

Arthur Vinícius Maciel Dantas

***DESRO: um protocolo de roteamento com
gerenciamento dinâmico de energia para redes de
sensores sem fio***

Viçosa -MG - Brasil

2010

Arthur Vinícius Maciel Dantas

***DESRO: um protocolo de roteamento com
gerenciamento dinâmico de energia para redes de
sensores sem fio***

Dissertação apresentada à Universidade Federal de Viçosa como parte das exigências do programa de Pós-Graduação em Ciência da Computação para obtenção do título de *Magister Scientiae*.

Orientador: Carlos de Castro Goulart

Co-orientadores:

Mauro Nacif Rocha

Ricardo dos Santos Ferreira

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE INFORMÁTICA

Viçosa -MG - Brasil

2010

**Ficha catalográfica preparada pela Seção de Catalogação e
Classificação da Biblioteca Central da UFV**

T

D192d
2010

Dantas, Arthur Vinicius Maciel, 1984-

DESRO: um protocolo de roteamento com gerenciamento dinâmico de energia para redes de sensores sem fio / Arthur Vinicius Maciel Dantas. – Viçosa, MG, 2010.
xiii, 50f. : il. ; 29cm.

Orientador: Carlos de Castro Goulart.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 49-50.

1. Redes de computadores. 2. Protocolo de aplicação sem fio (Protocolo de rede de computadores). I. Universidade Federal de Viçosa. II. Título.

CDD 22.ed. 004.6

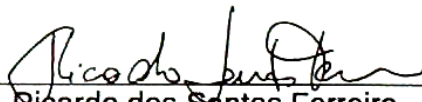
ARTHUR VINÍCIUS MACIEL DANTAS

**DESRO: um protocolo de roteamento com gerenciamento
dinâmico de energia para redes de sensores sem fio**

Dissertação apresentada à
Universidade Federal de Viçosa,
como parte das exigências do
Programa de Pós-Graduação em
Ciência da Computação, para
obtenção do título de *Magister
Scientiae*.

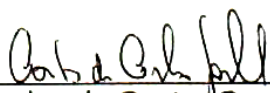
APROVADA: 25 de fevereiro de 2010.


Mauro Nacif Rocha
(Co-Orientador)


Ricardo dos Santos Ferreira
(Co-Orientador)


Vladimir Oliveira Di Iorio


Antônio Alfredo Ferreira Loureiro


Carlos de Castro Goulart
(Orientador)

*“[Eu] Pensava que nós seguíamos
caminhos já feitos, mas parece que não
os há. O nosso ir faz o caminho.”*

C.S. Lewis

*A Deus,
Aos meus pais, Azenilde e Antônio
Aos meus irmãos*

Dedico

AGRADECIMENTOS

Agradeço a Deus por tudo o que fez em minha vida e pelo que ainda fará; por ter me amado primeiro e ter estendido Sua mão, e O agradeço principalmente pelo que é.

Agradeço aos meus pais. Agradeço em especial à minha mãe, Azenilde, que nunca deixou de me incentivar mesmo diante das dificuldades e que sempre foi um porto seguro.

Aos meus irmãos agradeço pelas conversas e conselhos, auxílios e apoio nas mais diversas situações. Não posso deixar de agradecer-los também pelas brincadeiras durante as reuniões familiares.

Agradeço aos meus professores, Carlos, Mauro e Ricardo, e a todos que me auxiliaram durante o desenvolvimento deste trabalho. Agradeço também ao professor Loureiro pela participação na banca e por sugestões valiosas.

Agradeço aos meus amigos irmãos da Primeira Igreja Batista de Viçosa pelas brincadeiras, conselhos e bons momentos juntos, por terem me ensinado mais sobre Deus e pela comunhão.

A todos meus amigos, Raphael, Henrique, Douglas, Daniel, Juliana e Nívio dentre outros, que sempre torceram para que este trabalho fosse concluído.

Agradeço às grandes bandas que me ajudaram, com seu talento, a terminar um "trabalho movido a música": Switchfoot, Jars of Clay, Audio Adrenaline, As I Lay Dying, Red, Falling Up, etc.

Agradeço, por fim, a todos aqueles que contribuíram direta ou indiretamente para a realização deste trabalho.

*A Deus,
Aos meus pais, Azenilde e Antônio
Aos meus irmãos*

Dedico

AGRADECIMENTOS

Agradeço a Deus por tudo o que fez em minha vida e pelo que ainda fará; por ter me amado primeiro e ter estendido Sua mão, e O agradeço principalmente pelo que é. Em fevereiro de 1999 foi aceito no Centro Federal de Educação Tecnológica

(CEFET) Agradeço aos meus pais. Agradeço em especial à minha mãe, Azenilde, que nunca deixou de me incentivar mesmo diante das dificuldades e que sempre foi um porto seguro. Ingresso em Ciência da Computação em dezembro de 2006.

Aos meus irmãos agradeço pelas conversas e conselhos, auxílios e apoio nas mais diversas situações. Não posso deixar de agradecê-los também pelas brincadeiras durante as reuniões familiares. maio de 2010.

Agradeço aos meus professores, Carlos, Mauro e Ricardo, e a todos que me auxiliaram durante o desenvolvimento deste trabalho. Agradeço também ao professor Loureiro pela participação na banca e por sugestões valiosas.

Agradeço aos meus amigos-irmãos da Primeira Igreja Batista de Viçosa pelas brincadeiras, conselhos e bons momentos juntos, por terem me ensinado mais sobre Deus e pela comunhão.

A todos meus amigos, Raphael, Henrique, Douglas, Daniel, Juliana e Nívio dentre outros, que sempre torceram para que este trabalho fosse concluído.

Agradeço às grandes bandas que me ajudaram, com seu talento, a terminar um “trabalho movido a música”: Switchfoot, Jars of Clay, Audio Adrenaline, As I Lay Dying, Red, Falling Up, etc.

Agradeço, por fim, a todos aqueles que contribuíram direta ou indiretamente para a realização deste trabalho.

BIOGRAFIA

LISTA Arthur Vinícius Maciel Dantas, filho de Antônio Jacob Neto e Azenilde Maciel de Sousa, nasceu em 06 de Maio de 1984, em Eunápolis, Bahia.	ix
RESUM Em fevereiro de 1999 foi aceito no Centro Federal de Educação Tecnológica (CEFET) de Eunápolis, onde concluiu o ensino médio em dezembro de 2001.	xii
1. INT Em março de 2003, ingressou na Universidade Federal de Viçosa, concluindo o curso de Bacharelado em Ciência da Computação em dezembro de 2006.	2
Em Março de 2007, iniciou o mestrado no Programa de Pós-Graduação em Ciência da Computação, na Universidade Federal de Viçosa, submetendo-se a defesa de dissertação em 25 de fevereiro de 2010.	3
2. REDES DE SENSORES	5
2.1. Aplicações.....	5
2.2. Nó sensor.....	6
2.3. Arquitetura de rede de sensores sem fio.....	8
2.3.1. Introdução.....	8
2.3.2. Caracterização.....	8
2.3.3. Comunicação.....	9
2.4. Protocolos de roteamento.....	11
2.4.1. Tipos de protocolos.....	12
2.5. Simulações em redes de sensores sem fio.....	16
2.6. Conclusão.....	18
3. DESRO	19
3.1. DSR.....	20
3.1.1. Descoberta de rotas.....	20
3.1.2. Manutenção de rota.....	22
3.1.3. Funcionalidades adicionais.....	22
3.2. ESDSR.....	25
3.3. DESRO.....	27
3.4. Detalhes de codificação.....	30
4. AVALIAÇÃO DO DESRO	34
4.1. Metodologia.....	34
4.2. Simulação.....	34

SUMÁRIO

4.3. Resultados.....	16
4.3.1. Distribuição uniforme não-estocástica.....	17
LISTA DE FIGURAS E TABELAS.....	viii
LISTA DE ABREVIATURAS.....	ix
RESUMO.....	x
ABSTRACT.....	xii
1. INTRODUÇÃO.....	1
1.1. O problema e sua importância.....	2
1.2. Hipótese.....	3
1.3. Objetivos.....	3
1.4. Organização do texto.....	3
2. REDES DE SENSORES.....	5
2.1. Aplicações.....	5
2.2. Nó sensor.....	6
2.3. Arquitetura de rede de sensores sem fio.....	8
2.3.1. Introdução.....	8
2.3.2. Caracterização.....	8
2.3.3. Comunicação.....	9
2.4. Protocolos de roteamento.....	11
2.4.1. Tipos de protocolos.....	12
2.5. Simulações em redes de sensores sem fio.....	16
2.6. Conclusão.....	18
3. DESRO.....	19
3.1. DSR.....	20
3.1.1. Descoberta de rotas.....	20
3.1.2. Manutenção de rota.....	22
3.1.3. Funcionalidades adicionais.....	22
3.2. ESDSR.....	25
3.3. DESRO.....	27
3.4. Detalhes da codificação.....	30
4. AVALIAÇÃO DO DESRO.....	34
4.1. Metodologia.....	34
4.2. Simulações.....	34

4.3. Resultados.....	36
4.3.1. Distribuição uniforme não-aleatória.....	37
4.3.2. Distribuição aleatória uniforme.....	43
5. CONCLUSÕES E PERSPECTIVAS FUTURAS.....	47
8. REFERÊNCIAS BIBLIOGRÁFICAS.....	49
Figura 2.4. Pilha de protocolos.....	10
Figura 2.5. A e B transmitem o mesmo dado ao nó C.....	12
Figura 2.6. Funcionamento do SPIN.....	13
Figura 2.7. Funcionamento do <i>Directed Diffusion</i>	14
Figura 2.8. Hierarquia de clusters no TEEN.....	15
Figura 3.1. Nó A busca rota para o nó E.....	21
Figura 3.2. Nó atinge o limite de tentativas de transmissão.....	22
Figura 3.3. Envio de <i>Route Reply</i> usando cache de rotas.....	23
Figura 3.4. Possível duplicação de nó em um <i>Route Reply</i>	24
Figura 3.5. Roteamento no ESDSR.....	26
Figura 3.6. Pseudo-código do DESRO.....	29
Figura 3.7. Repasse de <i>Route Request</i>	32
Figura 4.1. Topologia uniforme não-aleatória.....	37
Figura 4.2. Taxa de entrega em cada simulação.....	38
Figura 4.3. Distribuição média de energia no DESRO.....	39
Figura 4.4. Distribuição média de energia no ESDSR.....	39
Figura 4.5. Áreas com maior consumo de energia.....	40
Figura 4.6. Desvio-padrão do consumo de energia nas simulações.....	40
Figura 4.7. Consumo de energia dos nós 44 e 45 no ESDSR.....	42
Figura 4.8. Consumo de energia dos nós 44 e 45 no DESRO.....	42
Figura 4.9. Topologia uniforme aleatória.....	43
Figura 4.10. Taxa de entrega em cada simulação.....	44
Figura 4.11. Nós capazes de transmitir diretamente ao servidor.....	45
Figura 4.12. Distribuição média de energia no ESDSR.....	45
Figura 4.13. Distribuição média de energia no DESRO.....	46
Tabela 3.1. Representação de parâmetros.....	30
Tabela 4.1. Parâmetros do CC2420.....	35
Tabela 4.2. Desvio-padrão e variância em cada simulação.....	41
Tabela 4.3. Desvio-padrão e variância em cada simulação.....	46

LISTA DE FIGURAS E TABELAS

Figura 2.1. Redes de sensores em um domicílio.	6
Figura 2.2. Arquitetura de um nó sensor.	7
Figura 2.3. Tipos de sorvedouros.	8
Figura 2.4. Pilha de protocolos.	10
Figura 2.5. A e B transmitem o mesmo dado ao nó C.	12
Figura 2.6. Funcionamento do SPIN.	13
Figura 2.7. Funcionamento do <i>Directed Diffusion</i> .	14
Figura 2.8. Hierarquia de clusters no TEEN.	15
Figura 3.1. Nó A busca rota para o nó E.	21
Figura 3.2. Nó atinge o limite de tentativas de transmissão.	22
Figura 3.3. Envio de <i>Route Reply</i> usando cache de rotas.	23
Figura 3.4. Possível duplicação de nó em um <i>Route Reply</i> .	24
Figura 3.5. Roteamento no ESDSR.	26
Figura 3.6. Pseudo-código do DESRO.	29
Figura 3.7. Repasse de <i>Route Request</i> .	32
Figura 4.1. Topologia uniforme não-aleatória.	37
Figura 4.2. Taxa de entrega em cada simulação.	38
Figura 4.3. Distribuição média de energia no DESRO.	39
Figura 4.4. Distribuição média de energia no ESDSR.	39
Figura 4.5. Área com maior consumo de energia.	40
Figura 4.6. Desvio-padrão do consumo de energia nas simulações.	40
Figura 4.7. Consumo de energia dos nós 44 e 48 no ESDSR.	42
Figura 4.8. Consumo de energia dos nós 44 e 48 no DESRO.	42
Figura 4.9. Topologia uniforme aleatória.	43
Figura 4.10. Taxa de entrega em cada simulação.	44
Figura 4.11. Nós capazes de transmitir diretamente ao sorvedouro.	45
Figura 4.12. Distribuição média de energia no ESDSR.	45
Figura 4.13. Distribuição média de energia no DESRO.	46
Tabela 3.1. Representação de parâmetros.	30
Tabela 4.1. Parâmetros do CC2420.	35
Tabela 4.2. Desvio-padrão e variância em cada simulação.	41
Tabela 4.3. Desvio-padrão e variância em cada simulação.	46

LISTA DE ABREVIATURAS

ACK - *Acknowledgment*

APL - *Academic Public License*

CTS - *Clear to Send*

DESRO - *Dynamic Energy Saving Routing*

DSR - *Dynamic Source Routing*

ESDSR - *Energy Saving Dynamic Source Routing*

GAF - *Geographic Adaptive Fidelity*

GHz - *Gigahertz*

GPS - *Global Positioning System*

IEEE - *Institute of Electrical and Eletronics Enginners*

LEACH - *Low-energy adaptive clustering hierachy*

MAC – *Medium Access Control*

MANET - *Mobile Adhoc Network*

MHz - *Megahertz*

PDA - *Personal Digital Assistant*

QoS - *Quality of Service*

RAM - *Random Access Memory*

RSSF - *Rede de sensores sem fio*

RSSI - *Received Signal Strength Indication*

RTS - *Request to Send*

S-MAC - *Sensor MAC*

SPIN - *Sensor Protocols for Information via Negotiation*

TEEN - *Threshold sensitive Energy Efficient sensor Network*

WLAN - *Wireless Local Area Network*

Resumo

Este trabalho apresenta o DESRO (Dynamic Energy Saving Routing), um protocolo de roteamento dinâmico baseado no ESDSR (Energy Saving Dynamic Source Routing). O DESRO é um protocolo reativo para uso em redes estáticas planas e foi construído com o objetivo de sanar deficiências encontradas no ESDSR, tornando a decisão de roteamento de dados sensorizados mais eficiente. Possui dois mecanismos que trabalham conjuntamente para a realização do roteamento de dados: descoberta de rota e manutenção de rota.

