de nós que forma uma hierarquia de árvore, e cada nó selecionado em um nível particular transmite dados para o nó em um nível mais alto na hierarquia. Esse método garante a transmissão de dados em paralelo e reduz o atraso expressivamente. Como exemplo, na figura 3.5 o nó c3 é selecionado como o líder para a rodada 3. Como o nó c3 está na posição 3 (contando a partir do 0) da cadeia, todos os nós em posições pares enviarão para seus vizinhos da direita. Os nós que recebem de um nó do mesmo nível sobem para o próximo nível na hierarquia. No próximo nível, o nó c3 ainda está em uma posição ímpar (1). Novamente, todos os nós em posições pares vão agregar seus dados com os recebidos e enviar para a direita. No terceiro nível, o nó c3 não está mais em uma posição ímpar, então o nó c7 vai agregar seu dado e transmitir para c3. Finalmente, o nó c3 irá combinar seu dado com o recebido de c7 e transmitir a mensagem para o sink.

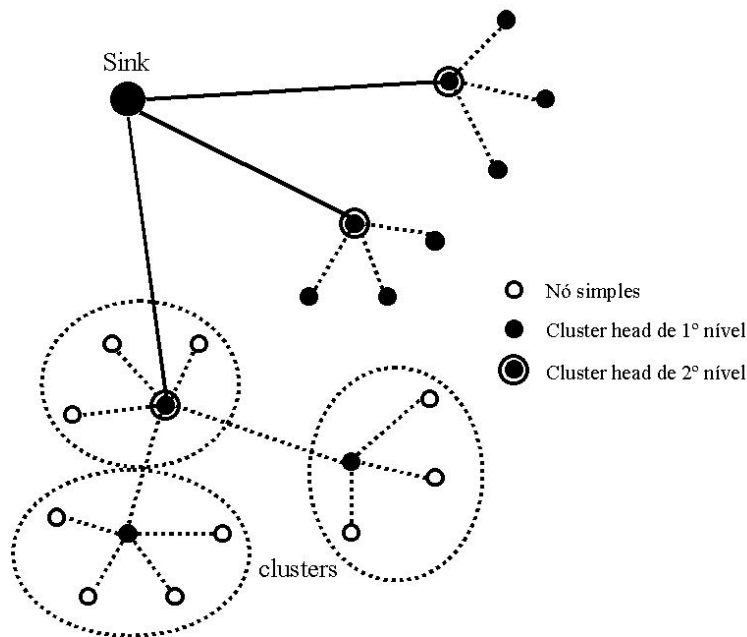


**Figura 3.5.** Agrupamento de dados em um esquema binário baseado em cadeia.

A abordagem que não se baseia em CDMA cria uma hierarquia de árvore de nós, e os efeitos de interferência são reduzidos por um cuidadoso escalonamento de transmissões simultâneas. Esse protocolo mostrou-se mais eficiente que o PEGASIS em torno de 60% [LIN 01].

### 3.3.4 TEEN

O *Threshold sensitive Energy Efficient sensor Network* (TEEN) [MAN 01] é um protocolo hierárquico desenvolvido para responder a mudanças repentinas nos atributos monitorados como temperatura. A habilidade de resposta rápida é importante para aplicações de tempo real, nas quais a rede opera em um modo reativo. O TEEN segue uma abordagem hierárquica com uso de um mecanismo data-centric. A arquitetura da rede de sensores é baseada em um agrupamento hierárquico onde os nós próximos formam *clusters*, e este processo prossegue no segundo nível até que o sink seja alcançado. O modelo é mostrado na figura 3.6.



**Figura 3.6.** Hierarquia e formação dos clusters no TEEN e APTEEN. [MAN 01]

Após a formação dos *clusters*, o *cluster head* difunde (*broadcast*) dois limiares para os nós, que são os limiares *soft* e *hard*. O limiar *hard* é o valor mínimo possível de um atributo para acionar um nó sensor a ligar seu transmissor e transmitir para o *cluster head*. O limiar *hard* permite aos nós transmitirem apenas quando o valor do atributo “sentido” estiver dentro da faixa de interesse, reduzindo o número de transmissões expressivamente. Quando um nó sensor captura um valor de atributo acima do limiar *hard*, ele apenas transmite o dado quando esse valor variar em uma quantidade igual ou maior que o limiar

*soft*. Conseqüentemente, o limiar *soft* reduzirá ainda mais o número de transmissões se houver nenhuma ou pequenas mudanças nos valores dos atributos monitorados. É possível ajustar ambos limiares *soft* e *hard* para controlar o número de transmissões de pacotes. Entretanto, o TEEN não é bom para aplicações que necessitem de leituras periódicas dos sensores, pois o sink pode não receber dado algum se os limiares não forem alcançados.

### 3.3.5 APTEEN

O protocolo APTEEN [MAN 02] é uma extensão do TEEN e visa cobrir leituras periódicas dos sensores, além de mudanças súbitas nos atributos. A arquitetura é a mesma do TEEN. Quando a formação de *clusters* alcança o sink, os *cluster heads* difundem os atributos e os limiares para todos os nós. Os *cluster heads* também realizam agregação de dados para economizar energia. O APTEEN possui três tipos de consultas: histórica, para analisar dados que já foram capturados; *one-time*, para conseguir uma visão geral da rede; e persistente, para monitorar um evento por um período de tempo.

As simulações dos protocolos TEEN e APTEEN superaram a performance do LEACH [HEI 00a]. Os experimentos demonstraram que o desempenho do APTEEN está entre o LEACH e o TEEN em termos de dissipação de energia e tempo de vida da rede [MAN 02]. O TEEN possui a melhor performance, pois diminui o número de transmissões [AKK 04]. Os principais pontos negativos das duas abordagens são a sobrecarga e complexidade da formação dos *clusters* em múltiplos níveis, a implementação de funções baseadas em limiares e o tratamento da nomeação de consultas baseada em atributos.

### 3.3.6 Energy-Aware Routing for Cluster-Based Sensor Networks

Younis *et al.* [YOU 02] propuseram um algoritmo diferente de roteamento hierárquico baseado em uma arquitetura de três níveis. Os sensores são agrupados em *clusters*. O algoritmo emprega *cluster heads*, chamados de *gateways*, que possuem uma reserva maior de energia que os sensores e assume que conhecem a localização dos nós sensores. Os *gateways* mantêm os estados dos sensores e configuram rotas com múltiplos nós para coletar os dados dos sensores. Um MAC baseado em TDMA é usado para os nós enviarem dados para o *gateway*. O *gateway* informa cada nó sobre os *slots* que os nós

devem receber dados de outros nós, e *slots* que o nó pode usar para suas próprias transmissões. O nó de comando (*sink*) se comunica apenas com os *gateways*.

É assumido que os sensores são capazes de operar em um modo ativo ou em espera, de baixo consumo. Os nós sensores em um *cluster* podem estar em um dos quatro principais estados: apenas sensoriando, apenas comunicando, sensoriando e comunicando, e inativo. No estado de sensoriamento, o nó monitora o ambiente e gera dados a uma taxa constante. No estado de comunicação, o nó não monitora o ambiente, mas seu módulo de rádio está ligado para encaminhar dados para outros nós ativos. Quando um nó está ao mesmo tempo monitorando o ambiente e comunicando com outros nós, é considerado que ele está no estado de sensoriamento-comunicação. Caso contrário, o nó é considerado inativo e pode desligar seus módulos de sensor e de comunicação.

A figura 3.7 mostra um exemplo dos estados dos sensores e as rotas em um *cluster*. Uma função de custo é definida entre qualquer par de nós em termos de energia, otimização de atraso e outras métricas de performance. Usando essa função de custo como sendo o custo de comunicação, um caminho de menor custo é definido entre os nós sensores e o *gateway*. O *gateway* monitora continuamente o nível de energia disponível em cada sensor que estiver ativo. A reconfiguração do roteamento é acionada por um evento relacionado à aplicação, requerendo um conjunto diferente de sensores para monitorar o ambiente.

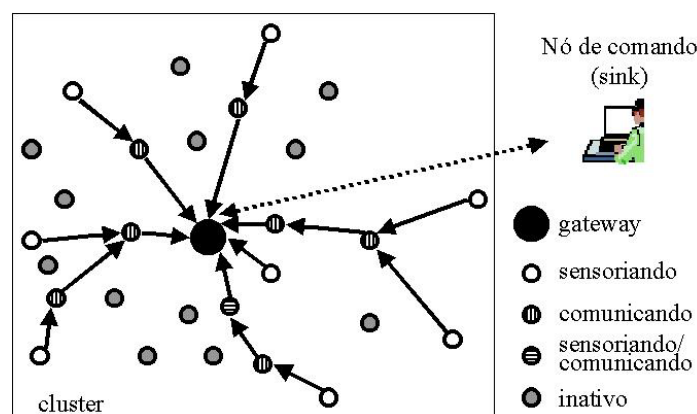


Figura 3.7. Um típico *cluster* em uma rede de sensores [YOU 02].

Uma variante desta abordagem de roteamento foi proposta em [YOS 02]. O algoritmo restringe a faixa mínima de transmissão para limitar o atraso. Os resultados de simulação demonstraram que tal abordagem tem uma boa performance com respeito a métricas de energia, tempo de vida da rede e atraso.

### 3.4 Protocolos Baseados em Localização

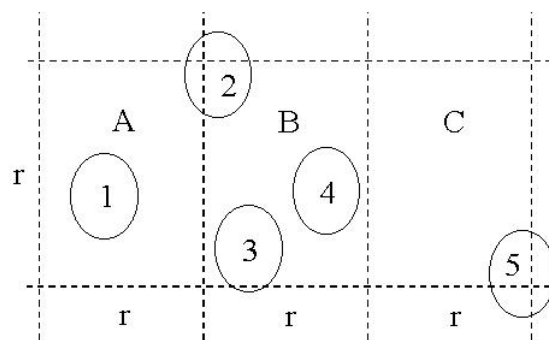
A maioria dos protocolos de roteamento para redes de sensores necessita de informação de localização para os nós sensores. Em grande parte dos casos, a informação de localização é necessária para calcular a distância entre dois nós para poder estimar o consumo de energia. Como não há um esquema de endereçamento para redes de sensores como os endereços IP, e os nós estão espalhados em uma região, a informação de localização pode ser utilizada no roteamento de dados de uma forma mais eficiente em termos de energia. Por exemplo, se a região a ser monitorada é conhecida, usando a localização dos sensores, as consultas podem ser difundidas apenas para uma região particular, reduzindo o número de transmissões significativamente. Alguns dos protocolos discutidos foram desenvolvidos primeiramente para redes *ad hoc* móveis [XUY 01][ROD 99][LIL 01]. Entretanto, eles podem ser aplicados em redes de sensores onde há pouca ou nenhuma mobilidade dos nós.

Existem outros protocolos baseados em localização para redes *ad hoc* sem fio, como *Cartesian* e *Trajectory-Based Routing* [NAT 02]. Porém, muitos desses protocolos não são aplicados em redes de sensores, pois eles não foram desenvolvidos considerando a economia de energia. Esta subseção limita-se a abranger apenas os protocolos baseados em localização que economizam energia.

#### 3.4.1 GAF

O *Geographic Adaptive Fidelity* (GAF) [XUY 01] é um algoritmo de roteamento baseado em localização desenvolvido inicialmente para redes móveis *ad hoc*, mas que pode ser aplicado em redes de sensores. O GAF conserva energia desligando nós desnecessários na rede sem afetar o nível de fidelidade do roteamento. Ele forma uma grade virtual para a área coberta. Cada nó usa sua localização GPS para se associar a um ponto na grade virtual.

Os nós associados com o mesmo ponto na grade são considerados equivalentes em termos de custo de roteamento de pacotes. Tal equivalência é explorada para manter alguns nós em estado de “dormência” para economizar energia. Além disso, o GAF pode aumentar o tempo de vida da rede substancialmente com o aumento do número de nós na rede. Um exemplo é mostrado na figura 3.8. O nó 1 pode alcançar qualquer um dos nós 2, 3 e 4 e os nós 2, 3 e 4 podem alcançar o nó 5. Então os nós 2, 3 e 4 são equivalente e dois deles podem “dormir”.



**Figura 3.8.** Exemplo de uma grade virtual no GAF. [XUY 01]

Os nós passam do estado de dormência para o ativo sucessivamente para promover o balanceamento de carga. Há três estados definidos no GAF: descoberta, para determinar os vizinhos na grade; ativo, para participar do roteamento; e dormência, quando o rádio é desligado. Qual nó irá dormir e por quanto tempo são parâmetros dependes da aplicação e são configurados durante o processo de roteamento. Para tratar a mobilidade, cada nó estima seu tempo de partida da grade e envia para seus vizinhos. Os nós dormentes ajustam seus tempos de dormência para manter a fidelidade do roteamento. Antes que o tempo de partida de um nó ativo termine, os nós dormentes acordam e um deles se torna ativo.

O GAF se empenha para sustentar a rede conectada como em [SUB 00], mantendo sempre um nó representativo no estado ativo em cada região da grade virtual. Os resultados de simulação mostram que o GAF tem uma performance ao menos tão boa quanto um protocolo normal de roteamento em redes *ad hoc* em termos de atraso e perda de pacotes, e aumenta a vida da rede economizando energia [XUY 01]. Embora o GAF seja um protocolo baseado em localização, ele pode também ser considerado como um protocolo hierárquico, onde *clusters* são baseados em localização geográfica. Para cada área da grade,

um nó representativo atua como um nó líder para transmitir dados para os outros nós. Porém, o nó líder não realiza agregação ou fusão de dados como no caso de outros protocolos hierárquicos.