Na descoberta de rota, o DESRO, quando não possui um caminho até o sorvedouro, envia, em *broadcast*, uma mensagem de requisição de rotas. A mensagem de requisição é repassada até chegar ao sorvedouro. Ao receber uma requisição o sorvedouro envia, ao originador desta, uma mensagem de resposta contendo a rota descoberta, isto é, todos os nós pelos quais a mensagem de requisição passou.

A manutenção de rotas é realizada quando um nó não consegue repassar uma mensagem de dados. Ele assume que o *link* está quebrado e envia uma mensagem avisando da quebra de *link* ao originador da mensagem de dados. Se o originador possuir uma rota alternativa passa a usá-la, caso contrário inicia uma nova descoberta de rota.

A decisão de qual rota será utilizada é baseada nos valores de energia mínima estimada da rota e do seu número de saltos. A energia mínima estimada é calculada a partir da energia restante do nó e da potência usada na transmissão para um nó vizinho.

O DESRO estende as funcionalidades do ESDSR adicionando um eficiente gerenciamento de rotas capaz de balancear a carga de roteamento entre os nós. O objetivo do balanceamento de carga é melhorar a distribuição de energia pela rede aumentando sua vida útil, evitando a morte dos nós causada por esgotamento da bateria.

O gerenciamento de rotas é realizado com o uso de um temporizador e/ou contador de pacotes, criados quando da descoberta de uma rota. Quando o temporizador é zerado, é enviada uma mensagem de atualização da energia mínima estimada da rota e o temporizador é resetado para o valor inicial pré-estabelecido. A mensagem de atualização da energia mínima estimada pode ser também enviada quando o contador de pacotes, que é incrementado a cada novo pacote de dados criado, chegar ao limite pré-estabelecido, quando, então, será zerado.

O ESDSR e DESRO foram comparados através de simulações no Castalia, um simulador de eventos discretos para redes de sensores sem fio. Foram realizadas simulações de distribuição em grade e distribuição uniforme aleatória dos nós. Os resultados obtidos mostraram que o gerenciamento de rotas foi capaz de atrasar a morte dos nós e aumentar a taxa de entrega de pacotes de dados.

Abstract

This work presents DESRO (Dynamic Energy Saving Routing), a dynamic routing protocol based on ESDSR (Energy Saving Dynamic Source Routing). DESRO is an on-demand protocol to be used on plane static wireless sensor networks and it was built to solve some deficiencies found in ESDSR. DESRO has a more efficient data routing and two mechanisms which work together to perform routing: route discovery and route maintenance.

In route discovery, DESRO sends a route request broadcast message when the node wants to send a data packet and has no route to the sink. The route request message is forwarded until the sink receives it. When this happens, the sink sends a route reply message to the node that generate the route request message, containing the list os all intermediate nodes. These nodes are listed in the header of a route request message.

The route maintenance is performed when a node is unable to forward a data message. The node assumes that the link is broken and sends a warning message to the node that has sent the data message. The data message originator, then, looks for an alternative route to the sink node in its routing table. If this node can not find an alternative route them, it initiates the mechanism of route discovery.

The decision of which route will be used is based on the minimum expected life of route and on the number of hops. The minimum expected life is calculated using the value of the radio power transmission used to send a data packet to the neighbor node.

DESRO extends the features of ESDSR through the inclusion of a efficient route management system capable of performing a load balancing among the nodes. The aim of promoting the load balancing is to get a better energy distribution on the network to postpone the death of nodes due to the lack of energy.

The route management system is realized through a timer and/or a data packet counter, created when a route is discovered. When the timer reaches a limit, it is reset and a route management message is sent. The route management message can be sent when the data packet counter reaches the previous stablished limit also. When it happens the data packet counter is reset.

The ESDSR and DESRO was compared through simulations on Castalia, a wireless sensor network discret-event simulator. It was performed some simulations with grid and randomic uniform distributions. The results show that the route management system was able to postpone the death of nodes and to increase the data delivery rate.

1 *Introdução*

A tecnologia de comunicação sem fio (*wireless*) entre computadores tem crescido em popularidade, sendo utilizada na área comercial e residencial além do seu já conhecido uso pelos militares. Ela hoje se encontra nos celulares, dispositivos móveis, *notebooks* e em diversos aparelhos. Esse crescimento foi possível pela padronização da forma de acesso e comunicação entre os nós da rede. A popularidade de uso das redes sem fio, em especial a WLAN (*Wireless Local Area Network*) se deu graças aos esforços do IEEE (*Institute of Electrical and Electronics Engineers*) que, após anos de estudo, apresentou em 1997 o padrão 802.11. Este padrão especifica o funcionamento da camada de acesso ao meio (MAC – *Medium Access Control*) e da camada física das redes locais sem fio. O padrão 802.11 também especifica as faixas de frequências permitidas na comunicação entre os dispositivos (SARANGAPANI, 2007).

O princípio básico de funcionamento de uma rede sem fio é a transformação do sinal elétrico em sinal de rádio, sendo o ar o seu meio de comunicação. Isso permite que as redes sem fio apresentem vantagens em relação às redes comuns (cabeadas). A não necessidade de uma estrutura de fios e a mobilidade/portabilidade que ela permite são os maiores motivadores do seu uso.

Redes sem fio enfrentam problemas que não existem ou são facilmente contornáveis em redes cabeadas. A interferência no sinal é um grande problema enfrentado pelas redes sem fio. Diversos equipamentos (microondas, telefones sem fio, etc.) emitem ondas na mesma faixa de frequência ou em frequências próximas às utilizadas pelas redes. Além disso barreiras como paredes, árvores e metais afetam a qualidade do sinal de rádio da rede. A questão de segurança também é muito importante em redes sem fio, já que qualquer receptor na área de alcance poderá capturar as informações sendo transmitidas.

As redes sem fio podem ser divididas em duas categorias: rede infra-estruturada e rede não infra-estruturada. As redes infra-estruturadas não permitem a comunicação direta dos dispositivos da rede; para que haja comunicação entre os nós é necessário um centralizador (ponto de acesso) que será responsável pela comunicação. As redes sem infra-estrutura, também co-

nhecidas como redes *ad hoc*, se caracterizam pela comunicação direta entre os dispositivos da rede, não havendo necessidade de centralizadores na comunicação. Os dispositivos em uma rede sem infra-estrutura funcionam como estações e roteadores e a comunicação em redes sem infra-estrutura é ainda mais desafiadora que em redes estruturadas. A capacidade dos dispositivos de alterarem suas localizações mudando a topologia da rede torna necessária a utilização de protocolos capazes de se adaptar rapidamente às mudanças na rede.

Redes de sensores sem fio (RSSFs) são um tipo especial de redes *ad hoc*. Possuem como características principais o tamanho reduzido dos dispositivos, alta restrição de consumo de energia e baixa largura de banda.

1.1 O problema e sua importância

A constante evolução tecnológica, a miniaturização dos circuitos eletro-mecânicos e a melhoria da comunicação sem fio possibilitaram a criação de sensores cada vez menores e que hoje são utilizados em redes sem fio para os mais diversos fins. Surgiu assim um novo tipo de rede sem fio não estruturada, as redes de sensores sem fio (RSSFs).

Redes de sensores sem fio têm sido utilizadas com sucesso em diversas aplicações: sensoriamento externos (florestas, vulcões, etc.) e internos (fábricas, escritórios, etc.), e também no auxílio a diversas outras áreas de pesquisa como a biologia, onde o estudo de hábitos de animais podem ser executados com o mínimo de intervenções no ambiente. Estas redes já são também utilizadas em segurança pública e em ambientes de combate (LOUREIRO et al., 2003).

Para que as RSSFs venham a auxiliar as mais diversas áreas é importante que consigam fornecer dados confiáveis. É necessário, portanto, uma comunicação eficaz entre os nós da rede. Contudo, as RSSFs possuem grandes limitações devido à escassez de recursos, principalmente energético. Essa limitação se deve ao fato de que, em geral, os sensores possuem baterias limitadas e não há a possibilidade de recarga de energia. Assim, as RSSFs devem possuir uma comunicação não somente eficaz mas também eficiente. Para alcançar uma comunicação eficiente é necessário o uso de protocolos que priorizem as limitações enfrentadas por uma RSSF.

Os protocolos de roteamento para estas redes devem possuir a economia de energia como um de seus principais fatores de QoS (*Quality of Service*), juntamente com a entrega dos dados. Assim, os protocolos de roteamento comuns às redes sem fio padrão não são aplicáveis às RSSFs, que necessitam de protocolos específicos. Os protocolos de roteamento para RSSFs devem usar de maneira eficiente a energia da rede e devem possuir um mecanismo de adaptação

a mudanças na distribuição da energia na rede e a falhas.

A escolha do protocolo de roteamento a ser utilizado em uma RSSF é dependente das características da rede (quanto à sua configuração, sensoriamento e comunicação) e do seu objetivo (sensoriamento externo, interno, etc.). A escolha equivocada de protocolos de roteamento em RSSFs pode levar a uma utilização inadequada dos recursos da rede, acarretando a diminuição do seu tempo de vida e/ou diminuindo a qualidade dos dados sensorizados.

1.2 Hipótese

O balanceamento da carga de roteamento de dados produz uma melhor distribuição da energia disponível em uma rede de sensores sem fio. Esse balanceamento pode ser alcançado com o gerenciamento da energia restante nas rotas conhecidas.

Um mecanismo de roteamento de dados capaz de se adaptar às mudanças na distribuição de energia poderá aumentar a vida útil de uma rede de sensores sem fio.

1.3 Objetivos

Este trabalho possui como objetivo geral fazer uma análise de desempenho - usando um simulador - de protocolos de roteamento para RSSFs capazes de usar de maneira eficiente a energia da rede.

Como objetivos específicos:

- estudo e análise das características de um conjunto de protocolos visando a seleção de um protocolo alvo;
- modificação e/ou inclusão de novas características no protocolo alvo visando melhoria em seu desempenho;
- comparação, através de simulação, das versões original e modificada.

1.4 Organização do texto

Este capítulo abordou alguns conceitos iniciais do trabalho. O protocolo de roteamento DESRO (*Dynamic Energy Saving Routing*), gerado a partir de melhorias em protocolo existente, é foco

deste trabalho. A motivação desta iniciativa é evitar gastos desproporcionais de energia nos nós de uma RSSF através de uma melhor distribuição de energia na rede.

Este trabalho está organizado da seguinte forma: no Capítulo 2, é apresentada a revisão bibliográfica, na qual as redes de sensores sem fio são descritas com enfoque nos protocolos de roteamento; o Capítulo 3 descreve, de forma detalhada, o protocolo DESRO e protocolos que o originaram; no Capítulo 4, são abordados, além da metodologia utilizada, os resultados e as análises obtidas a partir das simulações efetuadas; o Capítulo 5 encerra o trabalho apresentando as conclusões finais e direções para trabalhos futuros.

2 *Redes de Sensores sem Fio*

A evolução na construção e miniaturização de dispositivos eletro-mecânicos, aliada à convergência de tecnologias como a Internet, redes de comunicações e sistemas de informação, possibilitou o nascimento de uma nova geração de sensores e atuadores.

Cada vez mais baratos, leves e menores, os sensores se tornarão, num futuro próximo, uma realidade presente na vida cotidiana. Milhões de nós sensores atuarão nos mais diversos sistemas exercendo diversas tarefas de maneira colaborativa. Projeções de mercado estimam que em 2010 serão entregues meio bilhão de nós sensores em aplicações de redes de sensores sem fio (SOHRABY; MINOLI; ZNATI, 2007).

2.1 Aplicações

Características como tamanho reduzido, mobilidade e baixo custo permitem que as RSSFs sejam usadas em diversas aplicações de monitoramento e sensoriamento remoto. Com as redes de sensores é possível manter um sensoriamento constante de áreas de difícil acesso ou que ofereçam risco à saúde.

Algumas possíveis aplicações para redes de sensores se encontram nas seguintes áreas (LOUREIRO et al., 2003):

- **sensoriamento de florestas/lagos/vulcões:** Os nós são distribuídos na área e informam qualquer alteração na temperatura/qualidade da água prevenindo assim incêndios ou remediando acidentes;
- **agricultura de precisão:** A rede pode informar as condições do solo - umidade, composição - ajudando a maximizar a produção;
- **segurança nacional:** sensoriamento de ambientes contendo agentes biológicos ou químicos assim como detecção de radiação em usinas;

- **operações militares:** a utilização de sensores na detecção de tropas inimigas, substâncias perigosas, como as utilizadas em explosivos, ou no acompanhamento da movimentação da própria tropa pode auxiliar grandemente um exército;
- **imóveis inteligentes:** mais segurança e conforto com o uso de uma rede de sensores em casas e prédios controlando luminosidade, temperatura, qualidade do ar dentre outros. A Figura 2.1 ilustra uma rede de sensores em uma casa;
- **saúde:** obtenção de informações sobre o estado de saúde - por exemplo, os batimentos cardíacos - e detecção de substâncias nocivas presentes em um paciente.

Com maior miniaturização dos nós sensores e melhoria da comunicação, os sensores serão no futuro instalados em produtos comerciais aumentando sua durabilidade e confiabilidade. Isso criará o que alguns pesquisadores chamam de computação invisível (*invisible computation*), computação universal (*pervasive computation*) ou computação ubíqua (*ubiquitous computation*) (SOHRABY; MINOLI; ZNATI, 2007).

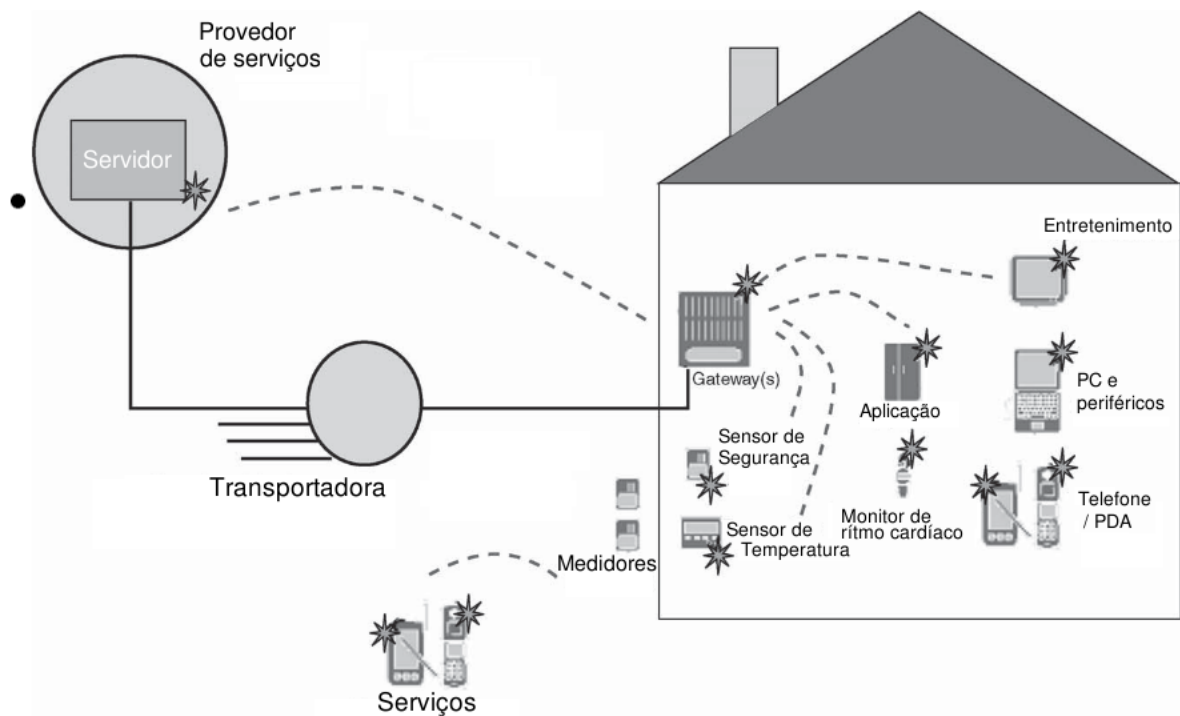


Figura 2.1: Rede de sensores em um domicílio. Fonte: (SOHRABY; MINOLI; ZNATI, 2007)

2.2 Nó Sensor

Uma RSSF é constituída de dezenas, centenas ou milhares de nós sensores. Estes nós possuem como tarefa principal a coleta de informações do ambiente, mas podem em alguns casos execu-

tar tarefas como roteamento de pacotes e processamento de dados. O nó é geralmente composto pelos componentes principais (Figura 2.2):

- **processador:** responsável pelo processamento dos dados sensorizados ou roteados podendo também ser usado na realização de uma tarefa cooperativa com outros nós sensores. O processador é bastante limitado, funcionando a baixas frequências (poucos MHz) a fim de economizar energia;
- **memória:** responsável pelo armazenamento da aplicação, dados e rotas utilizados pelo nó. Em geral, a quantidade de memória disponível a um nó sensor não passa de poucos megabytes, chegando em alguns casos a ter apenas poucos kilobytes;
- **unidade de energia:** responsável pelo fornecimento de energia às outras unidades do nó. Diversos tipos de baterias podem ser utilizadas nesta unidade e o nó pode possuir alguma forma de recarga de energia (ex: células fotoelétricas);
- **elemento de sensoriamento:** responsável pela transformação de eventos do ambiente em dados; é a interface do nó com o ambiente. Existem diversos tipos de sensores (acústicos, sísmicos, térmicos, sonoros, etc.) e um nó pode possuir mais de um sensor;
- **transceptor:** responsável pela transmissão e recepção de dados entre os nós. Os transceptores mais utilizados transmitem a baixas potências já que o maior consumo de energia em uma RSSF se deve às transmissões de dados.

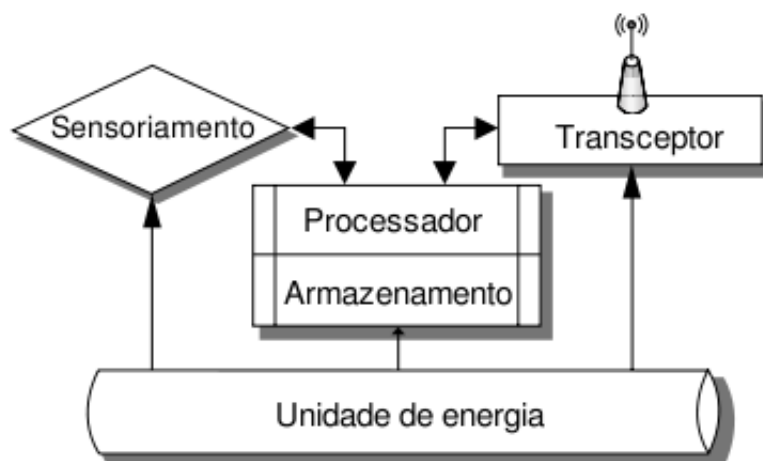


Figura 2.2: Arquitetura de um nó sensor. Fonte: (WU; TSENG, 2007)

2.3 Arquitetura de rede de sensores sem fio

2.3.1 Introdução

Uma rede de sensores sem fio é constituída de nós sensores responsáveis pelo sensoriamento do ambiente (fontes). Estes nós fontes são capazes de se comunicar entre si e, após captura de um dado do ambiente, enviam a informação a um nó especial chamado sorvedouro (*sink*). O sorvedouro é dito especial porque, em geral, difere dos outros nós em capacidade de hardware, energia e comunicação.

Existem três tipos possíveis de sorvedouro: ou ele pertence à rede de sensores e desta forma é apenas mais um nó sensor na rede; ou é externo à rede e desta maneira pode ser um *handheld* usado para interagir com ela; ou poderia, por fim, ser um *gateway* transmitindo para uma rede, externa ao ambiente sendo sensoriado, capaz de conectar usuários distantes da rede de sensores (KARL; WILLIG, 2007). A Figura 2.3 ilustra os tipos de sorvedouros.

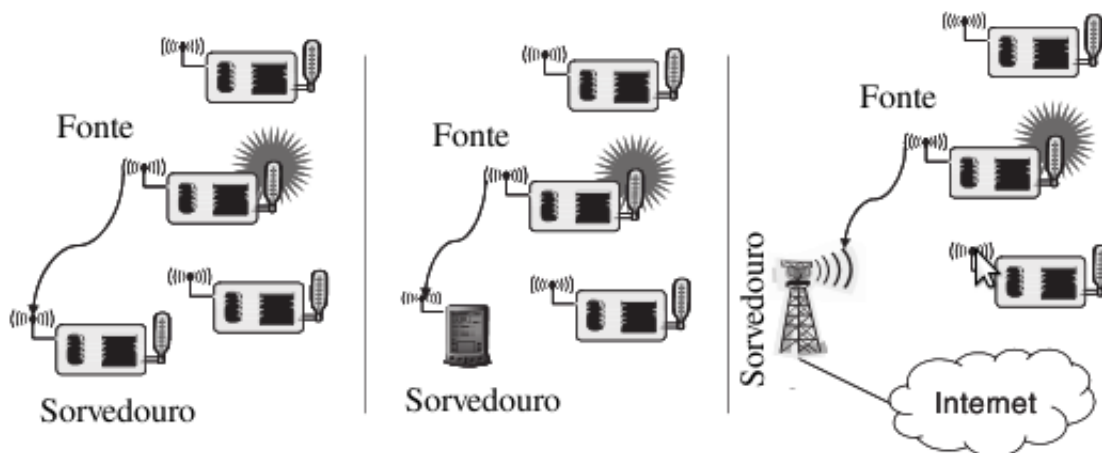


Figura 2.3: Tipos de sorvedouros. Fonte: (KARL; WILLIG, 2007)

2.3.2 Caracterização

Para Akyildiz et al. (2002) o projeto de uma rede de sensores sem fio é influenciada por fatores como tolerância a falhas, escalabilidade, custo de produção, ambiente operacional, topologia da rede, restrições de hardware, meios de transmissão e consumo de energia. Assim, uma rede de sensores sem fio possui características diversas e pode diferir de outras redes de sensores sem fio.

Uma RSSF pode ser organizada de forma plana - não há hierarquia entre os nós - ou de maneira hierárquica. Na organização hierárquica os nós organizam-se em grupos chamados

clusters e cada grupo possui um *cluster-head*. A hierarquia numa RSSF pode ser de qualquer nível.

Com relação à mobilidade a rede é dita móvel ou estacionária (quando não há mobilidade). A mobilidade pode estar presente no:

- **nó fonte:** neste caso os agentes de mobilidade são os nós sensores da rede, movendo-se sobre a região de sensoriamento;
- **sorvedouro:** os nós fonte permanecem estáticos mas o sorvedouro se move sobre a região de sensoriamento. Um usuário usando um PDA andando na região sensoriada é um exemplo de sorvedouro móvel;
- **evento:** a fonte de eventos é móvel quando a rede se destina a rastrear, por exemplo, um animal. Neste caso o movimento do animal deve ser percebido pela rede.

Algumas RSSF enviam dados periodicamente ao sorvedouro e são chamadas de redes periódicas (proativas). Agindo de forma diferente das redes periódicas estão as redes reativas, que enviam os dados somente na ocorrência de algum evento. Isso pode trazer o benefício de economia de energia mas nem todas as RSSF podem funcionar de maneira reativa. Como dito anteriormente, o projeto da rede é afetado por diversos fatores.

2.3.3 Comunicação

Quando uma informação é sensoriada do ambiente - temperatura, umidade, ruído etc. - o nó fonte a transmite ao nó sorvedouro. Se o nó sensor é capaz de transmitir diretamente ao sorvedouro dizemos que a comunicação é de um salto (*single-hop*). Se o nó sensor não for capaz de realizar a comunicação de um salto - um dos motivos pelo qual um nó fonte pode não conseguir transmitir o dado diretamente ao nó sorvedouro é se a distância entre os nós for maior do que o alcance do rádio - será utilizada a comunicação multi-saltos (*multi-hop*). Nesta comunicação podem ser usados diversos nós na transmissão do dado até o sorvedouro. Os nós intermediários funcionam como roteadores determinando a rota a ser seguida de forma análoga às redes comuns.

Vale notar que o uso de uma transmissão multi-saltos pode, em algumas situações, economizar energia. A atenuação do sinal de rádio é dado por cd^α sendo c uma constante, d a distância e α ($\alpha \geq 2$) a constante de atenuação; assim, a atenuação do sinal de rádio é, no mínimo, quadrática. Quando os nós estão muito próximos do chão o expoente pode chegar à

quarta ordem (AKYILDIZ et al., 2002). Se dividirmos a distância pela metade usando um nó como roteador a atenuação total será dada por $2c(d/2)^\alpha$. Entretanto, a fórmula de atenuação de sinal não leva em conta a energia mínima necessária pelo nó para transmitir, sendo assim, a soma de duas transmissões pela distância d não necessariamente consome menos energia que uma transmissão pela distância $2d$.

Como dito anteriormente, os nós sensores podem agir como roteadores e isso é possível graças à arquitetura de comunicação das RSSFs. A Figura 2.4 ilustra a pilha de protocolos de comunicação usada por uma RSSF. Essa ilustração é uma representação geral já que ela pode diferir de uma rede para outra visto que uma rede pode não precisar da camada de roteamento ou de um plano gerenciador de mobilidade.

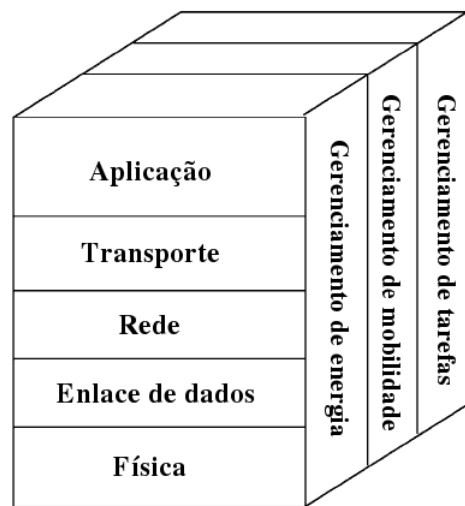


Figura 2.4: Pilha de protocolos. Fonte: (AKYILDIZ et al., 2002)

A camada física é responsável pelas técnicas de modulação na transmissão e recepção de sinais. A camada de enlace de dados é responsável pela escolha do momento adequado para a transmissão a fim de evitar colisões de pacotes. Para realizar essa escolha é, em geral, necessário escutar o canal. A camada de rede é a responsável pelo roteamento dos pacotes vindos da camada de transporte. Acima de todas essas camadas está a aplicação, que é quem decide o que fazer com o dado que foi sensoriado.

Há também os planos de gerenciamento de energia, mobilidade e tarefas. O plano de gerenciamento de energia decide como o nó se comportará de maneira a maximizar a eficiência no uso de sua bateria. Para isso o plano pode decidir, por exemplo, pelo desligamento do rádio em algumas situações. O plano de gerenciamento de mobilidade detecta o movimento de vizinhos e mantém as rotas atualizadas. O plano de gerenciamento de tarefas é responsável pela organização de tarefas cooperativas entre os nós. Esses planos são necessários às RSSF a fim de

que os nós possam trabalhar conjuntamente; sem os planos os nós trabalham individualmente apenas (AKYILDIZ et al., 2002).

2.4 Protocolos de Roteamento

Os nós fontes em uma RSSF podem atuar como roteadores de dados. Quando um dado sensoriado não pode ser transmitido diretamente ao sorvedouro resta a opção de transmiti-lo a um nó sensor mais próximo do sorvedouro a fim de que ele possa repassá-lo. Este nó intermediário roteia o dado pelo melhor caminho conhecido até o sorvedouro ou a outro nó sensor mais próximo do sorvedouro, e essa tarefa é executada até que o dado chegue ao seu destino.