### 3.4.2 GEAR

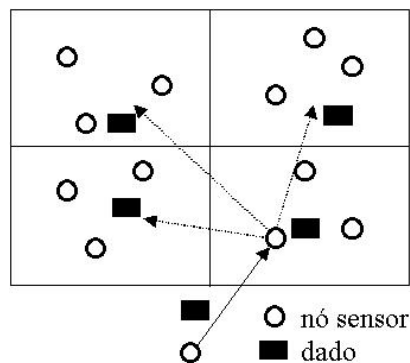
Yu *et al.* [YUY 01] propuseram o uso de informação geográfica durante a disseminação de consultas para regiões apropriadas, pois as consultas normalmente incluem atributos geográficos. O protocolo, chamado de *Geographic and Energy Routing* (GEAR), usa informação de energia e heurísticas de seleção geográfica de vizinho para rotear um pacote para uma região alvo. A idéia é restringir o número de interesses no Directed Diffusion considerando apenas uma certa região ao invés de enviar os interesses para toda a rede.

No GEAR, cada nó mantém um custo estimado e um custo aprendido durante a entrega de pacotes para o destino através dos vizinhos. O custo estimado é uma combinação de energia residual e distância do destino. O custo aprendido é um refinamento do custo estimado que calcula o custo do roteamento quando houver “buracos” na rede. Um buraco ocorre quando um nó não possui nenhum vizinho mais próximo da região alvo. Se não houver buracos, o custo estimado é igual ao custo aprendido. O custo aprendido é propagado um nó para trás toda vez que um pacote alcançar o destino, assim a configuração de rotas para o próximo pacote poderá ser ajustada. O algoritmo possui duas fases:

- 1) Encaminhamento de pacotes para a região alvo: Ao receber um pacote, o nó verifica seus vizinhos para saber se há um vizinho mais próximo da região alvo do que ele próprio. Se existir mais de um, o vizinho mais próximo da região alvo é escolhido como o próximo nó. Se todos eles estiverem mais longe da região alvo que o próprio nó, significa que existe um buraco. Neste caso, um dos vizinhos é escolhido para encaminhar o pacote baseado na função de custo aprendido. Esta escolha pode ser atualizada de acordo com a convergência do custo aprendido durante a entrega de pacotes.



- 2) Encaminhamento de pacotes dentro de uma região: Se o pacote alcançou a região alvo, ele pode ser difundido nesta região através de dois métodos: encaminhamento geográfico recursivo ou inundamento restrito. O inundamento restrito é bom quando há uma baixa densidade de sensores. Caso contrário, o encaminhamento geográfico recursivo é mais eficiente em termos de energia. Neste caso, a região é dividida em quatro sub-regiões e quatro cópias do pacote são criadas. O processo de divisão e encaminhamento continua até sobraem regiões com apenas um nó. Um exemplo é mostrado na figura 3.9.



**Figura 3.9.** Encaminhamento geográfico recursivo. Adaptado de [YUY 01].

O GEAR é comparado a um protocolo de roteamento similar para redes *ad hoc*, o GPSR [KAR 00], que é um dos primeiros trabalhos de roteamento geográfico que usa grafos planares para resolver o problema dos buracos. No caso do GPSR, os pacotes seguem o perímetro do grafo planar para encontrar suas rotas. Embora o GPSR diminua o número de estados que um nó deve permanecer, ele foi desenvolvido para redes móveis *ad hoc* e necessita de um serviço de localização para mapear posições e identificadores dos nós. O GEAR não apenas reduz o consumo de energia, como também tem um desempenho superior ao GPSR em termos de entrega de pacotes [YUY 01]. Os resultados de simulação mostram que o GEAR entrega 70 a 80% mais pacotes que o GPSR.

### 3.5 Protocolos Cientes de QoS

Apesar da maioria dos protocolos de roteamento propostos para redes de sensores se encaixarem nas categorias descritas até aqui, alguns seguem abordagens diferentes como a qualidade de serviço (QoS). Os protocolos cientes de QoS consideram o requisito de atraso

fim-a-fim durante a configuração dos caminhos em uma rede de sensores. Alguns protocolos são mostrados nesta seção.

### 3.5.1 SAR

O *Sequential Assignment Routing* (SAR) é o primeiro protocolo para redes de sensores que inclui a noção de QoS em suas decisões de roteamento [AKY 02][SOH 00]. É uma abordagem dirigida a tabelas e usa múltiplos caminhos para alcançar eficiência no consumo de energia e tolerância à falhas. O protocolo SAR cria árvores de nós levando em consideração métrica de QoS, energia residual em cada caminho e nível de prioridade de cada pacote. Através do uso de árvores, múltiplos caminhos são formados. Um desses caminhos é selecionado de acordo com os recursos de energia e a qualidade de serviço do caminho. A recuperação de falhas é feita através da manutenção da consistência da tabela de roteamento entre os nós fonte e destino de cada caminho. Qualquer falha local ativa um procedimento de restauração de caminhos. Os resultados de simulação mostram que o SAR oferece um baixo consumo de energia. O SAR mantém múltiplos caminhos dos nós para o sink. Esta abordagem garante tolerância à falhas, porém, o protocolo sofre com a sobrecarga para manter as tabelas e os estados em cada nó sensor, especialmente quando há um grande número de nós [AKY 02][AKK 04].

### 3.5.2 Energy-Aware QoS Routing Protocol

Um recente protocolo ciente de QoS para redes de sensores foi proposto por Akkaya e Younis [AKK 03]. O protocolo proposto estende a abordagem de roteamento proposta em [YOU 02] e acha um caminho de menor custo e mais eficiente em consumo de energia que satisfaz certo atraso fim-a-fim durante a conexão. O custo do *link* usado é uma função que captura a reserva de energia dos nós, a energia de transmissão, taxa de erro e outros parâmetros de comunicação.

Para tratar tráfego de tempo-real, gerado por sensores de imagens, e entrega rápida de pacotes ao mesmo tempo, é empregado um modelo de filas baseado em classes. O modelo permite o compartilhamento de serviços para tráfego de tempo-real e não-tempo-real. A razão  $r$  da largura de banda é definida como um valor inicial configurado pelo

gateway e representa a quantidade de largura de banda a ser dedicada para ambos os tráfegos de tempo-real e não-tempo-real em um *link* no caso de um congestionamento. O modelo de filas é mostrado na figura 3.10. O protocolo monta uma lista de caminhos de menor custo através do uso de uma versão estendida do algoritmo de Dijkstra, e seleciona um caminho da lista que satisfaz o requisito de atraso fim-a-fim.

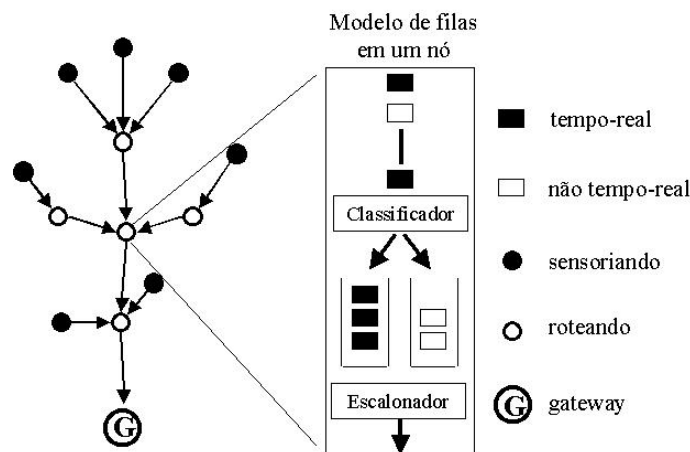


Figura 3.10. Modelo de filas em um nó particular. [AKK 03]

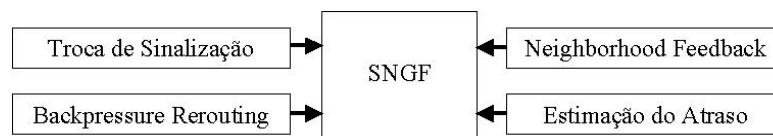
Os resultados de simulação mostram que o protocolo proposto tem um desempenho consistente com respeito às métricas de QoS e energia [AKK 03]. No entanto, o valor  $r$  é inicialmente o mesmo para todos os nós, o que não oferece um ajuste flexível do compartilhamento de largura de banda para *links* diferentes. O protocolo é estendido em [AKK 04a], onde é associado um valor diferente de para  $r$  para cada nó, alcançando uma melhor utilização dos *links*.

### 3.5.3 SPEED

Um protocolo de roteamento que considera qualidade de serviço para redes de sensores e descrito em [HET 03]. O protocolo requer que cada nó mantenha informação sobre seus vizinhos e usa encaminhamento geográfico para encontrar os caminhos. Além disso, o SPEED se empenha para garantir uma certa velocidade para cada pacote na rede. Cada aplicação pode estimar o atraso fim-a-fim para os pacotes através do quociente da

distância ao *sink* pela velocidade do pacote. O SPEED também pode evitar congestionamento quando o tráfego for intenso na rede.

O módulo de roteamento no SPEED é chamado de *Stateless Geographic Non-Deterministic Forwarding* (SNGF) e trabalha com outros quatro módulos na camada de rede, como mostrado na Figura 3.11. O mecanismo de trocas de sinalização coleta informação sobre os nós e suas localizações. A estimativa do atraso em cada nó é basicamente feita calculando-se o tempo decorrido quando um ACK é recebido de um vizinho em resposta a um pacote transmitido. Através dos valores de atraso observados, o SNGF seleciona um nó, que satisfaz o requisito de velocidade. Se um nó não puder ser encontrado, sua taxa de sucesso em transmissões é verificada. O módulo *Neighborhood Feedback Loop* é responsável pelo fornecimento do valor de sucesso de transmissão, o qual é calculado através das taxas de perda dos vizinhos de um nó (os nós que não forneceram a velocidade desejada), e é informado ao SNGF. Se a taxa de sucesso for menor que um número gerado aleatoriamente entre 0 e 1, o pacote é descartado. Finalmente, o módulo *Backpressure-Rerouting* é usado para prevenir vazios, quando um nó falha em encontrar um próximo nó, e para eliminar congestionamentos enviando mensagens de volta aos nós fontes para que eles procurem novas rotas.



**Figura 3.11.** Componentes de roteamento do SPEED. [HET 03]

Quando comparado com *Dynamic Source Routing* (DSR) [JOH 96] e *Ad-hoc On-Demand Vector Routing* (AODV) [PER 02], o SPEED mostra um desempenho melhor em relação ao atraso fim-a-fim e à taxa de perdas. Além disso, a energia total de transmissão é menor devido à simplicidade do algoritmo de roteamento, por exemplo, a sobrecarga para controle dos nós é menor e também pela distribuição do tráfego. Esse balanceamento de carga é alcançado através do mecanismo de dispersão de pacotes por uma grande área [HET 03].

### 3.6 Considerações Finais

O roteamento em redes de sensores tem atraído bastante atenção nos últimos anos e introduziu novos desafios que não são encontrados no roteamento em redes tradicionais. Neste Capítulo foram mostrados os resultados de pesquisas em roteamento de dados em redes de sensores e as abordagens classificadas nas três categorias principais: *data-centric*, hierárquica e baseada em localização. Alguns outros protocolos consideraram a qualidade de serviço em seus mecanismos de roteamento. No entanto, existem também os protocolos híbridos que se enquadram em mais de uma categoria, como mostra a Tabela 3.1. Foi incluída na tabela a informação de agregação de dados, pois é uma consideração importante para os protocolos de roteamento em termos de economia de energia e otimização de tráfego.

**Tabela 3.1.** Classificação dos protocolos de roteamento para redes de sensores.

<i>Protocolo de roteamento</i>	<i>Data-centric</i>	<i>Hierárquico</i>	<i>Baseado em localização</i>	<i>QoS</i>	<i>Agregação de dados</i>
SPIN [HEI 99]	x				x
Directed Diffusion [INT 00]	x				x
Rumor Routing [BRA 02]	x				x
Shah et al. [SHA 02]	x		x		
GBR [SCH 01]	x				x
CADR [CHU 02]	x				
COUGAR [YAO 02]	x				x
ACQUIRE [SAD 03]	x				
LEACH [HEI 00a]		x			x
TEEN [MAN 01]	x	x			x
APTEEN [MAN 02]	x	x			x
PEGASIS [LIN 02]		x			X
Younis et al. [YOU 02]		x	x		
Subramanian et al. [SUB 00]		x			x
GAF [XUY 01]		x	x		
GEAR [YUY 01]			x		
Akkaya et al. [AKK 03]		x		x	
SAR [AKY 02][SOH 00]				x	
SPEED [HET 03]			x	x	

Os protocolos que nomeiam os dados e consultam os nós baseando-se em alguns atributos dos dados são categorizados como *data-centric*. Muitos pesquisadores seguem

este paradigma para evitar a sobrecarga de formação de *clusters*, ou o uso de nós especializados. Porém, os esquemas de nomeação, como o par atributo-valor, podem não ser suficientes para consultas complexas e eles normalmente são dependentes da aplicação. A padronização de um esquema eficiente de nomeação é uma importante direção para futuras pesquisas relacionadas a esta categoria.

Já os protocolos baseados em agrupamento dos nós em *clusters* empenham-se para transmitir dados eficientemente ao sink. Na maioria das vezes os *cluster heads* são nós especializados com maiores reservas de energia. Um *cluster head* realiza agregação nos dados coletados pelos nós de seu *cluster* e transmite para o sink. A formação dos *clusters* é o desafio mais interessante nas pesquisas da área, pois otimiza as métricas de consumo de energia e atraso. Os fatores que afetam a formação de *clusters* e a comunicação entre *cluster heads* são questões em aberto para pesquisas futuras. Outro problema para se explorar é a agregação de dados entre os *clusters*.

Os protocolos que utilizam informação de localização dos nós sensores são classificados como baseados em localização. O problema da utilização inteligente da informação de localização para ajudar o roteamento a economizar energia é o tópico principal de pesquisa.

Embora o desempenho desses protocolos seja promissor em termos de consumo de energia, mais pesquisas são necessárias para resolver o problema da qualidade de serviço (QoS) imposto pelos sensores de imagem e vídeo e pelas aplicações de tempo-real. O roteamento ciente de energia e QoS em redes de sensores garantirá largura de banda (ou atraso) durante a conexão e também fornecerá o uso de caminhos de menor custo. O roteamento com QoS nas redes de sensores possui diversas aplicações que incluem rastreamento de alvos em tempo-real em batalhas, notificação de eventos críticos em aplicações de monitoramento, etc. Atualmente há pouca pesquisa no tratamento de requisitos de QoS em um ambiente com grandes limitações de energia como as redes de sensores.

Outra questão importante para os protocolos de roteamento é a mobilidade dos nós. A maioria dos protocolos assume que os nós sensores e o sink são estáticos. Porém, podem

ocorrer situações onde o sink e possivelmente os nós precisem ser móveis. Neste caso, a atualização e propagação constantes da posição do sink e dos nós podem consumir excessivamente a energia dos nós.

Outra linha de pesquisa para protocolos de roteamento inclui a integração de redes de sensores com redes tradicionais, como a Internet. A maioria das aplicações em segurança e monitoramento ambiental requer que os dados coletados dos nós sensores sejam transmitidos para um servidor remoto para análise. As consultas dos usuários poderiam ser feitas ao sink através da Internet. Como os requisitos de roteamento de cada ambiente são diferentes, há a necessidade de pesquisas para manipular esse tipo de situação.

## 4 Projeto do PEQ, um Protocolo de roteamento para aplicações que requerem monitoramento periódico, orientado a eventos e baseado em consultas

---

### 4.1 Considerações Iniciais

Com os recentes avanços nas tecnologias de redes sem fio e sensores com capacidade de processamento e comunicação, as redes de sensores sem fio estão sendo amplamente utilizadas para o monitoramento de ambientes em diversas classes de aplicação [BOU 04] [MIN 01]. Uma das aplicações mais apelativas é o monitoramento de situações críticas. Em uma prisão, por exemplo, é importante manter um monitoramento confiável do ambiente físico, especialmente quando surgem situações de emergência, como rebeliões de prisioneiros que podem levar a incêndios e perda de vidas humanas e de patrimônio. Nessas situações, é importante que as informações do ambiente possam ser “sentidas” enquanto o estado de emergência estiver em progresso, pois essas informações podem ser usadas pelos grupos de segurança ou resgate para o melhor gerenciamento da operação. Entretanto, para manter o fluxo de informações dos sensores durante uma emergência, uma solução de rede de sensores sem fio precisa tratar eventuais falhas nos nós sensores, que podem ocorrer devido a sensores queimados, comunicação prejudicada por interferências, esgotamento de energia, etc. Além de tolerante a falhas, uma rede de sensores para tais ambientes precisa ser confiável, oferecer baixa latência e economizar energia. Em termos de economia de energia, em um estado silencioso de monitoramento, os nós sensores podem ser programados para transmitir os dados em um esquema periódico (envie a temperatura a cada 10 minutos), ou podem seguir uma abordagem baseada em eventos (envie a temperatura apenas quando estiver acima de 60°C).

Algumas soluções existentes para economia de energia são baseadas no desligamento dos módulos de comunicação dos nós, os quais permanecem neste estado de “dormência” até serem “acordados” quando um evento desejado é acionado [INT 00]. Por



outro lado, em aplicações baseadas em consultas na rede, estas podem ser propagadas arbitrariamente aos nós sensores, e algumas soluções existentes para economia de energia não são adequadas, pois a transição de um estado de dormência para um estado de transmissão de dados pode ter um alto custo em termos de energia [CHT 02]. Além disso, a economia de energia e a tolerância a falhas podem apresentar interesses conflitantes quando novos caminhos, envolvendo nós dormentes, precisam ser rapidamente configurados devido a falhas em nós de outros caminhos.

Este capítulo descreve o PEQ (*Periodic, Event-driven, and Query-based Protocol*), um protocolo de roteamento rápido e confiável para redes de sensores sem fio que pode ser empregado em aplicações baseadas em periodicidade, eventos e consulta. O PEQ provê baixa latência para a notificação de eventos, rápida reconfiguração de caminhos falhos, alta confiabilidade na entrega de pacotes com baixa dissipação de energia. A baixa latência é alcançada com o uso dos caminhos mais curtos para a entrega dos eventos. Subscrições rápidas de novos interesses (para esquemas baseados em consulta) são possíveis devido ao conceito de entrega dirigida de subscrições, onde o envio de novas subscrições é acelerado através do uso do caminho inverso usado para notificação de eventos. Este conceito mostra ser eficiente em termos de economia de energia, pois menos pacotes são disseminados pela rede. A tolerância a falhas pode ser acionada localmente por um nó sensor, quando é detectada uma falha em um nó vizinho, onde um novo caminho é formado para contornar o problema.

Ao oferecer baixa latência, alta confiabilidade mesmo na presença de falhas, subscrição rápida de novos interesses e economia de energia, o PEQ torna-se uma boa escolha para as mais variadas áreas de aplicação como saúde (monitoramento de sinais vitais do corpo, instrumentação médica, localização de objetos e pessoas nas dependências de um hospital, etc), transportes (controle de tráfego, rastreamento de veículos, etc), militar (monitoramento do campo de batalha, apoio à tomada de decisões estratégicas em tempo real, etc), resgate e combate (monitoramento de situações críticas, como incêndios, prisões, etc), entre outras.

A rede de sensores é configurada através de uma árvore de *hops*, que é construída durante a fase de configuração inicial da rede. O paradigma *publish/subscribe* [EUG 03] é utilizado e adaptado para promover a interação entre os nós sensores e o *sink*. Para melhor descrever o algoritmo do PEQ, uma disposição em forma de grade dos nós sensores foi utilizada. Porém, o algoritmo pode ser aplicado a uma rede onde os nós foram espalhados aleatoriamente, como comprovado nas simulações.

Este capítulo está organizado da seguinte forma: a seção 4.1 descreve o algoritmo do PEQ dividido em etapas como configuração da rede, subscrição e notificação de eventos, e reparo de caminhos no caso de falhas. A seção 4.2 discute os experimentos realizados, os cenários de simulação, métricas e resultados obtidos. A seção 4.3 descreve a evolução do PEQ para uma abordagem hierárquica, o HPEQ. Finalmente, a seção 4.4 conclui o trabalho.

## 4.2 Descrição do PEQ

O PEQ é um protocolo para redes de sensores sem fio que visa atender os três esquemas de comunicação entre a aplicação e a rede, ou seja, Periódico, dirigido a Eventos e baseado em consultas (*Query-based*). A motivação principal para o trabalho descrito aqui é guiada pela necessidade de atender a todos os seguintes requisitos simultaneamente: baixa latência, confiabilidade, reparo rápido de caminhos na presença de falhas e economia de energia. Embora diversas soluções interessantes tenham sido propostas na literatura, elas basicamente não atendem a todos os requisitos ao mesmo tempo. Além disso, algumas soluções requerem hardware especial ou processamento sofisticado nos nós sensores. A idéia básica do algoritmo do PEQ é usar nós simples e baratos, com processamento simples e sem qualquer hardware especial, usando o nível de *hop* como a informação principal para minimizar a transmissão de dados. O algoritmo de roteamento do PEQ é realizado em três passos. O primeiro passo envolve a construção da árvore de *hops*. O *sink* inicia o processo de construção da árvore, que será usada para propagação de mensagens de subscrição e dados na rede. O segundo passo inclui a propagação de subscrições pela rede. Finalmente, o último passo é responsável pela entrega de dados dos sensores ao *sink*, usando rotas mais

rápidas e de baixo custo, em termos de energia. Durante este passo, é realizado também o reparo de caminhos caso algum nó falhe.

As próximas seções descrevem o paradigma *publish/subscribe* como o mecanismo de interação entre nós sensores e o *sink*, seguido pela descrição das etapas de roteamento. É assumido que os nós estão dispostos em forma de grade (ou matriz), separados a uma mesma distancia uns dos outros, de modo que a cobertura de transmissão de um nó é capaz de alcançar seus oito nós vizinhos. A comunicação entre os nós é simétrica, ou seja, todos possuem o mesmo alcance de transmissão. A solução também pode ser aplicada para uma rede onde os nós foram dispostos de forma aleatória, por exemplo, quando os nós são lançados no ambiente por um avião.

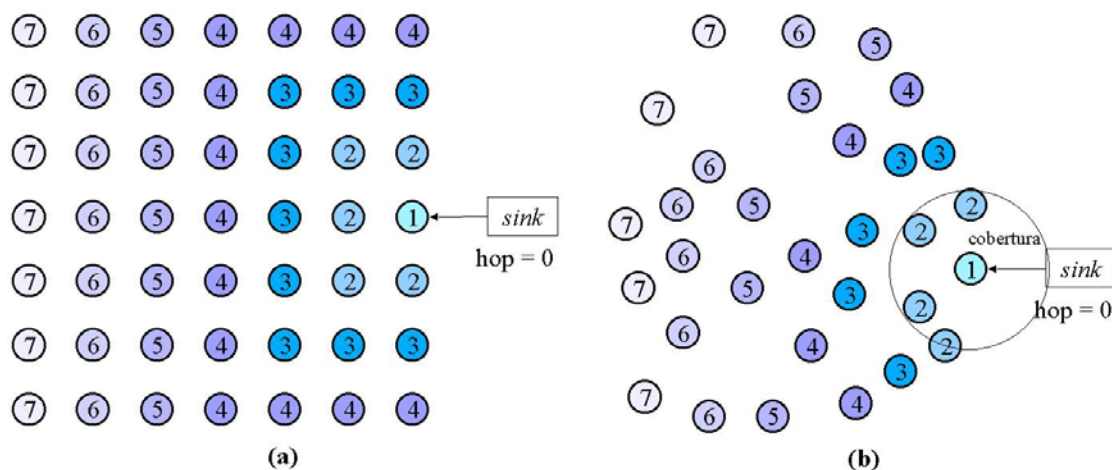
#### 4.2.1 O Paradigma *Publish/Subscribe* para Redes de Sensores

As redes de sensores podem possuir milhares de nós, cada um produzindo eventos que são entregues a um ou mais *sinks*. Diversos esquemas de comunicação podem ser usados para promover a interação entre os nós sensores que produzem eventos e os *sinks* que os consomem. Exemplos de tais esquemas incluem passagem de mensagem, invocação remota, notificação, espaços compartilhados, etc. O problema básico com esses esquemas é que eles não oferecem um desacoplamento total entre os participantes, tornando o sistema menos flexível e expansível [EUG 00]. Eugster et al. [EUG 03] realizou um excelente estudo sobre todos esses esquemas e os comparou com o paradigma *publish/subscribe*, o qual tem recebido atenção especial, pois ele desacopla produtores e consumidores no tempo (publicadores e subscritores não precisam estar ativos na interação ao mesmo tempo), espaço (publicadores e subscritores não precisam conhecer uns aos outros) e fluxo (publicadores e subscritores não precisam estar sincronizados para interagir). No paradigma *publish/subscribe*, o *sink* expressa interesse em um certo tipo de informação através da subscrição em determinados nós sensores. Quando um nó detecta alguma informação, ele verifica se existe alguma subscrição para este tipo de dado e o envia para o *sink* subscritor através de uma mensagem de notificação de modo assíncrono.

O mecanismo do publish/subscribe pode ser usado para receber e transmitir notificações dos seguintes tipos de interesse: informação periódica, baseada em consulta e dirigida a eventos. O algoritmo do PEQ possui uma solução simples e efetiva para a propagação de subscrições e notificações de eventos, como descrito nas próximas seções.

#### 4.2.2 Configuração Inicial da Rede

Na rede de sensores sem fio considerada, um nó sensor não possui uma visão global da rede, ou seja, um nó possui apenas informação de seus nós vizinhos mais próximos (aqueles que estão dentro de sua área de cobertura de comunicação). Em um primeiro momento, cada nó conhece apenas o nível de *hop* no qual ele se encontra. A árvore de *hops* é iniciada por um *sink*, que transmite para seus vizinhos um par atributo-valor chamado *HOP*. O algoritmo para a construção da árvore de *hops* é baseado no inundamento da rede, com um valor de *HOP* iniciado no *sink*, que é armazenado, incrementado e transmitido aos nós vizinhos (broadcast). Os nós vizinhos armazenam, incrementam e transmitem o valor de *HOP* recebido para seus nós vizinhos, e assim por diante, até que toda a rede esteja configurada com os níveis de *hops*. A figura 4.1 mostra a rede configurada para ambos os casos de disposição dos nós: grade e aleatória.



**Figura 4.1.** Configuração inicial da rede. (a) Nós dispostos em forma de grade. (b) Nós lançados aleatoriamente no ambiente.

Como a comunicação entre os nós da rede é realizada através da difusão de sinais de rádio (RF), todos os nós vizinhos recebem a transmissão. Dessa forma, um nó que acabou de transmitir uma mensagem, pode receber a mesma mensagem de seu vizinho, gerando um

caminho fechado (*loop*). Para evitar essas transmissões inúteis e que geram gasto de energia, um conjunto de regras foi estabelecido como parte do algoritmo de difusão de *HOP*. Uma das regras locais estabelece que cada nó, ao receber uma transmissão com o *HOP*, compara o *HOP* recebido com o seu *HOP* local. Se o valor do *HOP* local for maior que o valor recebido, o nó atualiza seu *HOP* com este valor, incrementa seu valor e o retransmite para seus vizinhos. No caso do *HOP* armazenado no nó ser menor ou igual ao *HOP* recebido, o nó não atualiza seu *HOP* e não transmite. O algoritmo de configuração inicial é mostrado na Figura 4.2.

```
// configTable (hop, sinkID, subTimeStamp);
// routingTable (nID, sID, sinkID, coord);
// subscriptionTable (type, criteria, coord,
// sinkID, destID, senderID, timestamp, hop);
config.hop = 1;
config.sinkID = sinkID;
config.subTimeStamp = clock();
config.sendConfigMsg();
// Quando um nó recebe uma mensagem de configuração,
// ele verifica sua configTable.
entry = configTable.get(config.sinkID);
if (entry) // Existe a entrada?
{
    if (entry.hop > config.hop)
    {
        entry.hop = config.hop;
        config.hop = config.hop + 1;
        config.sendConfigMsg();
    }
}
else // A entrada não existe!!
{
    entry.sinkID = config.sinkID;
    entry.hop = config.hop;
    configTable.add(entry);
    config.hop = config.hop + 1;
    config.sendConfigMsg();
}
```

**Figura 4.2.** Estrutura de dados e algoritmo de configuração inicial da rede.

A estrutura de dados do algoritmo compreende três tabelas: *configTable*, *routingTable* e *subscriptionTable*. A tabela *configTable* possui os parâmetros de configuração associados aos *sinks*. A tabela *routingTable* é usada por um nó para encaminhar mensagens para seus nós vizinhos. Finalmente, a tabela *subscriptionTable* é usada para armazenar subscrições que um nó recebe. A *routingTable* possui quatro campos:

*sinkID*, *senderID*, *destID*, e *coord*. O atributo *coord* é usado para indicar a posição do nó, assim a aplicação pode saber de qual local da rede vieram as leituras, e também o *sink* pode enviar uma subscrição diretamente para uma região delimitada por este atributo, ao invés de enviar para toda a rede ou para nós específicos.