Os protocolos de roteamento de dados em redes comuns - cabeadas ou sem fio - não podem ser usados em uma RSSF devido às características particulares de uma rede de sensores. Algumas destas características são:

- **escalabilidade:** uma RSSF pode possuir de poucas unidades a milhares de nós sensores. Nestes casos o protocolo de roteamento pode não conhecer a topologia da rede, dificultando a escolha do melhor caminho;
- **mobilidade:** a mobilidade dos nós sensores dificulta o roteamento de dados. Quando um nó procura uma rota para enviar um dado sensoriado pode escolher um caminho que não existe mais. Portanto o dado pode nunca ser entregue;
- **redundância de dados:** quando um evento ocorre na área de aplicação da rede, os dados coletados por um nó sensor podem ter sido também coletados por um nó vizinho e ambos transmitirão ao sorvedouro. Nesta situação o mesmo dado será transmitido duas vezes na rede, gerando redundância. A Figura 2.5 ilustra uma redundância quando um evento ocorre na região r . Os protocolos não específicos para RSSF não tratam a questão de redundância dos dados;
- **hardware:** o hardware que compõe o nó sensor possui fortes restrições de tamanho, potência e consumo de energia. De fato o processador de um nó sensor deve possuir uma baixa tensão fim de economizar energia; ocorrendo o mesmo com os demais componentes.

Os protocolos de roteamento comumente usados em redes sem fio não têm como uma das metas principais o baixo consumo de energia, que é o principal fator para um protocolo em uma RSSF; nem mesmo a arquitetura do Bluetooth é eficiente o bastante. Assim, é necessário

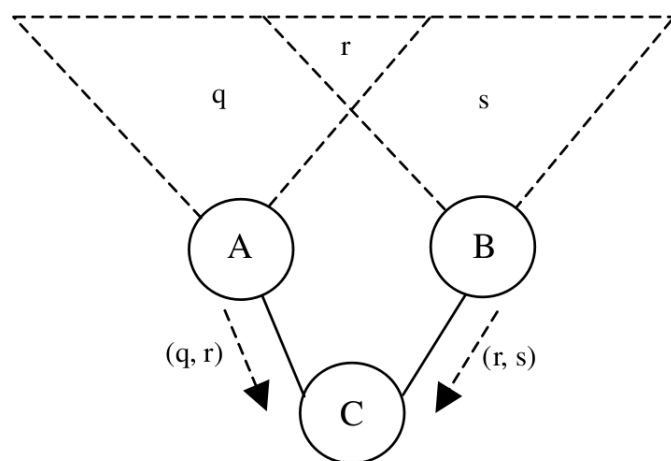


Figura 2.5: A e B transmitem o mesmo dado ao nó C. Fonte: (AKKAYA; YOUNIS, 2005)

o uso de protocolos especiais e a utilização de técnicas que os tornem mais eficientes. Uma importante técnica utilizada pelos protocolos em RSSF é a agregação (ou fusão) de dados.

A agregação de dados é utilizada na diminuição do número de mensagens e do tamanho dos dados trafegados pela rede, diminuindo o consumo de energia; consiste na detecção de dados redundantes e sua eliminação da rede, fusão de pacotes distintos e processamento parcial de dados sensorizados sendo enviado o resultado deste processamento.

2.4.1 Tipos de protocolos

De uma maneira geral todos os protocolos de roteamento para RSSFs existentes hoje podem ser classificados como: protocolos centrados em dados, protocolos hierárquicos, protocolos baseados em localização ou híbridos.

2.4.1.1 Protocolos centrados em dados

Quando a RSSF possui muitos nós, a identificação global única de cada nó sensor pode não ser eficiente. Nestes casos um protocolo que usa essa identificação (na tabela de rotas, por exemplo) não pode ser usado. Os protocolos centrados em dados se importam unicamente com a informação e são capazes de transmitir os dados mesmo em uma rede cujos nós não possuem identificadores únicos. Isso é possível porque estes protocolos trabalham com uma espécie de consulta de dados: “*Quais áreas possuem temperatura acima de 30 graus Celsius?*”.

Dentre os mais importantes protocolos centrado em dados estão o SPIN (*Sensor Protocols for Information via Negotiation*) (HEINZELMAN; KULIK; BALAKRISHNAN, 1999) e o *Directed Diffusion* (INTANAGONWIWAT; GOVINDAN; ESTRIN, 2000).

O SPIN foi um dos primeiros protocolos centrados em dados e possui a idéia de nomear os dados sensorizados através de meta-dados. A Figura 2.6 ilustra os passos necessários para transmitir dados no SPIN: antes de enviar propriamente os dados, o nó envia uma mensagem (ADV) anunciando que ele possui aquele dado (a); se algum dos vizinhos desejar a informação envia uma mensagem de requisição (REQ) (b) e só então o dado é transmitido (c); assim que o dado chega no nó que o requisitou é enviada uma nova mensagem de anúncio (d), partindo agora do novo nó; esse procedimento se repete até que o dado chegue ao sorvedouro (e) e (f).

O SPIN consegue economizar energia evitando o envio redundante de um mesmo dado pois os nós verificam se já possuem (se lhes interessa) aquela informação antes de requisitá-la. Os meta-dados que nomeiam os dados sensorizados não são definidos pelo protocolo, sendo antes definidos pela aplicação da rede.

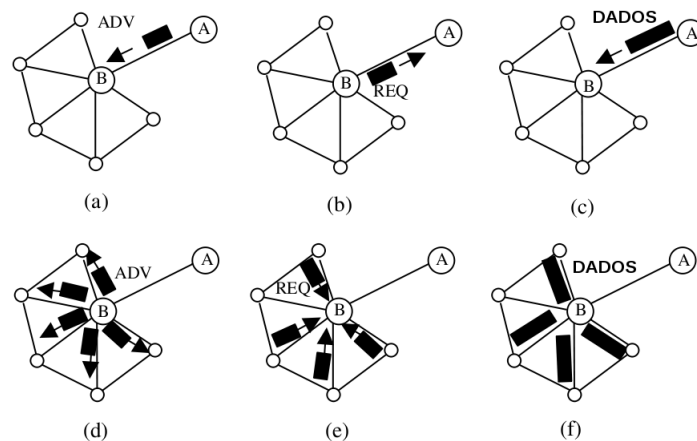


Figura 2.6: Funcionamento do SPIN. Fonte: (AKKAYA; YOUNIS, 2005)

O protocolo *Directed Diffusion* trabalha de maneira semelhante ao SPIN mas o sorvedouro é quem transmite em *broadcast* um interesse à rede. Esse interesse ou *query* não é nada mais do que uma lista de meta-dados do tipo atributo-valor contendo as informações que o sorvedouro deseja, por exemplo: (temperatura = 30; área = (0,0, 200, 200)).

O nó sensor ao receber o interesse grava-o em sua cache e o retransmite em *broadcast* pela rede. Quando um nó sensor obtiver um dado ele verificará em sua cache se o sorvedouro ainda tem interesse pelo dado - isto é, se o interesse não expirou - e então o transmite pelo gradiente de caminho criado quando do recebimento do interesse; esse gradiente determina qual o nó vizinho que transmitiu o interesse até ele. Vale notar, portanto, que o *Directed Diffusion* necessita distinguir de alguma maneira os nós vizinhos e, para isso, utiliza um identificador local como os identificadores de *clusters* no Bluetooth. A Figura 2.7 ilustra o funcionamento do protocolo: o sorvedouro envia um interesse (a); criação dos gradientes (b); detecção de evento e envio dos dados (c).

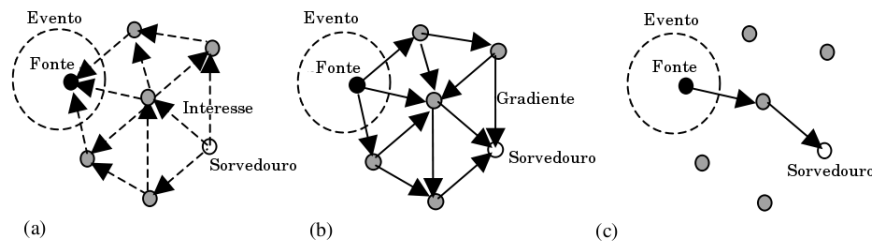


Figura 2.7: Funcionamento do Directed Diffusion. Fonte: (AKKAYA; YOUNIS, 2005)

2.4.1.2 Protocolos hierárquicos

Os protocolos hierárquicos foram desenvolvidos com o intuito de solucionar o desafio de escalabilidade da RSSF através da criação de grupos de nós (*clusters*). A comunicação multi-salto se dá dentro do *cluster*, quando os nós sensores enviam a informação até o líder (*cluster head*) e o líder de cada *cluster* envia o dado até o sorvedouro - possivelmente através de comunicação multi-salto entre os líderes de grupo. Desta maneira os nós sensores dentro de um *cluster* possuem uma visão limitada da rede não exergando além do seu grupo, permitindo uma economia de recursos. Os *clusters* são definidos a partir de informações como energia e proximidade dos nós sensores. Além de rotear dados os líderes de grupo podem executar a agregação de dados.

Um importante protocolo hierárquico, LEACH (*Low-energy adaptive clustering hierarchy*) (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000), é capaz de definir dinamicamente quais serão os líderes de *clusters*; os *clusters* são definidos através da potência de sinal recebida pelo líder. O LEACH trabalha em redes homogêneas - todos os nós são idênticos - e quando o líder está com pouca energia, é executado um procedimento para determinar qual nó será o novo líder.

O líder possui a tarefa de roteamento e agregação de dados e o protocolo não suporta transmissões multi-saltos, o que exige que o sorvedouro esteja dentro do raio de alcance do rádio do líder. Essa característica limita o uso de LEACH, não permitindo o seu uso em sensoriamento de grandes áreas.

Outro importante protocolo, TEEN (*Threshold sensitive Energy Efficient sensor Network*) (MANJESHWAR; AGRAWAL, 2001), é um protocolo hierárquico semelhante ao LEACH na formação de *clusters*, com a diferença de que há hierarquia nos *clusters*. A Figura 2.8 ilustra a hierarquia de *clusters* no TEEN.

Além de ser um protocolo hierárquico, o TEEN utiliza a informação sensorizada, assim como os protocolos centrado em dados, para determinar se envia ou não uma mensagem ao sorvedouro. Assim que os *clusters* são formados, os líderes enviam dois valores de limiar aos

nós sensores do seu grupo: *hard threshold* e *soft threshold*. Antes de enviar uma mensagem o nó compara o valor sensoriado e o *hard threshold* para decidir se deve enviar a mensagem ao líder, se a decisão foi por enviar ele compara o valor agora com o *soft threshold* para verificar se este novo valor é muito próximo ao valor já enviado anteriormente. Se a mudança for menor que o *soft threshold* a mensagem não será enviada.

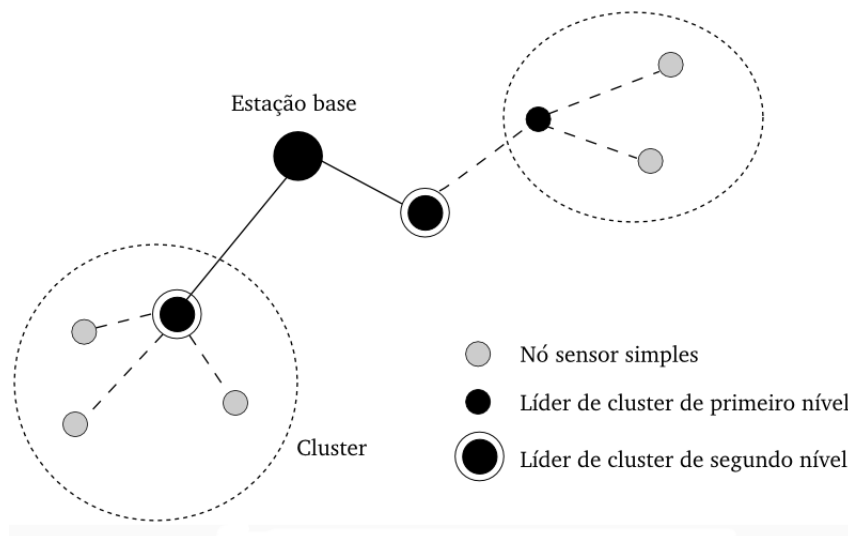


Figura 2.8: Hierarquia de *clusters* no TEEN.

2.4.1.3 Protocolos baseados em localização

Este protocolos, também chamados de protocolos geográficos, utilizam a informação de localização dos nós para determinar o melhor roteamento. Uma vantagem desta abordagem é que não há a necessidade de um identificador global dos nós. Conhecendo a localização da área onde se deseja capturar informações a transmissão de uma consulta - pelo *Directed Diffusion*, por exemplo - poderia ser mais eficiente, pois será transmitida diretamente na direção da região.

A localização do nó na rede pode ser determinada com o auxílio de um GPS (*Global Positioning System*), mas isso nem sempre é possível devido às restrições dos nós sensores. Uma alternativa é o emprego de técnicas que estimam a distância entre os nós; sabendo a posição correta de três nós no raio de alcance é possível executar uma triangularização e determinar com boa precisão a localização do nó sensor. A abordagem utilizada por Shee, Wang e Hsieh (2005) é capaz de determinar a localização dos nós sensores sem o uso de GPS.

A informação de localização pode também ser usada no desligamento de nós sensores muito próximos, isto é, sensoriando uma mesma área permitindo uma economia da energia total da rede ao evitar o envio de dados redundantes. Essa técnica é utilizada no protocolo GAF (*Geographic Adaptive Fidelity*) (XU; HEIDEMANN; ESTRIN, 2001). O GAF constrói uma grade

(*grid*) virtual e determina que apenas um nó sensor estará ativo numa área da grade em um dado momento, os outros nós dormirão até que sejam acordados a fim de que o nó até então acordado possa dormir; o objetivo é ter um e somente um nó ativo por área de grade.

2.4.1.4 Protocolos híbridos

Alguns protocolos agem de maneira proativa ou reativa dependendo da necessidade da rede em um dado instante e são, devido a esse comportamento, chamados de protocolos híbridos. Dentre os protocolos híbridos pode-se citar o Multi (FIGUEIREDO; LOUREIRO; NAKAMURA, 2004), Multi-K (GONÇALVES et al., 2008) e Multi-Geo (SILVA; GOULART, 2009).