### 4.2.3 Propagação de Mensagem de Subscrição

No paradigma *publish/subscribe*, para um *sink* ser notificado sobre eventos capturados do ambiente físico pelos nós sensores, ele precisa subscrever em um ou mais nós, através de critérios (temperatura > 60°C, presença de fumaça, etc) que precisam ser comparados antes de um evento ser enviado, ou seja, é um mecanismo *data-centric*. O envio de eventos apenas quando eles satisfazem algum critério reduz o tráfego na rede, causando um menor gasto de energia e estendendo a vida da rede. Após a configuração inicial da rede, a única informação que um nó possui é o nível de *hop* no qual ele está. Apenas esta informação não é suficiente para uma eficiente propagação de subscrição. Na ausência de alguma informação sobre qual nó da rede possa satisfazer o interesse do *sink*, uma solução para propagar a subscrição inicial para o nó correto seria inundar a rede com o interesse. Devido a este fato, uma mensagem de subscrição inicial pode ser enviada na etapa de configuração da rede.

Cada nó na rede mantém uma pequena tabela de subscrições e outra de roteamento. Cada registro da tabela de subscrições representa uma subscrição distinta. Durante a propagação da mensagem de subscrição, cada nó, ao recebê-la, compara o atributo *coord* da subscrição com suas próprias coordenadas. Se forem iguais, a subscrição é destinada para este nó e então é armazenada em sua tabela de subscrições. Caso contrário, o nó apenas retransmite a subscrição para seus vizinhos.

Durante a propagação da subscrição, quando um nó recebe um pacote de subscrição, ele processa o pacote e apanha o ID do nó que transmitiu e o *sinkID*. Então o nó atribui o ID do nó que transmitiu ao campo *destID* de sua *routingTable*, e adiciona o *sinkID* ao registro dessa rota. Essa informação será usada para encaminhar dados para o *sink*. Quando um nó precisa entregar dados a um *sink*, ele apenas verifica os campos *sinkID* e *destID* de sua *routingTable* e encaminha (*unicast*) o pacote para o nó correspondente ao *destID*. A

configuração inicial da rede já carrega consigo uma subscrição, desta forma todos os nós da rede são configurados com rotas para entregar dados ao *sink*.

#### 4.2.4 Envio da Mensagem de Notificação

Quando a informação é capturada do ambiente físico por um nó sensor, este verifica sua tabela de subscrições para determinar se existe algum interesse registrado. Se algum critério for satisfeito, o nó verifica o *senderID* do nó que transmitiu a subscrição. Após isso, o nó monta um pacote de notificação que contém os seguintes campos: *type*, *value*, *coord*, e *sinkID*, e o envia para seus vizinhos. Após receber o pacote, cada nó vizinho compara o *destID* recebido com o seu próprio ID. Se forem iguais, o nó armazena os atributos *coord* e *senderID* em sua tabela de rotas, configura o destinatário do pacote como sendo o *destID* de sua tabela de rotas e transmite. Cada nó no caminho repete o algoritmo até o pacote alcançar o *sink*.

Pode ser notado que, devido à característica da configuração inicial da rede, cada nó processa mensagens apenas dos nós que estão em um nível de *hop* anterior. Por exemplo, um nó no nível 5 recebe mensagens apenas de nós de nível 4, e assim por diante. Esta característica facilita a escolha do vizinho que transmitiu mais rápido, e também evita *loops* de mensagens. Supondo que um *sink* *S* envia uma subscrição para a rede configurada como mostra a figura 4.3(a) e considerando que o nó localizado no topo da rede e à esquerda seja o sensor que produz um evento que satisfaz a subscrição do *sink* *S*, a figura 4.3(b) mostra o caminho formado até o *sink* *S* para o envio da mensagem de notificação. Note que as setas indicam os *links* que poderiam formar caminhos alternativos, dependendo apenas da escolha de cada nó pelos *links* mais rápidos que entregaram a subscrição.

Uma característica importante da configuração dos *hops* pode ser observada na etapa de envio da notificação. Quando um nó recebe uma transmissão de um vizinho, ele apenas retransmite a mensagem se esta partiu de um nó de *hop* superior em uma unidade. Por exemplo, apenas os nós com *hop* = 4 repassam a informação transmitida por nós com *hop* = 5, e assim por diante. O algoritmo para a notificação é mostrado na figura 4.4.

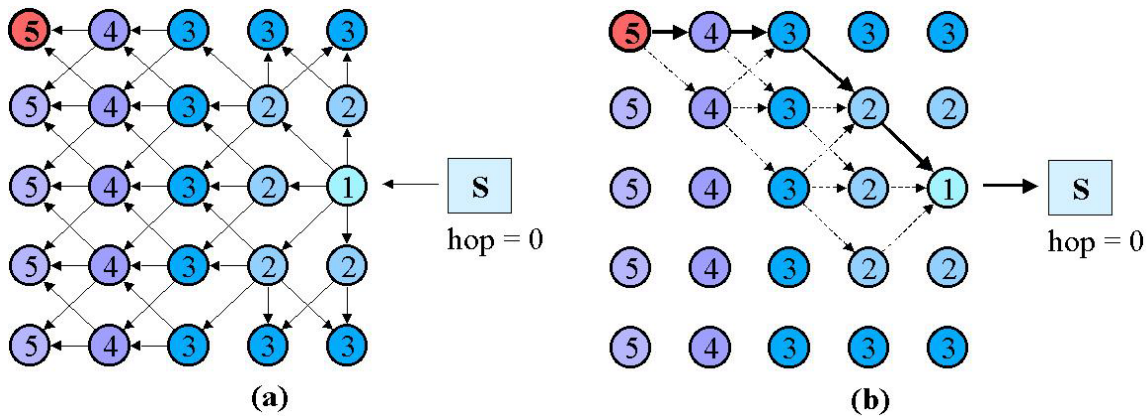


Figura 4.3. (a) Envio da subscrição. (b) Entrega dos dados ao sink.

```

// Quando um nó recebe um pacote de notificação.
if (notif.destID == node.ID)
{
    // Recupera o registro correspondente ao sinkID
    route = routingTable.get(notif.sinkID);
    // Armazena o senderID na tabela de rotas.
    route.senderID = notif.senderID;
    // Armazena o coord na tabela de rotas.
    route.coord = notif.coord;
    // Atribui o ID do destinatário.
    notif.destID = route.destID;
    // Envia a mensagem.
    notif.sendNotificationMsg();
}

```

Figura 4.4. Algoritmo de notificação de eventos.

#### 4.2.5 Envio Dirigido de Mensagens de Subscrição

O objetivo principal do algoritmo de propagação de mensagens de notificação é encontrar o caminho mais rápido entre o nó que produziu o evento e o *sink*. O caminho utilizado para entrega de notificações pode ser usado posteriormente pelo *sink* para enviar subscrições para a mesma região delimitada pelo atributo *coord*. Para isso, durante a entrega da notificação, cada nó no caminho para o *sink* armazena, na tabela de roteamento, os atributos *coord* e *senderID* do nó vizinho que transmitiu a mensagem de notificação. O *senderID* na tabela de rotas serve para configurar o caminho reverso do envio de notificações. Assim, só transmitirão os nós que serviram de rota para notificações anteriores. Resumindo, para acelerar novas subscrições para uma região de sensores, a etapa de subscrição pode usar o caminho inverso utilizado por notificações, como mostrado



na figura 4.5 (chamado de envio dirigido). Isto é útil quando subscrições do tipo consulta (*query-based*) são utilizadas. Caso contrário, se o nó não possuir um valor para *coord* compatível com o da transmissão, o nó transmite sem especificar o *senderID*, assim todos vizinhos transmitirão a mensagem de subscrição.

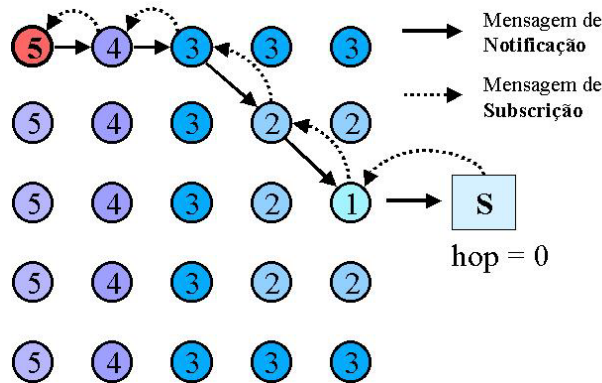


Figura 4.5. Envio dirigido de subscrições.

Como o envio dirigido de subscrições utiliza o mesmo caminho criado para mensagens de notificação, apenas aqueles nós que formam o caminho é que gastam energia transmitindo. Os outros nós, ou apenas recebem e não transmitem (no caso dos nós vizinhos aos nós do caminho, e que não pertencem a ele), ou nem sequer recebem mensagens. A figura 4.6 mostra um mapa que representa o consumo de energia da rede na utilização do caminho especificado. Nos nós mais escuros há um maior consumo de energia.

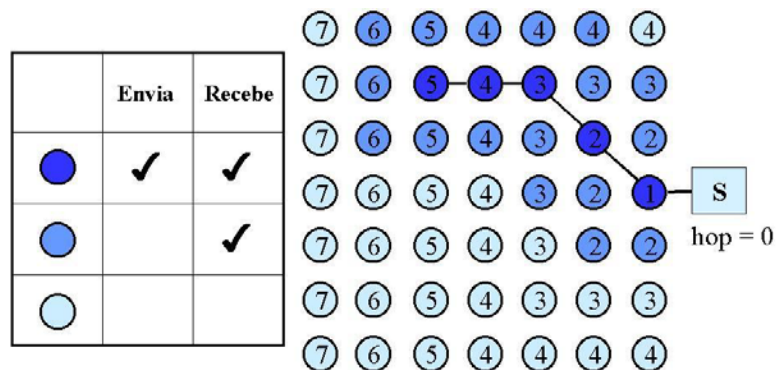


Figura 4.6. Comparação do consumo de energia em um caminho.

### 4.2.6 Reparo de Caminhos

Os caminhos criados para o envio de mensagens de notificação de eventos são únicos e eficientes (promovem baixa latência e economizam energia). Esses caminhos também podem ser usados para o envio dirigido de novas subscrições. Entretanto, pelo fato de um nó sensor usar um caminho único para encaminhar dados ao *sink*, qualquer falha em um dos nós desse caminho evitará que os dados alcancem o *sink*. Possíveis causas de falhas incluem: baixa energia, destruição física do nó, interferência do sinal, barreiras na comunicação, etc. Vários algoritmos de roteamento foram propostos na literatura, como discutido no capítulo 3. Alguns utilizam mecanismos de inundamento periódico [INT 00][INT 03][BOU 03], iniciados pelo *sink*, para reparar caminhos e descobrir novas rotas para contornar o problema. Esse mecanismo não é satisfatório em termos de economia de energia, pois há um grande consumo para enviar mensagens de reparo (broadcast). Além disso, durante os intervalos entre um reparo e outro, esses algoritmos são incapazes de encaminhar dados entre os nós falhos, causando perda de pacotes. Inicialmente, a solução apresentada no PEQ consistia de “escutar” as transmissões do vizinho para saber se ele encaminhou com sucesso o pacote que recebeu. Por exemplo, um nó sensor envia um pacote para seu vizinho e aguarda um tempo para detectar se este vizinho conseguiu transmitir com sucesso. Se o nó não escutou a transmissão do vizinho, ele infere que seu vizinho não está funcionando corretamente e inicia um algoritmo de escolha de outro vizinho para enviar seus pacotes. Porém, este mecanismo de “escutar” transmissões de vizinhos é dependente de MAC. Para ser capaz de funcionar em qualquer tecnologia, o algoritmo do PEQ foi modificado para oferecer um mecanismo de reparo baseado em ACK. O mecanismo consiste de duas partes: detecção de falha do nó destino e seleção de um novo destinatário.

Logo após a etapa de configuração inicial da rede, cada nó possui apenas um nó destino, em sua tabela de rotas, para encaminhar dados ao *sink*, devido ao caminho único criado. Quando um nó, chamado aqui de pai, precisa enviar dados para seu nó vizinho, chamado de filho, ele simplesmente envia o pacote e inicia um temporizador e aguarda a resposta (ACK) do nó filho. Se o nó pai recebeu o ACK do filho, ele pode deduzir que o filho está “vivo”. O nó filho envia a mensagem ACK logo após ter encaminhado adiante o

pacote recebido do pai, para evitar o problema que ocorre quando o nó filho “morre” depois de enviar o ACK e antes de encaminhar o pacote.

Se o nó pai não recebe o ACK do filho, deve ter ocorrido algum problema com o filho e outro nó vizinho deve ser adotado como novo filho para encaminhar pacotes. Então o pai imediatamente envia uma mensagem SEARCH para todos seus vizinhos. Os vizinhos respondem com a mensagem (REPLY) que contém o nível de *hop* e o ID. O próximo passo é seleccionar o novo nó filho. O pai então escolhe o vizinho com o menor nível de *hop* para ser seu novo filho. No caso de empate, o pai escolhe o vizinho que transmitiu primeiro. O pai atualiza sua tabela de rotas para facilitar o encaminhamento de pacotes subsequentes. Para evitar caminhos fechados (*loops*), o nó pai atualiza seu nível de *hop* com o valor incrementado do *hop* do filho. Se nenhum vizinho responder a mensagem SEARCH, o pai retransmite a mensagem. Se o pai estiver isolado (não alcança nenhum vizinho), a única solução seria aumentar o alcance de suas transmissões. Um exemplo de construção de um novo caminho é mostrado na figura 4.7.