O Multi é um protocolo híbrido que mistura comportamento reativo e proativo, dependendo da frequência de eventos. Quando ela é alta, o Multi funciona de forma proativa mantendo uma árvore de espalhamento com as rotas que abrangem toda a rede. Quando a frequência de eventos é baixa, o Multi opera de forma reativa. A mistura dos dois comportamentos se mostrou efetiva na redução do consumo de energia quando comparado com protocolos puramente reativos ou puramente proativos.

O Multi-K é um protocolo derivado do Multi que propôs uma redução na abrangência da árvore de espalhamento da rede, para que ela abrangesse apenas os nós que estivessem efetivamente gerando transmissão de dados. Desta forma consegue-se uma redução do consumo de energia na fase de funcionamento proativo, através da redução do número de mensagens de controle para a manutenção da árvore de espalhamento da rede.

O Multi-Geo propôs modificações no funcionamento do Multi-K considerando a rede dividida em áreas geográficas, cada qual com um nó líder. A transmissão entre áreas vizinhas só pode ser feita através do nó líder. Desta forma, conseguiu-se uma redução no consumo de energia tanto na fase proativa com a hierarquização da árvore de espalhamento, quanto na fase reativa com a limitação das mensagens de difusão que passaram a ser confinadas a uma determinada região.

2.5 Simulações em redes de sensores sem fio

O grande número de possíveis aplicações para redes de sensores sem fio fez da área um recente alvo de pesquisa, que só foram possíveis graças aos novos avanços da tecnologia. Porém, a grande diversidade de configurações que as RSSFs apresentam em cada camada (aplicação, rede, enlace de dados, física) são um desafio para os pesquisadores. Além disso, os protoco-

los utilizados em cada camada devem ser específicos, contemplando os requisitos e restrições apresentadas pela rede, sendo também recentes e ainda pouco utilizados na prática.

Para Karl e Willig (2007) certos requisitos são característicos de RSSFs, dentre eles pode-se citar: adaptação ao tipo de serviço, tolerância a falha, escalabilidade, densidade variável dos nós, nós programáveis. Ainda segundo Karl e Willig (2007), para a realização destes requisitos são necessários novas técnicas e o uso de mecanismos já conhecidos como comunicação multi-salto sem fio, operações eficientes quanto ao consumo de energia, auto-configuração e processamento colaborativo dentre outros. Todas essas características e mecanismos de RSSFs tornam a pesquisa na área um trabalho desafiador e nem sempre é possível dispor de nós em quantidade suficiente ou *hardware* compatíveis. Uma saída viável é o uso de simuladores de rede.

É possível encontrar na literatura diversos estudos importantes realizados com o auxílio de simuladores de eventos discretos como o NS-2 (NS-2, 2009). Essa é uma prática comum devido à facilidade na alteração dos algoritmos e modelos utilizados nas diversas camadas da rede durante a simulação. Com o uso de simuladores é possível testar redes de sensores com diversas configurações de densidade de nós, quantidade de dispositivos, diferentes *hardwares*, além de programar de forma mais rápida os protocolos para cada camada.

Após estudo e análise dos simuladores disponíveis e já utilizados na literatura a escolha recaiu sobre o Castalia (CASTALIA, 2009) pelos seguintes motivos:

- diferente do NS-2 o Castalia é um simulador de eventos discretos específico para RSSFs;
- é construído sobre o framework Omnet++ (OMNET++, 2009), o que trás mais confiabilidade quanto ao código fonte já que o simulador usa a *engine* de simulação provida pelo Omnet++. Esta *engine*, que se encontra na versão 4.0, se mostrou confiável após ter sido exaustivamente testada e modificada em diversos estudos. O Omnet++ é um ambiente de simulação de redes genérico capaz de realizar simulações de diversos tipos de redes e protocolos;
- é desenvolvido em C++, uma linguagem de programação de alto nível conhecida por sua eficiência;
- o código fonte possui estrutura modular, o que facilita seu entendimento e alteração;
- possui licença Academic Public License (APL, 2009) (mesma do Omnet++), que permite o uso e alteração do código fonte dos módulos sem qualquer ônus;

- possui um modelo de canal de transmissão sem fio atualizado para os mais recentes estudos;

2.6 Conclusão

O Capítulo 2 abordou aplicações, arquiteturas e protocolos de roteamento para RSSFs. Dentre as características e restrições apresentadas, a disponibilidade de energia é de grande importância. Quando não há a possibilidade de troca ou recarga da bateria usada por um nó, a rede deve tornar o consumo de energia o menor possível.

Os protocolos de roteamento para RSSFs buscam o uso eficiente da energia disponível. Neste capítulo foram apresentados protocolos de roteamento que fazem uso de diversas técnicas para economizar energia. Não foram apresentados, no entanto, os protocolos DSR (*Dynamic Source Routing*) (JOHNSON; MALTZ, 1996) e ESDSR (*Energy Saving Dynamic Source Routing*) (TARIQUE; TEPE; NASERIAN, 2005), importantes para este trabalho. Por isso eles serão apresentados detalhadamente no Capítulo 3 quando da apresentação do protocolo DESRO.

Apesar dos protocolos de roteamento apresentados neste capítulo economizarem energia através do uso de diversas técnicas, nenhum contempla todas as características apresentadas em Doshi, Bhandare e Brown (2002): escolha de rota, controle da potência de transmissão, descoberta de rota e monitoramento da rota. Essas características motivaram a criação do DESRO e serão apresentadas de forma mais completa no Capítulo 3.

3 *DESRO*

Redes de sensores sem fio possuem diversas limitações de recursos quando comparadas às redes de computadores comumente usadas. A limitação da quantidade de energia disponível em cada nó para a realização de diversas tarefas é uma importante questão e vem sendo tratada em diversas pesquisas. Em algumas aplicações, os nós não têm sua bateria trocada após seu uso e, nesses casos, a eficiência no uso da energia deve ser buscada. Com esse intuito, diversos protocolos de roteamento têm sido desenvolvidos: *Directed Diffusion*, GAF e Multi, dentre outros.

Outro importante fator que deve ser levado em conta pelo protocolo, além da eficiência no uso da energia, é a topologia da rede. Se a rede possui uma topologia conhecida e estática, o protocolo de roteamento usado pode tirar vantagem dessa informação para otimizar o roteamento dos dados. Se, entretanto, a rede não possui uma topologia conhecida ou se essa topologia varia no tempo, o protocolo usado deve ser capaz de se adaptar a essas mudanças. Protocolos com essa característica são chamados *ad hoc*.

Segundo Doshi, Bhandare e Brown (2002) os protocolos *ad hoc* são classificados em proativos (*table driven*) e reativos (*source initiated on demand*). Os protocolos proativos possuem uma tabela de rotas para cada nó que é atualizada periodicamente, mesmo que a rede não esteja transmitindo dado algum. Os protocolos reativos, por sua vez, enviam requisições de rota quando há algum dado a ser transmitido. Nesse mesmo trabalho são identificadas características desejáveis a um protocolo *ad hoc* para um uso eficiente da energia. Dentre elas pode-se citar:

- **escolha da rota:** o dado deve ser roteado levando-se em conta a energia disponível nos nós que fazem parte da rota. Assim, quando da escolha da rota a ser usada o protocolo deve levar em conta o número de saltos e a energia disponível;
- **controle da potência de transmissão:** antes de transmitir o dado o nó deve ser capaz de estimar a distância até o destino e transmitir usando a menor potência de rádio necessária para uma perfeita recepção pelo destino;

- **descoberta de rota:** o protocolo deve ser capaz de descobrir as rotas que consomem menos energia;
- **monitoramento da rota:** o protocolo deve ser capaz de atualizar a energia estimada da rota a fim de que o valor estimado não fique distante do valor real com o passar do tempo;

O DESRO (*Dynamic Energy Saving Routing*) é um protocolo de roteamento *ad hoc* desenvolvido neste trabalho. É um protocolo reativo para redes de sensores sem fio planas e estáticas e que contempla as características apresentadas acima.

O DESRO pode ser utilizado em diversas aplicações de sensoriamento, e suas técnicas para uso eficiente de energia são especialmente úteis em aplicações onde a bateria dos nós não pode ser recarregada/trocada. Devido às suas características, o DESRO é adequado para o sensoriamento de ambientes naturais, agricultura de precisão ou qualquer outra aplicação onde a taxa de transmissão de dados não é alta. Vale notar que o DESRO pode ser usado em qualquer aplicação onde o ESDSR é utilizado pois possui um funcionamento similar a este.

Para um melhor entendimento do funcionamento do DESRO é importante, entretanto, o conhecimento prévio dos protocolos que o inspiraram e que já apresentam algumas das características desejáveis a um protocolo de roteamento *ad hoc*.

3.1 DSR

O DSR (*Dynamic Source Routing*) (JOHNSON; MALTZ, 1996) é um protocolo de roteamento plano para redes sem fio *ad hoc* multi-salto. É capaz de se auto-organizar e promove um roteamento livre de ciclos (*loop-free*). Possui dois mecanismos que trabalham conjuntamente para a realização do roteamento de dados: descoberta de rota e manutenção de rota.

3.1.1 Descoberta de rotas

O DSR assume que cada nó da rede possui um identificador global único. Isso é necessário devido à maneira como o roteamento é realizado. Os nós sensores possuem tabelas de rotas em sua memória e a rota é gravada na mensagem sendo enviada determinando o caminho exato pelo qual ela trafegará - pode haver mais de uma rota para o mesmo destino, neste caso é escolhida a rota com menor número de saltos. Assim, quando um nó sensor deseja enviar um dado ao sorvedouro (ou para qualquer outro nó) ele verificará, em sua cache de rotas, se possui uma

rota até o nó destino, se possuir criará um pacote e enviará o dado, caso contrário iniciará o mecanismo de descoberta de rotas.

Ao verificar que não possui uma rota para um destino **E** desejado, o nó sensor **A** grava a mensagem que deseja enviar em um *buffer* chamado *Send Buffer*, e cria uma mensagem de descoberta de rota do tipo *Route Request* contendo o identificador da mensagem e o destino desejado. A mensagem gravada no *Send Buffer* aguardará pela rota desejada até que atinja o limite de tempo de vida, quando é então destruída. Se a resposta com a rota desejada chegar antes do limite de tempo a mensagem é retirada do *Send Buffer* e enviada normalmente.

A Figura 3.1 ilustra este mecanismo, quando o nó **A** envia em *broadcast* um *Route Request* cujo destino é o nó **E**.

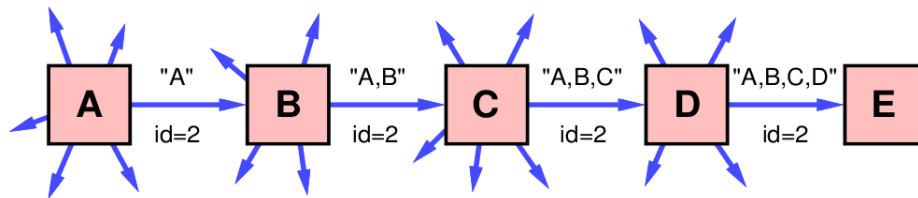


Figura 3.1: Nó A busca rota para o nó E. Fonte: (JOHNSON; MALTZ, 1996)

Ainda na Figura 3.1, quando o nó **B** recebe o *Route Request* verifica se é o destino da mensagem e se já recebeu este *Route Request* anteriormente (caso tenha recebido o pacote é descartado); não sendo ele o destino adiciona seu identificador à mensagem e a repassa, também em *broadcast*, aos nós vizinhos. Esse procedimento se repete até que a mensagem chega ao destino, quando então o nó **E** envia uma mensagem do tipo *Route Reply* ao nó **A**.

O *Route Reply* possui uma rota informando por quais nós a mensagem deve passar até ser entregue ao destino (nó **A**) e uma rota de resposta, pela qual o nó **A** poderá enviar os dados até **E**. A rota de resposta é gerada a partir da rota de nós intermediários, que foram adicionando seus identificadores quando da passagem do *Route Request*. Assim, quando o *Route Request* chegou ao nó **E** carregava a informação de que havia passado pelos nós **A**, **B**, **C** e **D**. Ao receber o *Route Reply* o nó **A** gravará a rota de resposta em sua cache de rotas para uso futuro.

Para enviar a resposta ao nó **A**, o nó **E** precisa conhecer um caminho até **A**. Se, ao receber o *Route Request* o nó **E** não conhecer nenhum caminho por onde possa enviar a resposta enviará um *Route Request* com a rota de resposta anexada a ele. Esta mensagem de *Route Request* com anexo é tratada da mesma maneira que o *Route Request* sem anexo, assim, quando o nó **A** receber esta mensagem irá, além de gravar a rota de resposta em sua cache de rotas, enviar uma mensagem de *Route Reply* para o nó **E** - agora ele já possui uma rota para **E**. Isso evita uma recursão infinita na descoberta de rotas.

Quando houver a garantia de que os *links* entre os nós são bidirecionais, o DSR pode otimizar a descoberta de rotas evitando o envio de mensagens *Route Request* com anexos. O destino irá, mesmo não possuindo uma rota na cache, responder ao *Route Request* através da inversão da rota de nós intermediários que transportaram o *Route Request*.

3.1.2 Manutenção de rota

Ao enviar um pacote de dados o nó fonte guarda uma cópia em seu *Send Buffer* e aguarda uma confirmação de recebimento. Essa confirmação é em nível de protocolo de roteamento, isto é, mesmo que a camada MAC tenha recebido o ACK, o DSR deve receber a confirmação. O nó que envia/repassa uma mensagem é responsável por confirmar o recebimento desta pelo próximo nó na rota - isto se repete até a chegada ao destino. Se o protocolo de camada MAC não possuir o mecanismo de confirmação de recebimento de mensagens - a maioria dos protocolos de camada MAC utilizados em RSSFs não o possuem - o DSR emulará este mecanismo enviando ele mesmo uma mensagem de confirmação.

Quando o nó não recebe a confirmação de recebimento dentro de um limite de tempo, o pacote é reenviado. Os pacotes permanecem no *Send Buffer* até que o nó receba a confirmação de recebimento do pacote ou até que o limite máximo de tentativas de envio ou o limite de tempo de vida seja alcançado.