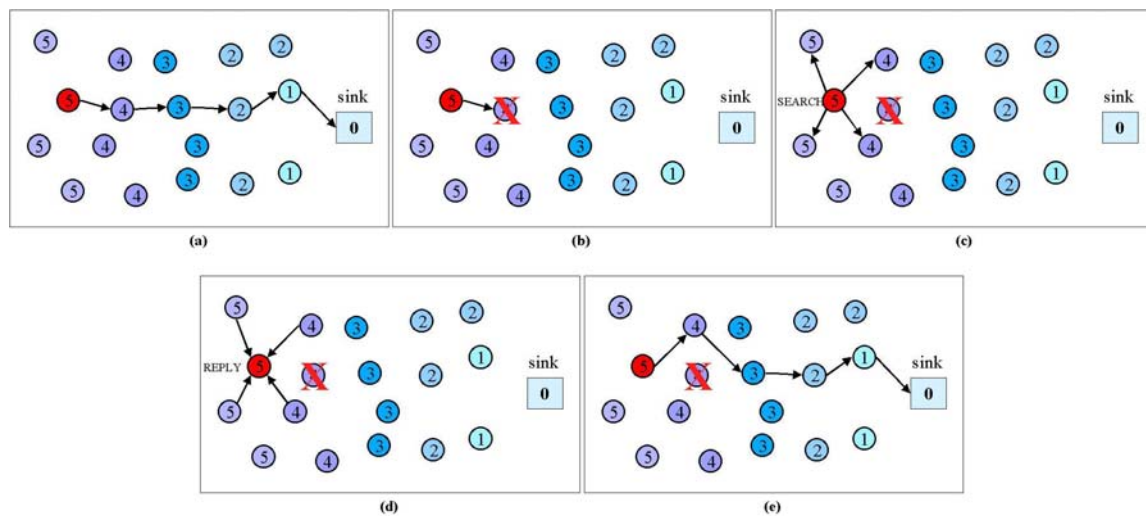


Figura 4.7. Mecanismo de reparo de caminhos do PEQ.

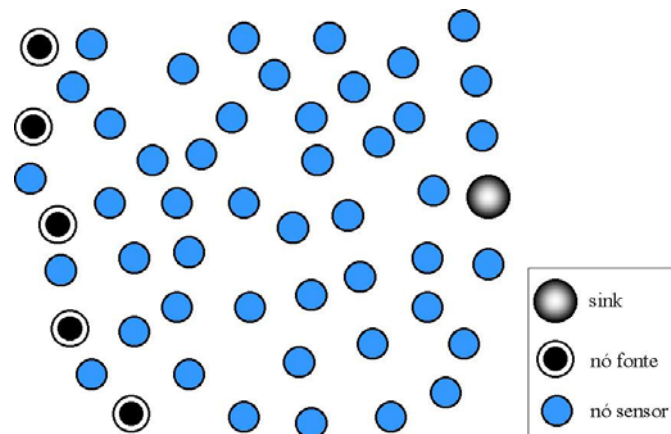
### 4.3 Experimentos de Simulação

Esta seção descreve a investigação do desempenho do PEQ através de um extenso conjunto de experimentos de simulação. Os resultados foram comparados com o conhecido

esquema de comunicação *Directed Diffusion* [INT 00] para redes de sensores, pois é o que mais se aproxima do PEQ.

### 4.3.1 Cenário de Simulação e Métricas

O PEQ foi implementado no simulador de redes ns-2 [NS2 04], utilizando as linguagens C e OTCL. O cenário de simulação consiste de diversos campos de sensores com diferentes tamanhos, variando de 100 a 500 nós sensores. Os nós foram colocados aleatoriamente no campo e foram utilizados cinco nós fontes e um *sink*. Esse nós fontes foram posicionados no lado esquerdo do campo e o *sink* no lado direito, como mostra a



**Figura 4.8.** Exemplo de cenário de simulação.

figura 4.8. Desse modo, quando o tamanho da rede aumenta, os caminhos entre nós fontes e o *sink* também aumentam, garantindo uma melhor observação do impacto do tamanho da rede no desempenho do PEQ.

A tabela 4.1 contém os parâmetros de simulação utilizados. A dissipação de energia dos nós está baseada nos estudos reportados em [INT 00], e o alcance das transmissões foi configurado em 20 metros para representar realisticamente o módulo de rádio dos sensores.

O PEQ e o *Directed Diffusion* foram simulados com os mesmos valores e cenários. Cada valor reportado nos gráficos foi extraído de uma média de 20 simulações. O atraso na entrega de eventos e a taxa de sucesso da entrega são métricas críticas para o desempenho

de aplicações de monitoramento em tempo-real. Foram utilizadas as seguintes métricas para avaliar o PEQ:

- **Atraso médio sink-fonte-sink:** é a latência média observada entre o momento que o *sink* requisita uma informação da rede e o momento que o evento correspondente é recebido pelo *sink*;
- **Atraso médio fonte-sink:** tempo médio medido entre a transmissão de um pacote por um nó fonte e seu recebimento pelo *sink*;
- **Taxa de entrega de pacotes:** é a relação entre o número de pacotes distintos recebidos pelo *sink* e o número de pacotes originalmente enviados pelos nós fontes.

**Tabela 4.1.** Parâmetros de simulação.

Parâmetros	Valores
Tempo de simulação	500 seg.
Número de nós	100-500
Taxa de envio de dados	10 eventos/s
Alcance do rádio	20 m
Energia para transmissão	14.88 mW
Energia para recepção	12.50 mW
Energia no modo ocioso	12.36 mW
Energia no modo Sleep	0.016 mW

### 4.3.2 Resultados Obtidos

A métrica sink-fonte-sink é particularmente importante para aplicações de monitoramento em tempo real, que demandam respostas rápidas e confiáveis, como localização de pessoas em uma situação de emergência. Em tal cenário, pessoas podem se deslocar rapidamente em poucos segundos, e a informação de localização pode não refletir com precisão a realidade, impossibilitando tomadas de decisão que poderiam salvar vidas se aplicadas dentro de um tempo seguro. O PEQ usa uma mensagem de subscrição para propagar a configuração inicial da rede que constrói os caminhos para o *sink*. Quando os nós fontes recebem a subscrição, eles usam esses caminhos para entregar dados ao *sink*. O *Directed Diffusion*, por outro lado, propaga a subscrição (interesses) para todos os nós. Após isso, os nós fontes enviam eventos exploratórios para o *sink* usando caminhos

múltiplos, e o *sink* irá reforçar um desses caminhos. A criação de caminhos do PEQ possui menos passos e é mais rápida que o *Directed Diffusion*, resultando em atrasos menores, como

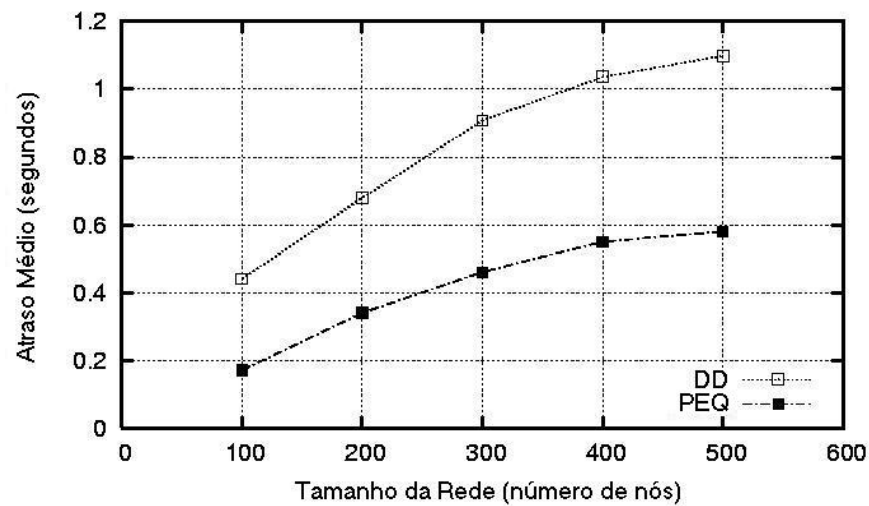


Figura 4.9. Gráfico dos atrasos médios sink-fonte-sink.

mostrado na figura 4.9.

As falhas nos nós foram simuladas desligando-se uma porcentagem dos nós da rede. Esses nós foram escolhidos aleatoriamente no campo de sensores e desligados durante a simulação simultaneamente, e permaneceram desligados até o término da simulação. O atraso médio da entrega de pacotes em uma rede com falhas é mostrado na figura 4.10.

Com o aumento do tamanho da rede, o atraso também aumenta devido ao maior número de nós que um pacote tem que atravessar para alcançar o *sink*. Isso faz sentido, pois o algoritmo precisa encontrar nós “vivos” para reparar caminhos quebrados. Além disso, com o aumento da porcentagem de falhas na rede, os novos caminhos encontrados são mais longos. Como mostrado no gráfico da figura 4.10, para um tamanho fixo da rede, por exemplo 500 nós, o atraso do PEQ aumenta de 0.049 para 0.058 segundos, uma latência aceitável para o cenário de aplicação de tempo real considerado.

Comparado com o *Directed Diffusion*, o algoritmo do PEQ resultou em latências menores, como pode ser observado na figura 4.11, onde foram simulados os algoritmos com 30% de falhas na rede. O PEQ é mais rápido devido ao mecanismo complexo que o DD emprega para encontrar caminhos. A confiabilidade da rede pode ser medida através da taxa de sucesso de entrega de eventos ao *sink*.

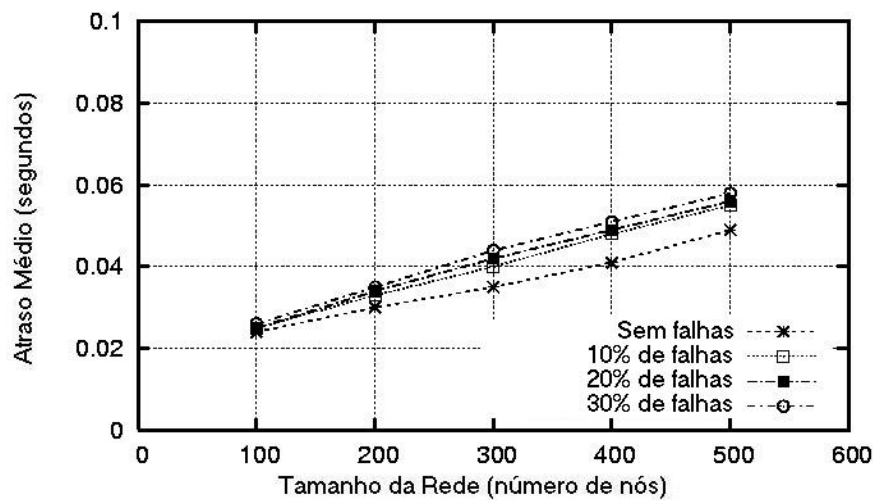


Figura 4.11. Gráfico do atraso médio do PEQ.



Figura 4.10. Gráfico dos atrasos médios sink-fonte-sink.

A 4.12 que o capaz de

figura mostra PEQ é manter

uma alta taxa de entrega de eventos mesmo com a presença de falhas na rede. Comparado

---

com o DD, o PEQ mostrou-se melhor nesta métrica, pois utiliza um menor número mensagens para configurar os caminhos e entregar os dados ao *sink*, resultando em uma menor quantidade de colisões de pacotes, como mostrado na figura 4.13. A energia média dissipada por nó no PEQ é apresentada na figura 4.14.



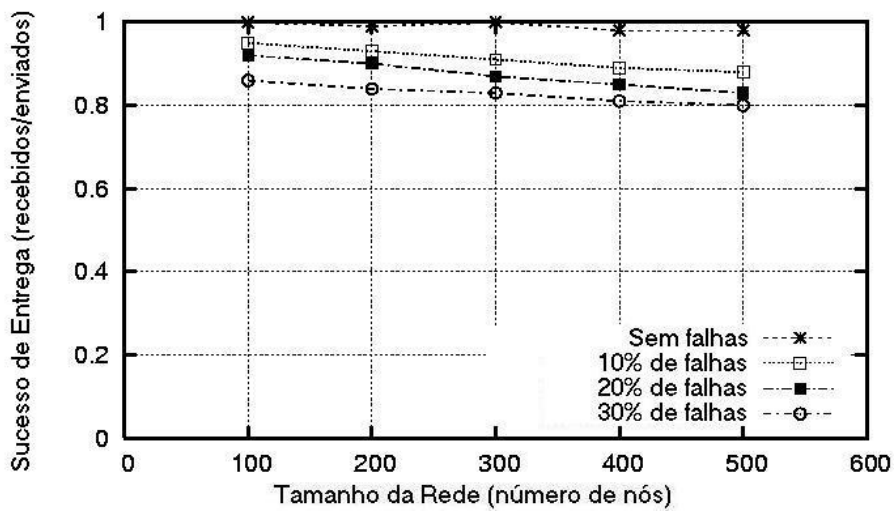


Figura 4.12. Gráfico da taxa de sucesso de entrega de eventos.

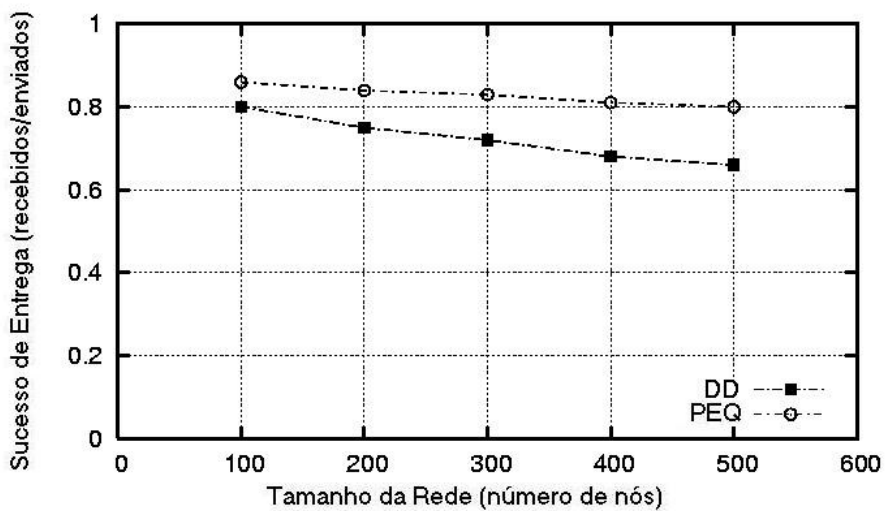


Figura 4.13. Gráfico de comparação das taxas de entrega do DD e PEQ (30% de falhas).

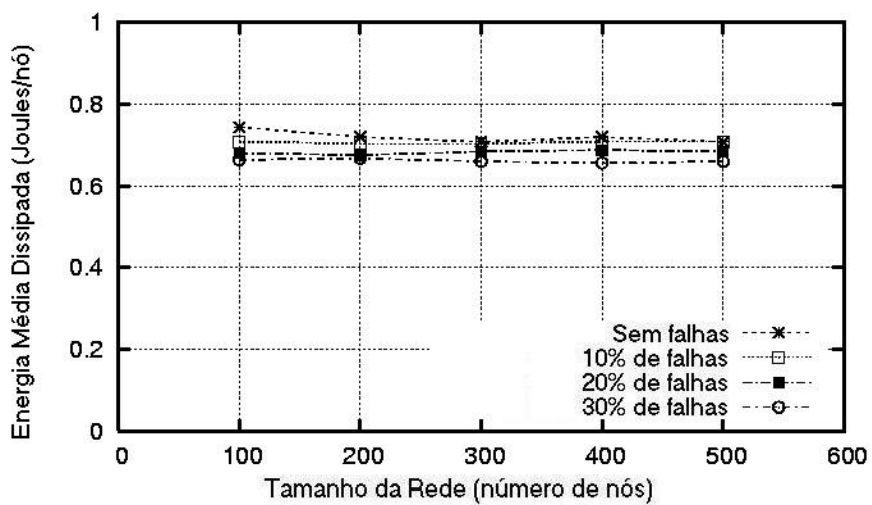


Figura 4.14. Gráfico da energia média dissipada por nó.

A figura 4.15 mostra uma comparação da energia média dissipada pelo DD e pelo PEQ. Como um grande número de nós permanece em um estado ocioso, a dissipação de energia não varia muito. Era de se esperar que o algoritmo consumiria mais energia encontrando caminhos alternativos devido às falhas dos nós, porém, o fato de que o módulo de rádio no estado ocioso consome praticamente a mesma quantidade de energia que no estado de recebimento de transmissões, o tempo ocioso de utilização de energia absolutamente domina todas as simulações.

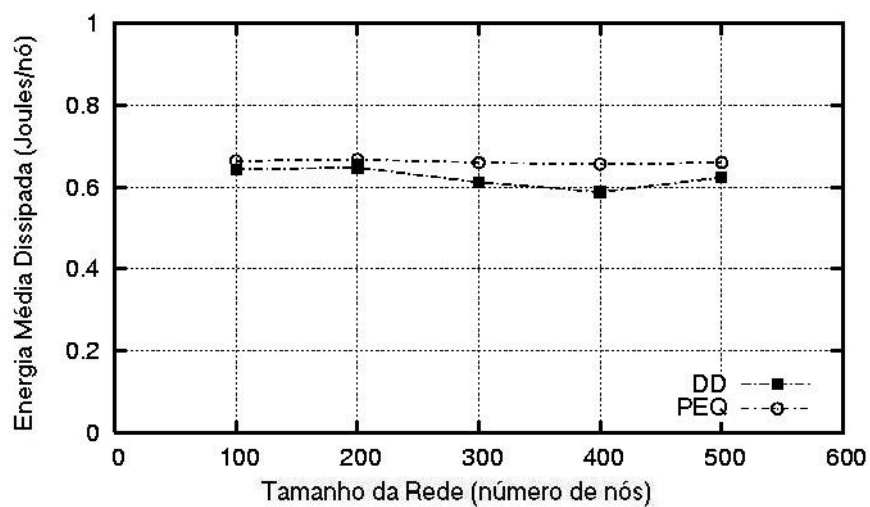


Figura 4.15. Gráfico de comparação da energia dissipada (30% de falhas).

Todos os experimentos reportados acima foram conduzidos com uma taxa de envio de dados fixa de 10 pacotes por segundo. Entretanto, é importante saber como uma taxa de envio de pacotes mais alta pode influenciar no atraso médio e na taxa de sucesso de entrega de pacotes. Portanto, foram realizados experimentos com taxas de 5, 10 e 15 pacotes por segundo. A figura 4.16 mostra que o aumento na taxa de pacotes afetou significativamente a latência observada, especialmente quando 15 ou mais pacotes por segundo foram gerados. Obviamente, uma taxa de 15 pacotes por segundo gera muito mais tráfego, conseqüentemente, colisões e perdas de pacotes são mais freqüentes, como mostra a figura 4.17, onde a taxa de sucesso na entrega de pacotes diminui quando mais pacotes por segundo são gerados.

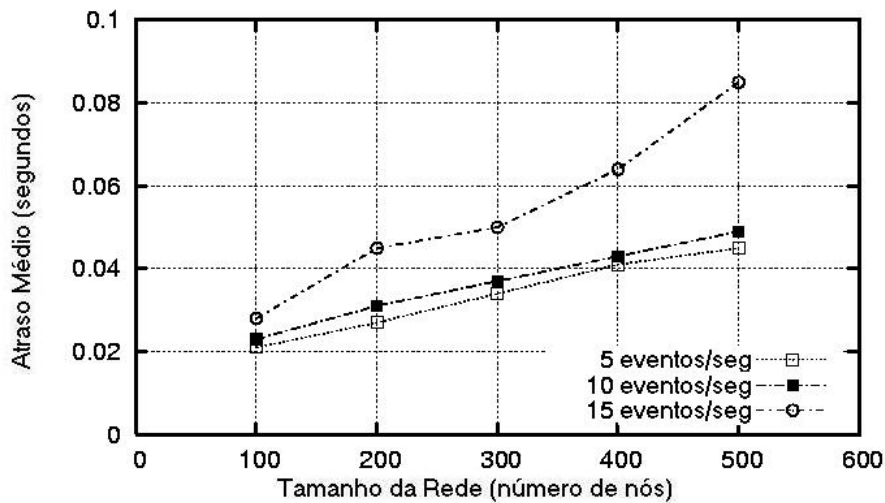


Figura 4.16. Gráfico do atraso médio com diferentes taxas de envio de pacotes.

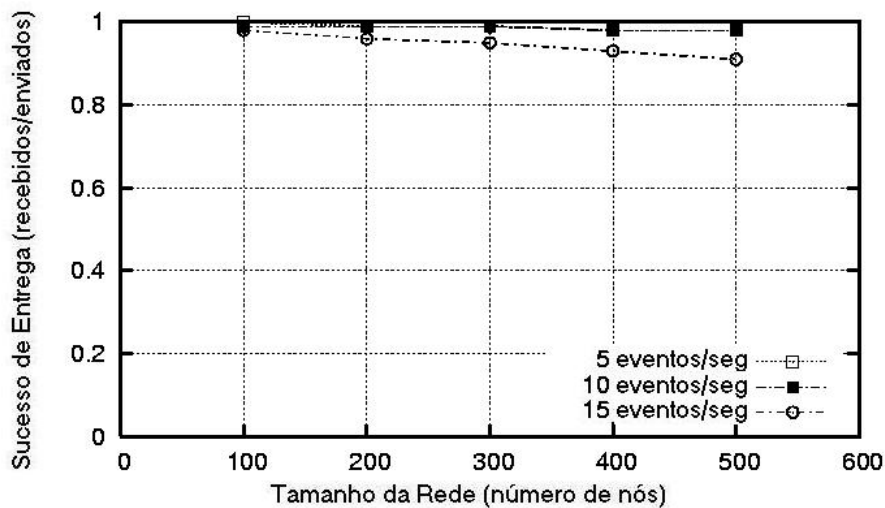


Figura 4.17. Gráfico da taxa de sucesso na entrega de pacotes.

#### 4.4 HPEQ - Uma Solução Hierárquica para reduzir o tráfego na rede

Como citado anteriormente, as redes de sensores sem fio podem conter centenas ou milhares de nós sensores. Deste modo, uma rede de sensores contém muitos dados que eventualmente precisam ser transmitidos para um centro de controle ou *sink*. Devido a esta grande quantidade de dados, surge a necessidade de métodos para agregação de dados [EST 99] [ELS 01][EST 00]. Além de ajudar a diminuir o tráfego de dados na rede, o que leva a uma diminuição da latência, a agregação de dados pode combinar diversas amostras de

dados e produzir um sinal mais preciso, mantendo os valores mais comuns e reduzindo os ruídos [CHO 00].

O algoritmo do PEQ, descrito na seção 4.1, considera apenas o melhor caminho ao entregar dados para o *sink*. Embora minimize a latência, esta abordagem acaba gerando muito tráfego de informações e, conseqüentemente, um consumo desnecessário de energia, pois todo nó da rede precisa encaminhar seus dados ao *sink*. Para solucionar este problema, é proposta uma evolução do PEQ chamada *Hierarchycal PEQ* (HPEQ).

Resumidamente, o HPEQ utiliza uma abordagem de *clustering* através da seleção aleatória de nós *agregadores* que formarão *clusters* de nós sensores. Os nós pertencentes a um cluster entregam seus dados ao agregador desse cluster que, por sua vez, encaminha os dados, agregados ou não, ao *sink*. Os objetivos do HPEQ são: distribuir uniformemente o consumo de energia, *balancear* a carga entre os nós sensores, reduzir a latência, e diminuir o tráfego em uma rede de sensores, aumentando assim seu tempo de vida. O balanceamento de carga e consumo é realizado através da alternância aleatória da função de agregador entre os nós da rede. Dessa forma, a função de agregador, que consome mais energia que um nó sensor não-agregador, não é fixa, evitando o desgaste prematuro dos nós escolhidos como agregadores.

O algoritmo do HPEQ é realizado em cinco etapas: configuração inicial da rede, seleção aleatória dos agregadores, formação dos clusters, transmissão dos dados para o agregador, e transmissão dos agregadores ao *sink*. Estas etapas podem se repetir quando múltiplos níveis de hierarquia são especificados na configuração. Cada etapa do algoritmo é detalhada a seguir.

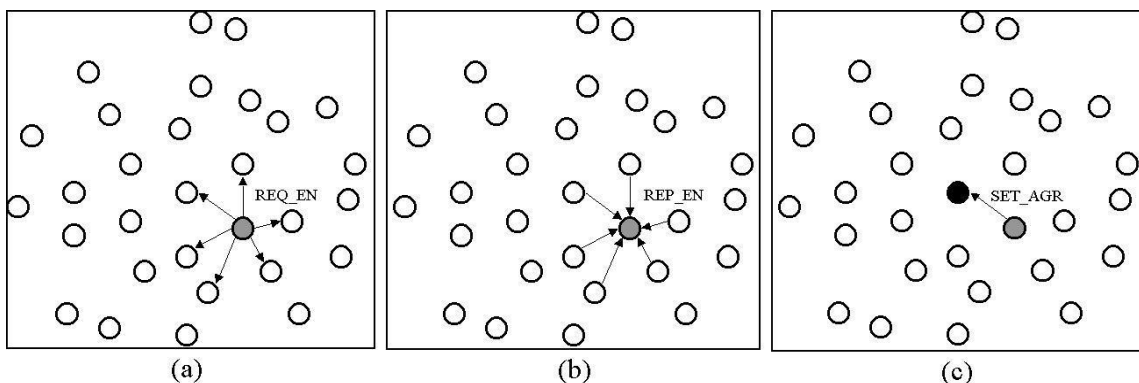
#### 4.4.1 Configuração Inicial da Rede

Esta etapa é idêntica à configuração inicial do PEQ, onde o *sink* inicia uma difusão pela rede para construir uma árvore baseada em níveis de *hops*, que representa os melhores caminhos entre cada nó da rede e o *sink*. A única diferença é a inclusão de um campo no cabeçalho do pacote que informa a porcentagem desejada de nós que se tornarão

agregadores. A difusão é realizada através de um *broadcast restrito*, ou seja, cada nó processa o pacote recebido para determinar se precisa ou não retransmiti-lo.

#### 4.4.2 Seleção Aleatória dos Agregadores

O esquema de escolha de agregadores do HPEQ baseia-se na idéia apresentada no LEACH [HEI 00a] que, originalmente, é realizada da seguinte forma: após receber a configuração inicial, qualquer nó da rede pode se tornar um agregador a qualquer momento com uma certa probabilidade. Por exemplo, se a porcentagem desejada de agregadores for de 5%, a probabilidade de um nó se tornar um agregador será  $p=0.05$ . Para isto, cada nó gera um número aleatório entre 0 e 1 e, se este número for menor que a probabilidade  $p$ , então o nó se torna um agregador. A idéia apresentada no LEACH não leva em consideração a energia de cada nó, portanto um nó sensor com baixa energia residual pode ser designado como agregador. Para minimizar este problema, o algoritmo do HPEQ proposto neste trabalho busca utilizar como agregadores os nós que apresentem maiores níveis de energia residual. A figura 4.18 exemplifica o esquema de seleção dos agregadores.