O nó sensor, ao atingir o limite de tentativas de envio e não havendo confirmação de recebimento, enviará uma mensagem do tipo *Route Error* ao originador do pacote contendo o *link* quebrado. Por exemplo, na Figura 3.2 o nó **C** não recebeu a confirmação de recebimento vinda do nó **D** e atingiu o limite de tentativas de envio; o nó **C** envia então um *Route Error* ao nó **A** informando que o *link* entre **C** e **D** está quebrado. O nó **A**, ao receber o *Route Error*, exclui as rotas que possuem o *link* quebrado da sua cache de rotas.

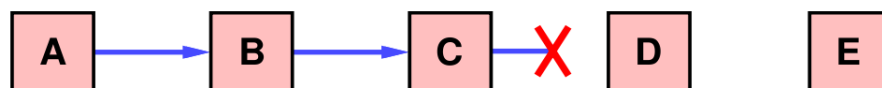


Figura 3.2: Nó atinge o limite de tentativas de transmissão. Fonte: (JOHNSON; MALTZ, 1996)

3.1.3 Funcionalidades adicionais

O DSR funciona corretamente comportando-se da maneira descrita acima mas algumas adições foram propostas por Johnson e Maltz (1996) a fim de torná-lo mais eficiente; dentre elas pode-se

citar:

- **Armazenamento de Informação de Rotas de Vizinhos**

Quando um nó repassa uma mensagem de dados, um *Route Request* ou um *Route Reply*, pode capturar rotas que ainda não possui em sua cache. Desse modo, na Figura 3.2, o nó **B** ao receber de **A** uma mensagem endereçada a **E** pode capturar rotas para **A**, **C**, **D** e **E**. Isso permitirá que numa eventual necessidade de rota haja maior chance de que a rota desejada já esteja em sua cache. Entretanto existem algumas limitações quando os *links* da rede não podem ser considerados bidirecionais.

Se os links não são bidirecionais não há como capturar rotas de um *Route Request*; no caso de uma mensagem de dados ou um *Route Reply* só podem ser capturadas rotas do nó atual até o destino (olhando para “frente”), nunca do nó atual para o nó fonte (olhando para “trás”). Isso pode ser melhor explicado pela Figura 3.2; quando o nó **B** repassar uma mensagem de dados, conseguirá capturar rotas para **C**, **D** e **E** mas não para **A**, já que não há a garantia de que os links são bidirecionais.

- **Respondendo Requisições de Rota usando Rotas Armazenadas**

Quando um nó recebe um *Route Request* e não é o destino mas possui em sua cache uma rota para o destino, poderá responder à solicitação criando, ele mesmo, um *Route Reply* que será enviado ao iniciador do *Route Request*. Dessa maneira o *Route Request* não será repassado e a mensagem de resposta chegará mais rápido ao iniciador da descoberta. A Figura 3.3 ilustra esse processo, quando o nó **C** recebe um *Route Request* com destino para **E** e retorna um *Route Reply* baseado em sua cache de rotas.

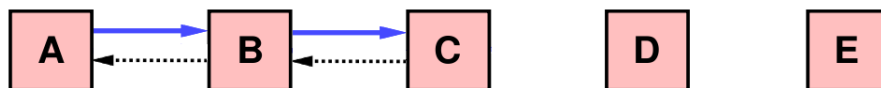


Figura 3.3: Envio de *Route Reply* usando cache de rotas.

Ao criar o *Route Reply* o nó usa os nós pelos quais o *Route Request* passou adicionando o caminho que possui em cache até o destino verificando se na rota resultante não há nós duplicados. Se houver duplicação ela é resolvida. Entretanto, existem situações em que o envio de um *Route Reply* torna-se impraticável após a retirada de nós duplicados. A Figura 3.4 ilustra uma dessas situações.

Neste caso o nó **F** possui uma rota para **E** em sua cache e recebe um *Route Request* vindo de **C** (mas originado em **A**). A rota de resposta após a resolução de nós duplicados é

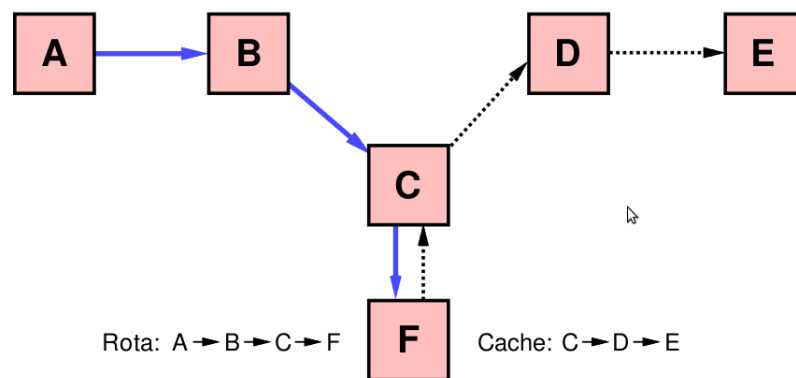


Figura 3.4: Possível duplicação de nó em um *Route Reply*. Fonte: (JOHNSON; MALTZ, 1996)

A-B-C-D-E, e desta forma o nó F não se encontra na resposta. O DSR estabelece que se o nó não estiver na rota de resposta não deve enviar um *Route Reply* a fim de que em caso de um problema na rota e envio de *Route Error* o nó originador do *Route Reply*, que continha a rota com problema, seja também informado de que a rota não é mais confiável. Se o nó F responder ao *Route Request* e o link entre D e E não estiver funcionando, o nó F não será informado pelo *Route Error* e poderá prejudicar novamente o nó A ou qualquer outro nó buscando rota para E.

- **Salvamento de pacote**

Em seu comportamento padrão o nó descarta a mensagem de dados quando esta atinge o limite máximo de tentativas de envio, e após o descarte envia um *Route Error*. Com o *Packet Salvaging* o nó verifica se possui em sua cache de rotas um caminho alternativo até o nó destino da mensagem e, se houver, a rota da mensagem é alterada sendo transmitida agora pela nova rota. Uma mensagem é salva apenas uma vez e, mesmo com salvamento, o *Route Error* é enviado a fim de alertar ao originador de que a rota não é mais confiável.

- **Aumento do Espalhamento de Mensagens de Erro de Rota**

O *Route Error* trafega por uma rota específica, a saber o inverso da rota pela qual a mensagem de dados passou até encontrar o *link* quebrado. A fim de que a informação sobre o *link* quebrado seja conhecida pelo maior número de nós na rede essa informação pode ser anexada a um *Route Request*.

Se um nó recebe um *Route Error*, elimina todas as rotas contendo o *link* quebrado e necessita de rota para um destino para o qual havia antes uma rota, envia um *Route Request* com a informação do *link* quebrado anexada. Desse modo, ao receber o *Route Request* com a informação anexada o nó vizinho tomará conhecimento de que aquele determinado *link* está quebrado, eliminará as rotas que faziam uso dele, e só então repassará o *Route*

Request ou criará um *Route Reply* baseado em sua cache. Esse *Route Reply* é agora mais confiável, já que o nó poderia responder com uma rota contendo o *link* quebrado caso não houvesse a informação anexada.

3.2 ESDSR

O ESDSR (*Energy Saving Dynamic Source Routing*) (TARIQUE; TEPE; NASERIAN, 2005) é um protocolo de roteamento para RSSFs *ad-hoc* baseado no DSR e seu funcionamento é bastante semelhante ao funcionamento do DSR, diferindo principalmente na maneira como uma rota de transmissão é escolhida.

O ESDSR procura economizar energia através da diminuição de potência utilizada pelo rádio quando transmitindo dados. Se o nó vizinho para o qual se está transmitindo se encontra próximo ao nó transmissor pode não ser necessário utilizar a potência máxima do rádio; neste caso uma transmissão em uma potência menor pode ser recebida sem problemas. Assim, o ESDSR calcula a potência mínima necessária para a transmissão e a usa.

No DSR quando há mais de uma rota para o mesmo destino, a rota com menor número de saltos é sempre escolhida. Essa decisão não envolve a questão de consumo de energia, isto é, mesmo que o consumo de energia seja maior quando o pacote de dados trafega pelo caminho com menor número de saltos este caminho será o escolhido. O ESDSR procura melhorar essa decisão calculando a expectativa mínima de vida estimada da rota, isto é, usando a rota $R(t)$ no tempo t tal que $R(t)$ maximize a Equação (3.1):

$$C(R, t) = \max(R_j(t)) \quad (3.1)$$

$$R_j(t) = \min\left(\frac{E_i}{P_i}\right) \quad (3.2)$$

onde $R_j(t)$ é a expectativa mínima de vida estimada para a transmissão pela rota R_j , E_i é a energia restante do nó i e P_i é a potência gasta pelo nó i quando transmitindo pela rota j . Ou seja, o ESDSR procura utilizar a rota com a maior energia estimada, que é dada pelo menor valor da relação $\frac{E_i}{P_i}$ para todos os nós da rota.

Para que a energia estimada de uma rota seja calculada é necessário o conhecimento do valor de P_i , que é a potência usada na transmissão pelo rádio do nó i no instante t . Essa potência varia para cada destino, isto é, quando o nó i transmite para o nó j ele o faz com uma determinada potência, quando transmite para o nó k o faz com uma potência independente da potência

usada quando transmitindo para o nó j .

No ESDSR os nós sensores possuem uma tabela (tabela de potências) contendo o nó vizinho e a potência usada para transmitir àquele nó. Se o nó i nunca transmitiu para o nó k não haverá nenhuma entrada para k na tabela de potências e, dessa maneira, quando houver a primeira transmissão ou quando for necessário conhecer essa potência para o cálculo de energia estimada da rota, será usada a potência máxima. Essa potência será, no entanto, atualizada sempre que houver uma transmissão para k .

Quando o nó i enviar uma mensagem para o nó k , gravará no cabeçalho do pacote o valor da potência de rádio utilizada na transmissão. O nó k , ao receber uma mensagem do nó i , terá então conhecimento do valor da potência em que o pacote foi transmitido (P_{tx}) e também o valor da potência recebida (P_{rec}). Com esses dados é possível calcular a nova potência P_{min} mínima usada por i quando transmitindo para k . O cálculo é realizado usando a equação (3.3):

$$P_{min} = P_{tx} - P_{rec} + P_{threshold} + P_{margin} \quad (3.3)$$

onde $P_{threshold}$ é o limiar de recepção do rádio e P_{margin} é uma margem de segurança para *links* instáveis devido a flutuações no canal. O resultado do cálculo de P_{min} é então gravado no *ACK*, que o nó k deve enviar para i , e quando ele for recebido o nó i atualizará sua potência de transmissão para k .

A tabela de potências só é usada para transmissões de dados. As transmissões de pacotes de controle do ESDSR (*Route Request*, *Route Reply*, dentre outros) são realizadas na potência padrão (máxima) do rádio, assim como as transmissões de controle (RTS/CTS/ACK) da camada MAC.

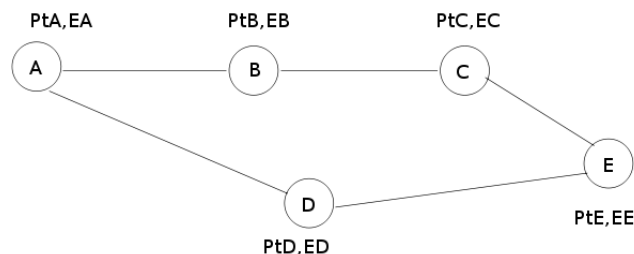


Figura 3.5: Roteamento no ESDSR. Fonte: (TARIQUE; TEPE; NASERIAN, 2005)

Suponha que na Figura 3.5 o nó **A** deseja enviar dados ao nó **E** mas não possui uma rota para isso. Um *Route Request* é então criado pelo nó **A** chegando ao nó **E** por dois caminhos:

A-B-C-E e A-D-E. Neste caso o nó **E** envia duas mensagens *Route Reply*, uma para cada rota descoberta.

No ESDSR a mensagem *Route Reply* possui um campo a mais se comparada à mensagem de resposta de rotas do DSR. Neste novo campo será guardado um valor de expectativa mínima de vida da rota. Essa expectativa é dada pela relação $\frac{E_i}{P_i}$. À medida que a *Route Reply* percorre a rede este valor é atualizado. Na Figura 3.5 o nó **C** calcula sua expectativa usando a sua energia restante, EC e a potência utilizada na transmissão para o nó **E**, PtC . Este valor é guardado no *Route Reply* e será substituído se o valor dado pela relação $\frac{EB}{PtB} - EB$ é a energia restante do nó **B** e PtB é a potência utilizada na transmissão para o nó **C** - for menor do que o valor de expectativa gravado até então no *Route Reply*. Isso se repete até que a resposta chegue à origem, o nó **A**. Desta maneira o nó **A** sabe qual a menor expectativa de vida para as rotas em sua cache e pode decidir pela rota A-B-C-E se esta possuir uma expectativa de vida maior do que a rota A-D-E. O DSR sempre escolheria a rota A-D-E pois esta possui menos saltos.

3.3 DESRO

O ESDSR contempla a maioria das características desejáveis em um protocolo de roteamento *ad hoc* para RSSFs citadas em (DOSHI; BHANDARE; BROWN, 2002), não contemplando, entretanto, o monitoramento das rotas.

Se o nó **A** deseja transmitir ao nó **Z** e não possui uma rota, enviará um *Route Request*. Após receber um *Route Reply* (ele pode receber várias respostas), que contém a energia estimada da rota, o nó **A** escolherá a rota com maior energia estimada e transmitirá por ela. Porém, com o passar do tempo, a rota com maior energia estimada pode já não mais possuir a maior energia já que vem sendo sempre escolhida pelo nó **A**. Se o tráfego for intenso, os nós que fazem parte da rota podem descarregar enquanto nós vizinhos, que poderiam ajudar no roteamento, possuem ainda bastante energia. Essa escolha constante de uma rota pode ocasionar até mesmo a partição da rede. O DESRO tem por objetivo corrigir essa deficiência do ESDSR através do monitoramento de rotas, alcançando assim uma melhor distribuição da energia pela rede.