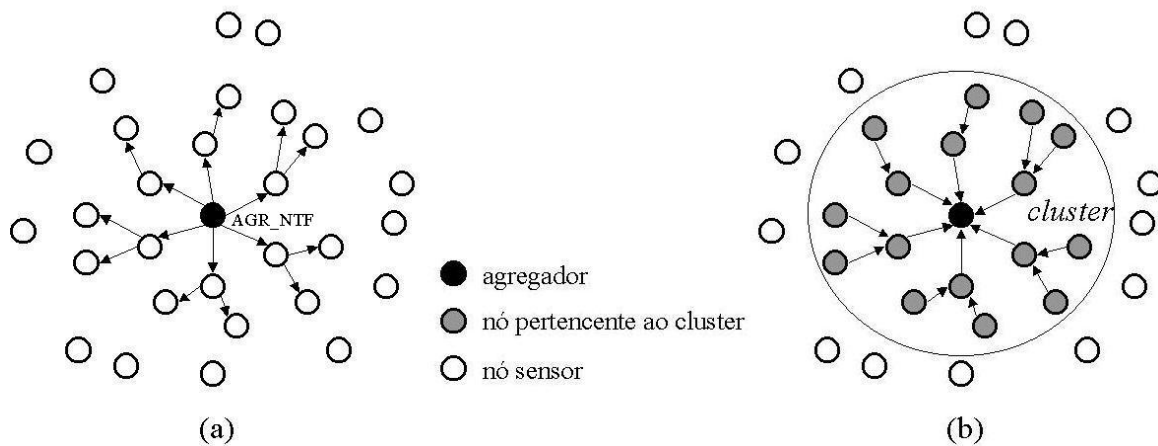
**Figura 4.18.** (a) O nó gerou um número menor que  $p$  e enviou REQ\_EN. (b) Os nós vizinhos recebem a mensagem e respondem REP\_EN. (c) O nó escolhe o vizinho com maior energia e o designa como agregador enviando SET\_AGR.

Assim como no LEACH, cada nó gera um número aleatório e o compara com o valor da probabilidade  $p$ . Porém, se o número gerado for menor que  $p$ , o nó não se torna um agregador, mas transmite uma mensagem REQ\_EN para seus vizinhos imediatos requisitando os valores de suas energias residuais. Cada vizinho responde transmitindo um

pacote REP\_EN contendo seu identificador e sua energia. Após receber os dados dos vizinhos, o nó compara os níveis de energia, inclusive o próprio, e designa o nó com maior energia como sendo o novo agregador, enviando uma mensagem SET\_AGR. Um nó exerce a função de agregador durante um tempo determinado, que é iniciado na configuração inicial da rede. Quando este tempo terminar, uma nova rodada de seleção de agregadores é iniciada e todo o algoritmo é executado novamente.

### 4.4.3 Formação dos *Clusters*

Após sua seleção, cada nó agregador precisa notificar sua nova função para os nós da vizinhança para a formação de grupos. O objetivo desses grupos (ou *clusters*) é que todos os nós pertencentes a um *cluster* transmitam seus dados coletados apenas para o agregador correspondente. A formação dos *clusters* é realizada através da propagação (*broadcast*) de uma mensagem de notificação SET\_AGR, que é exatamente o algoritmo de configuração inicial do PEQ, tratado na subseção 4.3.2, porém com um tempo de vida limitado (*tll*). Ao receber SET\_AGR, cada nó armazena em sua tabela de rotas o ID do nó que transmitiu, para configurar o caminho inverso pelo qual seus dados serão transmitidos ao agregador. O nó também verifica o valor do *tll* e, caso este valor seja maior que 0, o nó subtrai este valor em uma unidade e repassa SET\_AGR. Caso contrário, se o *tll* alcançou seu limite, o nó não repassa a mensagem. O uso do *tll* se faz necessário para limitar o tamanho dos *clusters*. Um nó pode receber SET\_AGR de mais de um nó agregador, que neste caso é escolhido o agregador que estiver mais próximo. Para saber qual agregador é o mais próximo, basta comparar o *tll* de cada mensagem SET\_AGR. Um *tll* maior indica uma maior proximidade com o agregador. Pode acontecer um empate na escolha do agregador mais próximo. Neste caso, o nó escolhe o agregador que transmitiu primeiro. A figura 4.19(a) mostra um agregador notificando sua vizinhança para formar seu *cluster*, e a figura 4.19(b) mostra o *cluster* e os caminhos formados entre os nós e o agregador. Neste exemplo foi utilizado um  $tll = 2$ . Cada agregador pode executar alguma função de agregação ou fusão de dados para diminuir o tráfego na rede.



**Figura 4.19.** (a) O agregador envia AGR\_NTF para seus vizinhos. (b) Formação do *cluster* e configuração dos caminhos para o agregador.

#### 4.4.4 Transmissão de Dados para os Agregadores

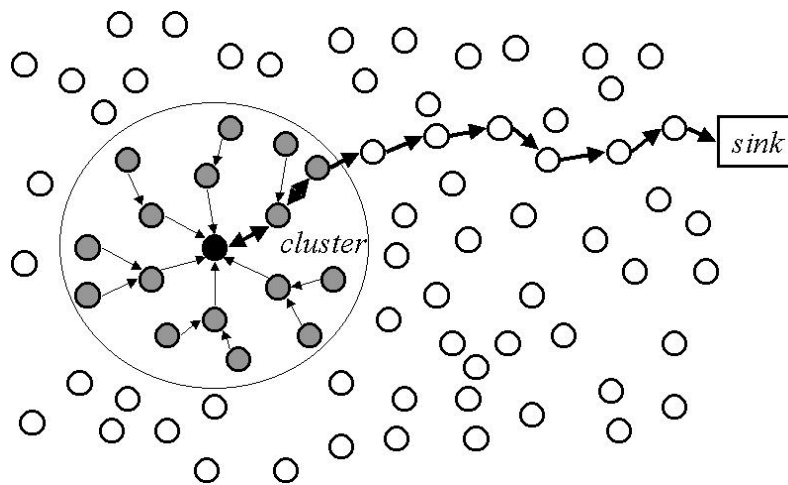
Quando um nó sensor captura alguma informação do ambiente físico, os dados devem ser transmitidos ao agregador do *cluster* do qual o nó faz parte. O algoritmo usado para enviar dados dos nós ao agregador é o mesmo utilizado pelo PEQ para encaminhar dados dos nós ao sink, detalhado na subseção 4.1.4. Pode-se pensar nesta etapa como o agregador do HPEQ sendo o *sink* do PEQ. O HPEQ também herda o mecanismo de reparo de caminhos do PEQ, tanto para a comunicação dos nós sensores com os agregadores quanto para a comunicação entre os nós com o *sink*.

#### 4.4.5 Transmissão de Dados para o *Sink*

Após receber os dados dos nós sensores, o agregador precisa encaminhar seus dados para o sink. Diferentemente do LEACH, que usa a comunicação direta entre os agregadores (*cluster-heads*) e o *sink*, o HPEQ utiliza uma comunicação *multihop* para encaminhar dados ao sink. Assim, o HPEQ supera uma importante limitação do LEACH, que é a expansibilidade [AKK 04]. No LEACH, como a comunicação entre os *cluster-heads* e o *sink* é direta, os *cluster-heads* devem aumentar sua potência de comunicação para transmitir a distâncias maiores. Se a rede for muito extensa, apenas os *cluster-heads* mais próximos ao *sink* conseguirão alcançá-lo. Os mais afastados serão incapazes de transmitir diretamente,

devido às limitações do hardware de comunicação e bateria dos nós sensores. Outras soluções [YOU 02][CER 01] utilizam nós especiais, com maior capacidade de comunicação e energia, para atuarem como agregadores. Tais soluções não só aumentam o custo da rede como necessitam de cuidados especiais na distribuição dos nós agregadores que, aliás, permanecem fixos durante o tempo de vida da rede, ocasionando um balanceamento de carga ineficiente.

Para garantir a expansibilidade do sistema, foi escolhida a comunicação *multihop* para o HPEQ, que funciona em uma rede de sensores homogênea, empregando sensores de baixo custo e que se comunicam a curtas distâncias e de forma colaborativa. Dessa forma, um agregador encaminha seus dados ao *sink* pela rota mais curta, formada por vários nós, que foi encontrada durante a etapa de configuração inicial da rede. A configuração inicial monta uma malha de nós sensores, com caminhos entre cada nó e o *sink*. A figura 4.20 mostra um agregador utilizando os nós dessa malha de sensores para enviar seus dados ao *sink*.



**Figura 4.20.** Comunicação *multihop* entre um agregador e *sink*.

É possível que alguns nós sensores não pertençam a um *cluster*. Caso isso ocorra, os nós sem *cluster* podem utilizar as rotas atribuídas durante a configuração inicial da rede para entregar os dados ao *sink*. Para minimizar o número de nós sem *cluster*, basta atribuir um valor maior para o *tll* dos pacotes de notificação SET\_AGR dos agregadores.



Entretanto, o número de mensagens trocadas para a formação dos *clusters* será maior e, conseqüentemente, o gasto de energia também aumentará. É necessário, através de simulações e/ou análises matemáticas, encontrar valores ótimos para o *ttl* e a probabilidade  $p$  de um nó se tornar um agregador que minimizem o gasto de energia, aumentando o tempo de vida da rede, e ao mesmo tempo garantindo a entrega rápida dos dados.

## 4.5 Considerações Finais

A implementação e simulação do protocolo PEQ levantaram questões importantes para o desenvolvimento do projeto. O tráfego gerado quando muitos nós sensores enviam pacotes para o *sink* foi considerado como o grande problema a ser resolvido. Outro problema é o alto consumo de energia nos nós próximos ao *sink* devido ao tráfego intenso que ocorre nessa região.

O protocolo HPEQ está em fase de implementação e simulação e, dependendo dos resultados obtidos com a simulação, novas camadas hierárquicas de agregadores podem ser formadas, bastando para isso repetir o algoritmo para cada camada, e configurando a rede com a porcentagem de agregadores desejada em cada nível hierárquico.

## 5 Conclusões

---

### 5.1 Considerações Iniciais

Este capítulo descreve as contribuições geradas com a realização deste trabalho, bem como trabalhos futuros e conclusões finais.

### 5.2 Contribuições Geradas

A especificação e implementação do protocolo PEQ, e sua evolução, o HPEQ, para redes de sensores sem fio geraram contribuições relevantes para a continuidade dos trabalhos no LRVNet, na medida em que deverão servir como protocolo base para o desenvolvimento de aplicações em redes de sensores. Além disso, o estudo do simulador de redes ns-2 e a implementação e simulação do protocolo, por serem pioneiros no LRVNet, geraram uma base de conhecimentos que facilitará futuras avaliações de desempenho e validações de protocolos e aplicações para redes de sensores sem fio.

#### 5.2.1 Artigos Publicados

As publicações obtidas até o momento foram:

- A. Boukerche, Richard Werner Nelem Pazzi, Regina B. de Araujo. ***“A Fast and Reliable Protocol for Wireless Sensor Networks in Critical Conditions Monitoring Applications”***, in The 7<sup>th</sup> ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04). Venice, Italy, October 4-6, 2004.
- Azzedine Boukerche, Richard Werner Nelem Pazzi, Regina B. Araujo: ***“A Novel Fault Tolerant and Energy-Aware Based Algorithm for Wireless Sensor Networks”***, in ALGOSENSORS 2004, Turku, Finland, July 16, 2004: 137-146.
- Richard Werner Nelem Pazzi, Regina B. de Araujo, Azzedine Boukerche. ***“Um Algoritmo Tolerante a Falhas e de Baixa Latência para Redes de Sensores sem Fio”***, no Simpósio Brasileiro de Telecomunicações (SBT2004). Belém, Brasil, 6-9 de Setembro, 2004.

- Azzedine Boukerche, Richard Werner Nelem Pazzi, Regina B. Araujo: “*Wireless Sensor Network Protocols for Critical Condition Monitoring*” in I2TS’2004, Sao Carlos, Brazil, December 6-9, 2004.

### 5.3 Trabalhos Futuros

Como continuidade ao trabalho aqui iniciado, as seguintes atividades ainda deverão ser realizadas:

1. Implementação e simulação completa do protocolo HPEQ, bem como sua avaliação perante o PEQ e outros protocolos para redes de sensores;
2. Incorporação do atendimento do requisito de Qualidade de Serviço (QoS) ao protocolo HPEQ;
3. Incorporação do atendimento do requisito de segurança ao protocolo HPEQ;
4. Implementação de camada de aplicação na rede de sensores para tratamento de controle (interpretação de eventos, ativação de atuadores etc).

### 5.4 Conclusões Finais

As redes de sensores sem fio estão cada vez mais sendo usadas para o monitoramento contínuo, para a detecção de eventos, localização, etc, nas mais diversas áreas de aplicação, como saúde, transporte, militar, combate a incêndio, e domiciliar. Duas das aplicações mais apelativas são supervisão de segurança e monitoramento de condições críticas. As redes de sensores estão sendo aplicadas para o monitoramento de ambientes físicos sujeitos a condições críticas como fogo, vazamento de gases tóxicos e explosões. Um grande desafio para essas redes de sensores é oferecer um canal rápido, confiável e tolerante a falhas para a difusão de dados em um ambiente hostil, e que satisfaça os requisitos dos diferentes cenários de aplicação, conhecidos por: baseado em consulta, dirigido a eventos, e periódico.

Este trabalho apresentou o PEQ, um protocolo de roteamento para redes de sensores que oferece: baixa latência para entrega de pacotes, reconfiguração dinâmica de caminhos quebrados, e alta confiabilidade de entrega de pacotes com baixa dissipação de energia. A

baixa latência é alcançada através do uso dos caminhos mais curtos até o *sink*. Subscrições rápidas de novos interesses são promovidas pelo conceito de entrega dirigida de eventos, no qual novas subscrições para uma região de sensores são rapidamente entregues através do caminho inverso usado para entrega de dados ao *sink*, ao invés de inundar a rede com as subscrições. A tolerância a falhas é acionada localmente pelos nós, ou seja, um nó sensor pode detectar a falha em um nó vizinho e iniciar um algoritmo de escolha de outro nó destinatário. Importantes métricas foram avaliadas através da simulação em um conceituado simulador de redes, o ns-2 [NS2 04], e comparadas ao paradigma *Directed Diffusion* [INT 00][INT 03]. Nos experimentos, o desempenho do PEQ foi superior ao *Directed Diffusion*, apresentando menores latências e melhor confiabilidade na entrega de eventos, tornando-se um bom candidato para satisfazer os rigorosos requisitos da entrega de pacotes em aplicações de monitoramento de situações de emergência, bem como nas outras áreas de aplicação das redes de sensores.