Após a descoberta de uma rota, o DESRO inicia um temporizador e/ou um contador de pacotes. Assim que o temporizador zerar ou o contador de pacotes atingir um limite pré-estabelecido, é enviada uma mensagem *Route Management* que percorrerá cada nó da rota até alcançar o destino. Após o envio da mensagem de gerenciamento, o temporizador é reiniciado e o contador zerado.

O nó destino, ao receber uma mensagem de gerenciamento, envia um *Route Reply* que

passará pelo caminho inverso do percorrido pela *Route Management*. O mesmo processo que ocorre em cada nó ao repassar um *Route Reply* gerado por um *Route Request* ocorrerá para o *Route Reply* gerado pelo *Route Management*. Desse modo o nó originador do *Route Request* receberá uma atualização para a energia estimada da rota e poderá decidir de forma mais eficaz por qual caminho enviar um pacote de dados. A Figura 3.6 ilustra, de forma resumida, o funcionamento do DESRO.

É importante notar que o *Route Management*, assim como todos os pacotes de controle do ESDSR, são enviados sem qualquer confirmação de recebimento pela camada de rede e na potência máxima de rádio. Por não haver essa confirmação de recebimento, o contador e o temporizador de uma rota são reiniciados quando um *Route Request* ou *Route Reply* é recebido.

O envio de um *Route Management* pode ser disparado por um:

- **temporizador:** assim que uma rota é descoberta, é criado um temporizador com valor pré-estabelecido para ela. Quando o temporizador chegar a zero, é hora de atualizar a energia estimada da rota; uma mensagem de gerenciamento é enviada e o temporizador é reiniciado;
- **contador de pacotes:** assim que uma rota é descoberta, é criado um contador de pacotes. Cada pacote enviado incrementa o contador que ao chegar no limite pré-estabelecido dispara o envio de uma mensagem de gerenciamento e é zerado;
- **temporizador e contador de pacotes:** neste caso, quando uma rota é descoberta, tanto o contador quanto o temporizador são criados e vai disparar o envio de mensagem de gerenciamento o evento que ocorrer primeiro: contador chegar ao limite ou temporizador zerar.

A escolha de qual abordagem de gerenciamento de rotas usar depende da carga de transmissão de dados e de geração de eventos pela rede. Esse processo não é automático no DESRO, o que significa que a abordagem é definida na implantação da rede e não muda durante a sua vida. É possível, no entanto, tornar o mecanismo de gerenciamento de rotas do DESRO adaptável a mudanças na carga de dados trafegados pela rede, o que permitiria ao protocolo decidir qual gatilho disparará o gerenciamento de rotas de maneira dinâmica. Para realizar essa implementação, seria necessária a adição de um algoritmo de estimativa de carga ao DESRO, o que o tornaria mais complexo e consumiria mais energia na fase de processamento. Além disso, o resultado da estimativa poderia não ser o melhor para a carga futura, isto é, o nó poderia estimar que a carga seria alta no futuro quando, na verdade, poderia ser menor do que no passado devido a diversos fatores.

```

LOOP (evento)
  Se evento = "enviar novo pacote de dados" então
    Se houver rota para o sorvedouro então
      Envia o pacote;
      Se gerenciamento de rotas é feito por contagem de pacotes então
        Soma mais um ao contador;
        Se contador alcançou o limite pré-estabelecido então
          Zera o contador;
          Cria mensagem de gerenciamento (Route Management) e a envia;
        Fim Se
      Fim Se
    Senão
      Envia uma requisição de rota (Route Request);
      Aguarda rota colocando o pacote no buffer por um tempo pré-determinado;
    Fim Se
  Fim Se
  Se evento = "temporizador zerou" então
    Reinicia temporizador;
    Cria mensagem de gerenciamento (Route Management) e a envia;
  Fim Se
  Se evento = "chegada de pacote de dados" então
    Se nó não é o sorvedouro Repassa pacote;
  Fim Se
  Se evento = "chegada de mensagem de requisição de rota" então
    Se nó é o sorvedouro então Envia uma mensagem de resposta de rota (Route Reply);
    Senão
      Repassa mensagem;
    Fim Se
  Fim Se
  Se evento = "chegada de mensagem de resposta de rota" então
    Adiciona/Atualiza rota na tabela de rotas;
    Cria/Atualiza temporizador e/ou contador de pacotes
    Se houver pacote de dados esperando por rota então Envia pacote de dados;
    Se não é o destino da mensagem então
      Calcula energia mínima estimada;
      Repassa mensagem;
    Fim Se
  Fim Se
  Se evento = "chegada de mensagem de gerenciamento" então
    Se não é o destino então Repassa mensagem;
    Senão Envia uma mensagem de resposta (Route Reply) ao originador;
  Fim Se
FIM LOOP

```

Figura 3.6: Pseudo-código do DESRO.

O mecanismo de gerenciamento de rotas do DESRO pode ser configurado segundo as necessidades da rede de sensores. Se a rede gera muitos dados, o contador de pacotes pode ser priorizado em relação ao tempo; quando a rede não é muito ativa, a abordagem por tempo pode ser suficiente. Se, entretanto, a rede alterna momentos de carga e de inatividade, o uso do trabalho conjunto de contador e temporizador pode ser uma alternativa interessante, caso em que o contador dispararia o gerenciamento quando a carga de atividade for alta e quando a carga de atividade for baixa o gerenciamento seria feito pelo temporizador.

Vale notar que o ESDSR e o DESRO funcionam de maneira idêntica quanto à descoberta e manutenção de rotas. A diferença entre os protocolos está no gerenciamento de rotas, que é encontrado no DESRO mas não no ESDSR. Portanto, os procedimentos executados pelo DESRO quando um nó falha, ou tem sua energia exaurida, são os mesmos procedimentos executados pelo ESDSR: se não há uma rota alternativa é gerada uma nova mensagem de descoberta de rotas.

3.4 Detalhes da codificação

O ESDSR e DESRO foram codificados seguindo suas respectivas especificações. Alguns pontos da codificação merecem, no entanto, destaque:

- **cabeçalho de pacotes:** os cabeçalhos dos pacotes do ESDSR e DESRO contém identificadores de nós e outros dados necessários ao roteamento. A Tabela 3.1 fornece o número de bytes usados nos parâmetros dos cabeçalhos dos pacotes do ESDSR e DESRO.

Parâmetro	Tamanho (em bytes)
Identificador de nó	1
Identificador da mensagem	2
Indicador de mensagem salva	1
Potência em que a mensagem foi transmitida	1
Energia estimada da rota	4

Tabela 3.1: Representação de parâmetros.

Vale notar que o identificador de nó usado na implementação deste trabalho é representado por 1 (um) byte; isso permite um máximo de 256 nós na rede. Se, no entanto, a rede possuir mais nós, basta configurar a quantidade de bytes que representarão o identificador de nó.

A mensagem de requisição de rotas (*Route Request*) possui em seu cabeçalho: identificador do nó originador, identificador do nó destino e identificador da mensagem. Além

desses valores, há também uma lista (de tamanho variável) de identificadores contendo cada nó intermediário que repassou a *Route Request*.

A mensagem de requisição de rotas com informação de *link* quebrado (*Route Request Error*) possui em seu cabeçalho os mesmos valores do cabeçalho da *Route Request*, além dos identificadores dos nós fonte e destino do *link* quebrado.

A mensagem de resposta (*Route Reply*) contém em seu cabeçalho: identificador para cada nó presente na rota de resposta e a energia estimada da rota.

A mensagem de erro (*Route Error*) contém em seu cabeçalho: identificador para cada nó presente na rota até o originador da mensagem de dados e os identificadores dos nós fonte e destino do *link* quebrado.

A mensagem de dados contém em seu cabeçalho: identificador para cada nó presente na rota pela qual tráfegará o pacote, identificador da mensagem, indicador de mensagem salva e potência em que a mensagem foi transmitida

- **repassa de *Route Request*:** neste trabalho, o modo como um nó verifica se o *Route Request* foi recebido anteriormente segue uma lógica diferente da implementação original do DSR (e ESDSR). No DSR original, o *Route Request* é descartado se o nó já tiver recebido um *Route Request* com o mesmo identificador e originado pelo mesmo nó (DOSHI; BHANDARE; BROWN, 2002). Nas codificações para o ESDSR e DESRO, realizadas neste trabalho, o *Route Request* só é descartado se o nó já tiver recebido um *Route Request* com o mesmo identificador, originado pelo mesmo nó e transmitido pelo mesmo nó. Assim, um *Route Request* com id 1, gerado pelo nó **A** e recebido de **B** será diferente de um *Route Request* com id 1, gerado pelo nó **A** e recebido de **C**. Essa alteração na lógica de descarte permite que mais rotas sejam descobertas e alcança uma das características desejáveis propostas em Doshi, Bhandare e Brown (2002), a saber, a descoberta de rotas que consomem menos energia. A Figura 3.7 ilustra este processo:
 - a) o nó **A** envia uma requisição a fim de descobrir rotas para **D**;
 - b) os nós **B**, **C** e **D** escutam a requisição; **B** e **C** a repassam;
 - c) o nó **D** responde a requisição de **A**;
 - d) o nó **D** responde as requisições repassadas por **B** e **C**;
 - e) o nó **D** responde a requisição que passou por **B** e por **C**; no DSR original o nó **C** descarta a requisição recebida de **B** porque assume já a ter recebido, o que não acontece na versão deste trabalho porque ele verifica também de quem a requisição foi recebida;

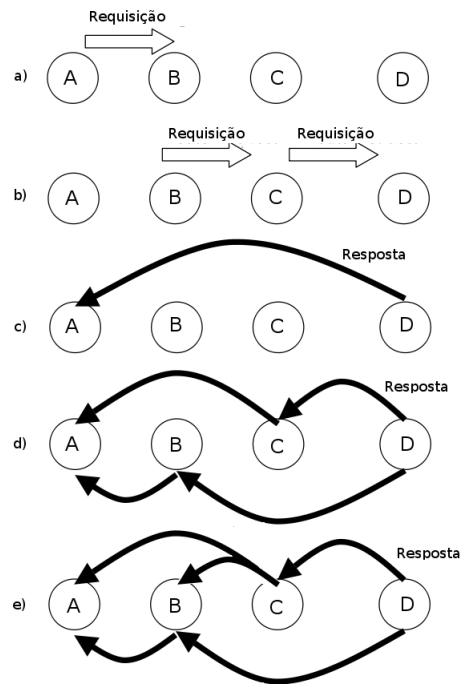


Figura 3.7: Repasse de *Route Request*.

- cálculo da potência de transmissão:** o DESRO, assim como o ESDSR, precisa conhecer a potência com que um pacote de dados foi recebido a fim de determinar a potência mínima de transmissão que será gravada no ACK. Para esse cálculo foi usado o RSSI (*Received Signal Strength Indication*), já implementado no Castalia, que informa exatamente a potência recebida e pode ser usado como boa estimativa para distância entre os nós (SRINIVASAN; LEVIS, 2006);
- captura de rotas:** a captura de rotas ocorre quando um nó recebe uma mensagem. Na especificação original do DSR, a captura de rotas ocorre nas mensagens *Route Request*, *Route Reply* e *Route Error*. Já no ESDSR e DESRO a captura de rotas se dá apenas com o *Route Reply*, porque esta é a única mensagem que carrega a informação de energia estimada da rota;
- gerenciamento de rotas:** o DESRO pode otimizar o gerenciamento de rotas se um nó puder aproveitar o gerenciamento iniciado por outro nó; isso é feito na implementação deste trabalho. Quando o nó A envia uma mensagem de gerenciamento para uma rota cujo destino é D, os nós intermediários B e C aproveitam a resposta de D e atualizam a energia estimada das rotas C-D e B-C-D caso existam em suas caches. Esse processo zera o contador e reinicia o temporizador. Neste caso, apenas uma mensagem de gerenciamento foi suficiente para gerenciar diversas rotas em diferentes nós. Além disso, na implementação deste trabalho, o gerenciamento é realizado apenas para rotas cujo destino

já foi destinatário de mensagens geradas pelo nó. Isso impede que a energia do nó seja gasta com gerenciamento de rotas para destinos nunca utilizados, já que durante a fase de descoberta de rotas várias são descobertas para nós para os quais nada será enviado.

4 Avaliação do DESRO

4.1 Metodologia

A fase inicial do trabalho consistiu no estudo e análise de protocolos de roteamento *ad hoc* para RSSFs que procuram utilizar de forma eficiente a energia durante o roteamento e transmissão de dados, e que ainda não contemplassem todas as características desejáveis a um protocolo *ad hoc* descritas em Doshi, Bhandare e Brown (2002). O objetivo era identificar os pontos fracos dos protocolos e uma maneira de saná-los.

O DESRO foi gerado a partir de alterações no ESDSR. Para a avaliação das alterações, foram realizadas implementações e simulações dos protocolos de roteamento ESDSR e DESRO.

4.2 Simulações

Foram realizadas 40 simulações utilizando os protocolos ESDSR e DESRO; 20 com distribuição em grade e 20 com distribuição aleatória uniforme, simulando uma aplicação de sensoriamento ambiental de uma área de 160x360 metros, onde 49 nós coletam dados do ambiente a cada 10 minutos e os enviam ao sorvedouro. A área é sensoriada por um mês e os nós possuem 20 Joules de energia inicial na distribuição em grade e 10 Joules na distribuição uniforme aleatória.

Foi utilizada uma escala de tempo de 30:1 e, deste modo, cada dia durou 2880 segundos fazendo com que a simulação durasse 86400 segundos (para 30 dias). No início da simulação cada nó acorda aleatoriamente entre 5 e 20 segundos e passa a transmitir, a cada 20 segundos, uma mensagem ao sorvedouro contendo 20 bytes de dados, referente ao sensoriamento do ambiente.

O gerenciamento de rotas do DESRO foi configurado com temporizador de 14400 segundos e contador de pacotes para 144 transmissões. Desse modo, a rota sendo utilizada num dado momento terá sua energia estimada atualizada assim que completar o equivalente a 1 dia de transmissão, enquanto que as rotas não utilizadas terão sua energia atualizada a cada 5 dias.

Esses valores foram escolhidos a fim de que o gerenciamento de rotas consumisse abaixo de 1% da energia consumida com a transmissão de dados.