Foi proposta também uma evolução do PEQ, chamada HPEQ, que é um protocolo de roteamento que utiliza níveis hierárquicos entre os nós para a entrega dos pacotes. O HPEQ, que se encontra em fase de implementação, propõe-se a minimizar o tráfego na rede através da técnica conhecida como agregação de dados, tornar a rede mais expansível, com o uso de nós de baixo custo, e reduzir a latência na entrega de pacotes.

## 6 Referências Bibliográficas

---

- [ABO 00] G.D. Abowd, J.P.G. Sterbenz, Final report on the interagency workshop on research issues for smart environments, *IEEE Personal Communications* (October 2000) 36–40.
- [AKK 03] K. Akkaya and M. Younis, “An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks,” in the *Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003)*, Providence, Rhode Island, May 2003.
- [AKK 04] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” to appear in the *Journal of Ad Hoc Networks*.
- [AKK 04a] K. Akkaya and M. Younis, “Energy and QoS aware routing in wireless sensor networks”, in special issue of *Journal of Cluster Computing on Ad Hoc Networks* (to appear).
- [AKY 02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.
- [ALE 01] <http://www.alertsystems.org>
- [BAU 00] P. Bauer, M. Sichitiu, R. Istepanian, K. Premaratne, The mobile patient: wireless distributed sensor networks for patient monitoring and care, *Proceedings 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine*, 2000, pp. 17–21.
- [BLU 04] <http://www.bluetooth.org>
- [BOU 03] Boukerche, A. Cheng, X. And Linus, J. *Energy-Aware Data-Centric Routing in Microsensor Networks*. In *MSWiM'03*, September 19, 2003, San Diego, California, USA. (2003).
- [BOU 04] Boukerche, A. and I. Nikolettseas, "Protocols for Data Propagation in Wireless Sensor Networks", Chapter 2, in *Wireless Communications systems and networks*, Edt. M. Guizani, Kluwer Publ., pp. 23-51, 2004.
- [BRA 02] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in the *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA)*, Atlanta, GA, October 2002.
- [BUC 98] A. Buczak and V. Jamalabad, "Self-organization of a Heterogeneous Sensor Network by Genetic Algorithms," *Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et. (eds.), Vol. 8, pp. 259-264, ASME Press, New York, 1998.
- [BUL 01] N. Bulusu, D. Estrin, L. Girod, J. Heidemann, Scalable coordination for wireless sensor networks: self-organizing localization systems, *International*

- Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK, July 2001.
- [CER 01] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, Habitat monitoring: application driver for wireless communications technology, ACM SIGCOMM'2000, Costa Rica, April 2001.
- [CHA 99] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, A. Wang, Design considerations for distributed micro-sensor systems, Proceedings of the IEEE 1999 Custom Integrated Circuits Conference, San Diego, CA, May 1999, pp. 279–286.
- [CHI 01] C. Chien, I. Elgorriaga, C. McConaghy, Low-power directsequence spread-spectrum modem architecture for distributed wireless sensor networks, ISLPED'01, Huntington Beach, California, August 2001.
- [CHO 00] S. Cho, A. Chandrakasan, Energy-efficient protocols for low duty cycle wireless microsensor, Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI Vol. 2 (2000), p. 10.
- [CHT 02] Chatzigiannakis, I., Nikolettseas, S. and Spirakis, P. *A Comparative Study of Protocols for Efficient Data Propagation in Smart Dust Networks*. In Proc. 2nd ACM Workshop on Principles of Mobile Computing – POMC'2002 (2002).
- [CHU 02] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks," *The International Journal of High Performance Computing Applications*, Vol. 16, No. 3, August 2002.
- [ELS 01] J. Elson, D. Estrin, Random, ephemeral transaction identifiers in dynamic sensor networks, Proceedings 21st International Conference on Distributed Computing Systems, Mesa, AZ, April 2001, pp. 459–468. October 2000, pp. 60–61.
- [ESS 00] I.A. Essa, Ubiquitous sensing for smart and aware environments, IEEE Personal Communications (October 2000) 47–49.
- [EST 00] D. Estrin, R. Govindan, J. Heidemann, Embedding the Internet, Communication ACM 43 (2000) 38–41.
- [EST 01] D. Estrin, L. Girod, G. Pottie, M. Srivastava, Instrumenting the world with wireless sensor networks, International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), Salt Lake City, Utah, May 2001.
- [EST 02] D. Estrin, et. al., <http://nesl.ee.ucla.edu/tutorials/mobicom02>
- [EST 99] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: scalable coordination in sensor networks, ACM MobiCom'99, Washington, USA, 1999, pp. 263–270.
- [EUG 00] Eugster, P. T., Guerraoui R. and Sventek, J. *Distributed Asynchronous Collections: Abstractions for Publish/Subscribe Interaction*. Elisa Bertino

- (Ed.): ECOOP 2000, LNCS 1850, pp. 252-276. Springer-Verlag Berlin Heidelberg (2000).
- [EUG 03]** Eugster, P. T., Felber, P., Guerraoui, R. and Kermarrec, A. *The many faces of publish/subscribe*. ACM Comput. Surv. 35(2): 114-131 (2003).
- [GIR 01]** L. Girod, D. Estrin, Robust range estimation using acoustic and multimodal sensing, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001), Maui, Hawaii, October 2001.
- [HAL 01]** B. Halweil, Study finds modern farming is costly, World Watch 14 (1) (2001) 9–10.
- [HED 02]** W. Ye, J. Heidemann and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", in the *Proceedings of IEEE Infocom 2002*, New York, NY, June 2002.
- [HEI 00]** W. R. Heinzelman, et al., "Energy-Scalable algorithms and protocols for Wireless Sensor Networks", in the *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00)*, Istanbul, Turkey, June 2000.
- [HEI 00a]** W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," in the *Proceeding of the Hawaii International Conference System Sciences*, Hawaii, January 2000.
- [HEI 99]** W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in the *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, August 1999.
- [HER 00]** C. Herring, S. Kaplan, Component-based software systems for smart environments, IEEE Personal Communications, October 2000, pp. 60–61.
- [HET 03]** T. He et al., "SPEED: A stateless protocol for real-time communication in sensor networks," in the *Proceedings of International Conference on Distributed Computing Systems*, Providence, RI, May 2003.
- [HOB 00]** G. Hoblos, M. Staroswiecki, A. Aitouche, Optimal design of fault tolerant sensor networks, IEEE International Conference on Control Applications, Anchorage, AK, September 2000, pp. 467–472.
- [IMI 99]** T. Imielinski, S. Goel, DataSpace: querying and monitoring deeply networked collections in physical space, ACM International Workshop on Data Engineering for Wireless and Mobile Access MobiDE 1999, Seattle, Washington, 1999, pp. 44–51.
- [INT 00]** C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the *Proceedings of the 6th Annual ACM/IEEE International Conference on*

- Mobile Computing and Networking (MobiCom'00)*, Boston, MA, August 2000.
- [INT 03] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed Diffusion for Wireless Sensor Networking. *IEEE Trans. on Networking*, February 2003.
- [JOH 96] D. B Johnson et al., "Dynamic Source Routing in Ad Hoc Wireless Networks", in *Mobile Computing*, edited by Tomas Imielinski and Hank Korth, Kluwer Academic Publishers, ISBN: 0792396979, 1996, Chapter 5, pages 153-181.
- [KAH 99] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: mobile networking for smart dust, *Proceedings of the ACM MobiCom'99*, Washington, USA, 1999, pp. 271–278.
- [KAR 00] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless sensor networks," in the *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, Boston, MA, August 2000.
- [LIC 97] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal on Selected areas in Communications*, Vol. 15, No. 7, September 1997.
- [LIL 01] L. Li and J. Y Halpern, "Minimum energy mobile wireless networks revisited," in the *Proceedings of IEEE International Conference on Communications (ICC'01)*, Helsinki, Finland, June 2001.
- [LIN 01] S. Lindsey, C. S. Raghavendra and K. Sivalingam, "Data Gathering in Sensor Networks using the Energy\*Delay Metric", in the *Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.
- [LIN 02] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GATHERing in Sensor Information Systems," in the *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 2002.
- [MAN 01] A. Manjeshwar and D. P. Agrawal, "TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks," in the *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.
- [MAN 02] A. Manjeshwar and D. P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," in the *Proceedings of the 2<sup>nd</sup> International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing*, Ft. Lauderdale, FL, April 2002.



- [MIN 00] R. Min, et al., "An Architecture for a power aware distributed microsensor node", in the *Proceedings of the IEEE Workshop on signal processing systems (SIPS'00)*, October 2000.
- [MIN 01] Min, R., Bhardwaj, M., Cho, S., Sinha, A., Shih, E., Wang, A. and Chandrakasan, A. *Low-Power Wireless Sensor Networks*, VLSI Design 2001, January 2001.
- [NAT 02] B. Nath and D. Niculescu, "Routing on a curve," in HOTNETS 1, Princeton, NJ, October 2002.
- [NOU 00] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, T. Porcheron, Monitoring behavior in home using a smart fall sensor, IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology, October 2000, pp. 607–610.
- [NS2 04] The Network Simulator ns-2. [www.isi.edu/nsman/ns](http://www.isi.edu/nsman/ns).
- [OGA 98] M. Ogawa et al., Fully automated biosignal acquisition in daily routine through 1 month, International Conference on IEEE-EMBS, Hong Kong, 1998, pp. 1947–1950.
- [PEA 01] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar, SPINS: security protocols for sensor networks, Proceedings of ACM MobiCom'01, Rome, Italy, 2001, pp. 189– 199.
- [PER 00] C. Perkins, Ad Hoc Networks, Addison-Wesley, Reading, MA, 2000.
- [PER 02] C. Perkins et al., "Ad hoc On-Demand Distance Vector (AODV) Routing," *Internet Draft draftietf-manet-aodv-11.txt*, June 2002.
- [PET 01] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, V.Z. Groza, Sensor-based information appliances, IEEE Instrumentation and Measurement Magazine (December 2001) 31–35.
- [POR 00] A. Porret, T. Melly, C.C. Enz, E.A. Vittoz, A low-power low-voltage transceiver architecture suitable for wireless distributed sensors network, IEEE International Symposium on Circuits and Systems'00, Geneva, Vol. 1, 2000, pp. 56–59.
- [POT 00] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors, Communications of the ACM 43 (5) (2000) 551– 558.
- [RAB 00] J.M. Rabaey, M.J. Ammer, J.L. da Silva Jr., D. Patel, S. Roundy, PicoRadio supports ad hoc ultra-low power wireless networking, IEEE Computer Magazine (2000) 42–48.
- [RAP 96] T. Rappaport, Wireless Communications: Principles and Practice, Prentice-Hall, Englewood Clis, NJ, 1996.
- [ROD 99] V. Rodoplu and T.H. Ming, "Minimum energy mobile wireless networks," *IEEE Journal of Selected Areas in Communications*, Vol. 17, No. 8, pp. 1333-1344, 1999.

- [SAD 03] N. Sadagopan et al., "The ACQUIRE mechanism for efficient querying in sensor networks," in the *Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska, May 2003*.
- [SAV 01] A. Savvides, C. Han, M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," *Proceedings of ACM MobiCom'01, Rome, Italy, July 2001*, pp. 166–179.
- [SCH 01] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks," in the *MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001*.
- [SHA 02] R. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", in the *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002*.
- [SHE 01] C. Shen, C. Srisathapornphat, C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Personal Communications*, August 2001, pp. 52–59.
- [SHI 01] E. Shih, et al., "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks", in the *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'01), Rome, Italy, July 2001*, pp. 272–286.
- [SIB 01] B. Sibbald, "Use computerized systems to cut adverse drug events: report," *CMAJ: Canadian Medical Association Journal* 164 (13) (2001) 1878, 1/2p, 1c.
- [SOH 00] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, Vol. 7, No. 5, pp. 16-27, October 2000.
- [SUB 00] L. Subramanian and R. H. Katz, "An Architecture for Building Self Configurable Systems," in the *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA, August 2000*.
- [TOS 04] <http://webs.cs.berkeley.edu/tos/index.html>.
- [TSE 01] Y. Tseng, S. Wu, C. Lin, J. Sheu, "A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks," *IEEE International Conference on Distributed Computing Systems, Mesa, AZ, April 2001*, pp. 419–424.
- [WAR 01] B. Warneke, B. Liebowitz, K.S.J. Pister, "Smart dust: communicating with a cubic-millimeter computer," *IEEE Computer* (January 2001) 2–9.
- [WIT 01] "With Glacier Park in Its Path, Fire Spreads to 40,000 Acres," *New York Times*, Vol. 150, Issue 51864, p. 24, 0p, 1 map, 4c, 9/2/2001.

- [WOO 01] A. Woo and D. Culler. "A Transmission Control Scheme for Media Access in Sensor Networks," in the *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'01)*, Rome, Italy, July 2001, pp. 221–235.
- [WUS 00] S. Wu, Y. Tseng, J. Sheu, Intelligent medium access for mobile ad hoc networks with busy tones and power control, *IEEE Journal on Selected Areas in Communications* (September 2000) 1647–1657.
- [XUY 01] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in the *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01)*, Rome, Italy, July 2001.
- [YAO 02] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," in *SIGMOD Record*, September 2002.
- [YOS 02] M. Youssef, M. Younis and K. Arisha, "A constrained shortest-path energy-aware routing algorithm for wireless sensor networks," in the *Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2002)*, Orlando, FL, March 2002.
- [YOU 02] M. Younis, M. Youssef and K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks", in the *Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002)*, Fort Worth, TX, October 2002.
- [YUY 01] Y. Yu, D. Estrin, and R. Govindan, "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," *UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023*, May 2001.