Na topologia em grade, a distância máxima de um nó ao sorvedouro é de 12 saltos, o que significa que serão gastos pelo menos 28 bytes com o gerenciamento de uma rota do nó mais distante do sorvedouro: 12 bytes para o pacote de gerenciamento e 14 bytes para o pacote de resposta. Por dia são transmitidos 2880 bytes de dados sensorizados. Assim, um gerenciamento de rota por dia consome, em média, 0,97% da energia consumida com transmissão de dados.

Os 5 segundos iniciais da simulação são reservados à sincronização dos nós pelo T-MAC (DAM; LANGENDOEN, 2003) - protocolo de camada MAC próprio para RSSF. O T-MAC foi configurado para usar um quadro de tempo de 610 milisegundos e controle RTS/CTS/ACK para transmissões *unicast*.

O T-MAC foi modificado para incluir em sua mensagem de confirmação - ACK - um campo contendo o valor da potência de sinal com que o pacote de dados foi recebido; esse valor é usado no cálculo da potência de transmissão do próximo pacote. Outra modificação foi feita, para permitir que o DESRO (e ESDSR) pudesse alterar a potência de transmissão do rádio de acordo com o nó vizinho para o qual se está transmitindo no momento; para isso foi criada uma nova função na interface da MAC.

Parâmetro	Valor
Largura de banda	250 kbps
Alcance	45 m
Límiar de recepção	-95 dBm
Potência gasta na recepção	62 mW
Potência de transmissão a 0 dBm	57.42 mW
Potência de transmissão a -1 dBm	55.18 mW
Potência de transmissão a -3 dBm	50.69 mW
Potência de transmissão a -5 dBm	46.2 mW
Potência de transmissão a -7 dBm	42.24 mW
Potência de transmissão a -10 dBm	36.3 mW
Potência de transmissão a -15 dBm	32.67 mW
Potência de transmissão a -25 dBm	29.04 mW

Tabela 4.1: Parâmetros do CC2420

Essas modificações podem ser feitas em qualquer protocolo de camada MAC não sendo obrigatório o uso do T-MAC para o correto funcionamento do ESDSR ou DESRO. Ou seja, o ESDSR e DESRO podem ser usados juntamente com qualquer protocolo de camada MAC desde que eles já possuam essas capacidades ou, quando não possuindo, através das modificações necessárias na interface e na mensagem de confirmação de recebimento de pacote.

Como módulo de rádio foi utilizado o CC2420. O T-MAC e o CC2420 estão presentes na versão do Castalia utilizada nas simulações (versão 2.3b). Para o CC2420 foi usada a modulação ideal, a fim de que não ocorressem falhas na transmissão causadas pela modulação do rádio; o modelo de propagação utilizado foi o *two ray path loss*, sem *fading*. Algumas das características do CC2420 podem ser vistas na Tabela 4.1:

Para análise do consumo de energia foi utilizado um modelo de consumo similar ao proposto em Doshi, Bhandare e Brown (2002) e usado no estudo original do ESDSR. A diferença entre os modelos de consumo de energia reside no fato que o modelo usado no presente estudo considera apenas a energia gasta com transmissões de pacotes de dados e de controle da camada de rede; a energia gasta com pacotes de controle da camada de enlace de dados (RTS/CTS/ACK) não foi computada. A energia gasta por um pacote de dados de comprimento D bytes transmitido numa potência P_t é dada pela Equação (4.1):

$$E(D, P_t) = \frac{P_t D}{L} \quad (4.1)$$

onde L é a largura de banda do *link* em bytes por segundo. Na Tabela 4.1 são descritos os valores possíveis de P_t para o CC2420.

As simulações foram realizadas em um *notebook* com processador Core 2 Duo 1.83GHz, disco rígido de 160GB e memória RAM de 2GB; sistema operacional Linux, distribuição Ubuntu 9.04; para a geração de gráficos foi utilizado o software QtiPlot versão 0.9.7; a análise dos dados de saída do Castalia foi realizada por um script criado na linguagem Lua (LUA, 2010). Os pontos observados foram:

- consumo de energia da rede;
- distribuição de energia na rede;
- taxa de entrega da rede;
- tempo de resposta da rede;

4.3 Resultados

Nessa seção são discutidos os resultados obtidos para as simulações nas distribuições uniforme não-aleatória e uniforme aleatória.

4.3.1 Distribuição uniforme não-aleatória

Na distribuição uniforme não-aleatória, os nós foram distribuídos distando 40 metros entre si. A Figura 4.1 ilustra a distribuição uniforme não-aleatória. O nó 49 atuou como sorvedouro; todos os outros nós atuaram como fonte.

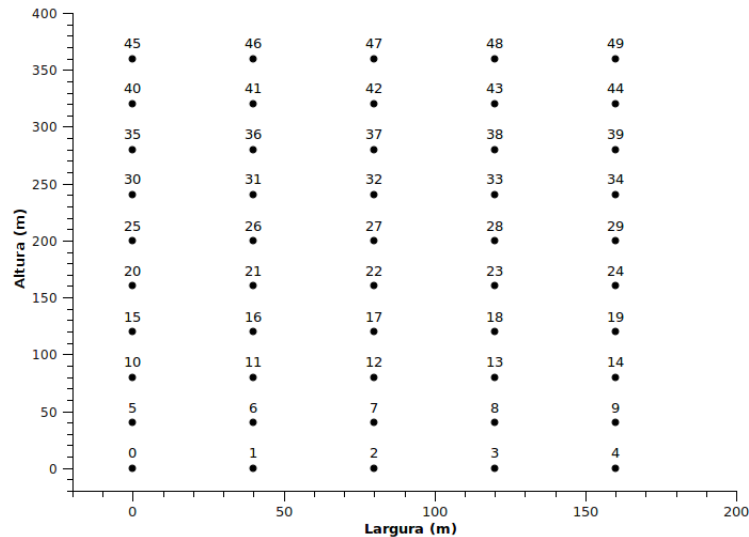


Figura 4.1: Topologia uniforme não-aleatória

O total de mensagens geradas pela rede para as 20 simulações foi de 4.233.363 para o DESRO e 4.223.242 para o ESDSR. A explicação para essa diferença de 10.121 mensagens a mais criadas quando o DESRO estava em uso é que, como será visto adiante, alguns nós esgotaram suas baterias durante as simulações com o ESDSR e, portanto, pararam de gerar pacotes de dados.

A taxa de entrega entre os protocolos foi semelhante, mostrando que o ESDSR consegue se adaptar rapidamente a mudanças na topologia da rede causadas por morte ou falha dos nós. Ao perceber uma quebra de link, o ESDSR altera a rota pela qual o dado é roteado não só para os novos pacotes como também para os pacotes já presentes na rede, quando se diz que o pacote foi “salvo”.

O DESRO, entretanto, não apenas gerou mais pacotes de dados como também os entregou mais ao sorvedouro. Foram 4.222.325 (99,74%) mensagens recebidas usando o DESRO e 4.160.242 (98,51%) usando o ESDSR, não computando mensagens repetidas. A Figura 4.2 exibe os resultados por simulação.

A maior taxa de entrega por parte do DESRO é justificada pelo melhor roteamento dos dados devido ao gerenciamento de rotas. O ESDSR, em alguns momentos, usou uma rota até que um dos nós da rota esgotasse completamente sua energia, para só então alterar sua escolha.

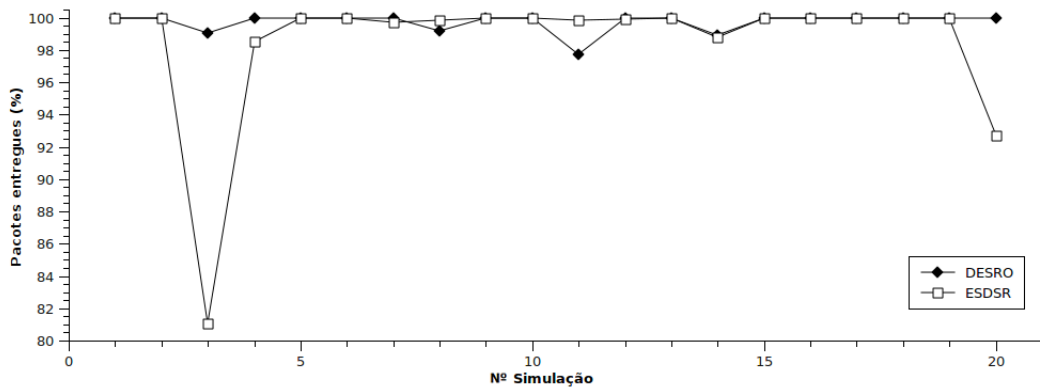


Figura 4.2: Taxa de entrega em cada simulação

Esse esgotamento de bateria do nó prejudicou o roteamento e, conseqüentemente, a taxa de entrega, apesar da boa resposta do ESDSR a mudanças na topologia da rede. É possível notar, pelos picos negativos na Figura 4.2, como a taxa de entrega no ESDSR foi prejudicada pela morte dos nós.

Por outro lado, o gerenciamento de rotas consome tempo e banda da rede, aumentando o tempo de resposta da rede, isto é, o tempo desde a criação de um pacote até sua chegada ao sorvedouro. Os tempos médios foram de 3,645 segundos para o DESRO e 2,95 segundos para o ESDSR. Desse modo, o DESRO obteve tempo de resposta cerca de 23,6% maior. É importante observar que o alto tempo médio, quando comparados a outros estudos, se deve principalmente à escolha do T-MAC como protocolo da camada MAC.

O T-MAC deriva do S-MAC (*Sensor MAC*) (YE; HEIDEMANN; ESTRIN, 2002), um protocolo que utiliza a estratégia de desligar o rádio periodicamente (*duty cycle*) alternando estados acordado-dormindo para obter economia de energia. Assim, é inerente ao S-MAC e T-MAC um *delay* conhecido por *sleep delay*, que ocorre quando um nó recebe um pacote e está pronto para retransmití-lo mas tem de aguardar o nó destino acordar. O valor médio para o *sleep delay* D_s por salto é dado pela equação (4.2) (YE; HEIDEMANN; ESTRIN, 2002):

$$D_s = \frac{T_{frame}}{2} \quad (4.2)$$

e, como já dito, o valor de quadro de tempo (T_{frame}) usado neste estudo foi de 610 milissegundos. É possível, através do ajuste de valor do quadro de tempo, conseguir tempos de resposta menores, mas este não é foco do presente estudo.

O objetivo principal da adição do gerenciamento de rotas ao ESDSR é melhorar a distri-

buição de energia na rede, impedindo que alguns nós fiquem sobrecarregados com relação ao roteamento de pacotes e acabem esgotando completamente suas baterias.

Quando um nó esgota sua bateria, ele deixa de ter qualquer atividade na rede e, em alguns casos, particiona a rede, originando regiões sem qualquer meio de comunicação com o servidor. Quando isso acontece, a taxa de entrega é prejudicada e a rede passa a não desempenhar completamente seu papel. As Figuras 4.3 e 4.4 ilustram a distribuição média de energia obtida a partir das 20 simulações feitas para o DESRO e ESDSR respectivamente.

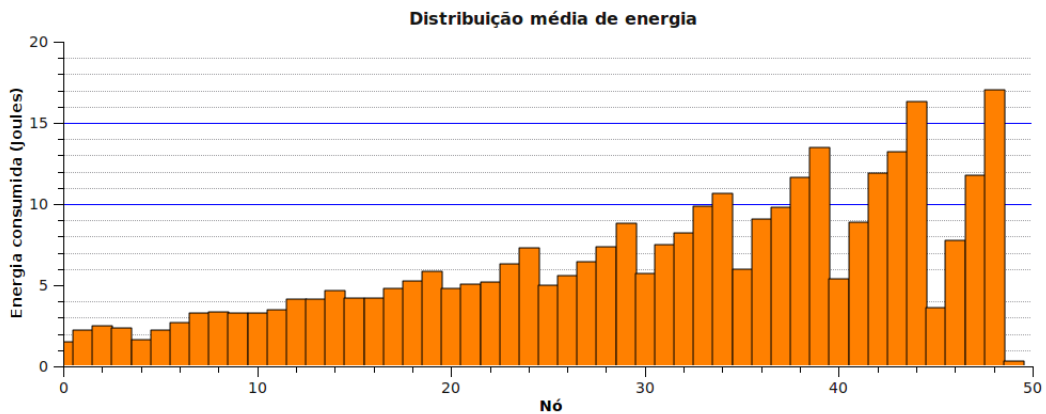


Figura 4.3: Distribuição média de energia no DESRO

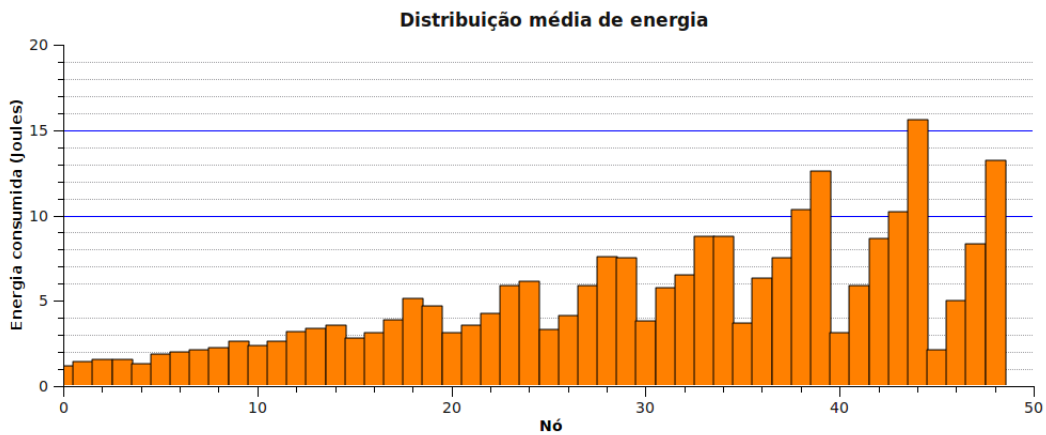


Figura 4.4: Distribuição média de energia no ESDSR

Pelos gráficos de distribuição média tem-se a impressão de que o ESDSR produz uma distribuição tão boa quanto o DESRO - talvez até melhor. O que acontece de fato é que a média aritmética camufla a grande variação na distribuição de energia verificada durante simulações do ESDSR.

A Tabela 4.2 fornece os valores de desvio-padrão e variância para o consumo de energia calculados para os nós marcados na Figura 4.5. A escolha destes nós se deve ao fato de que

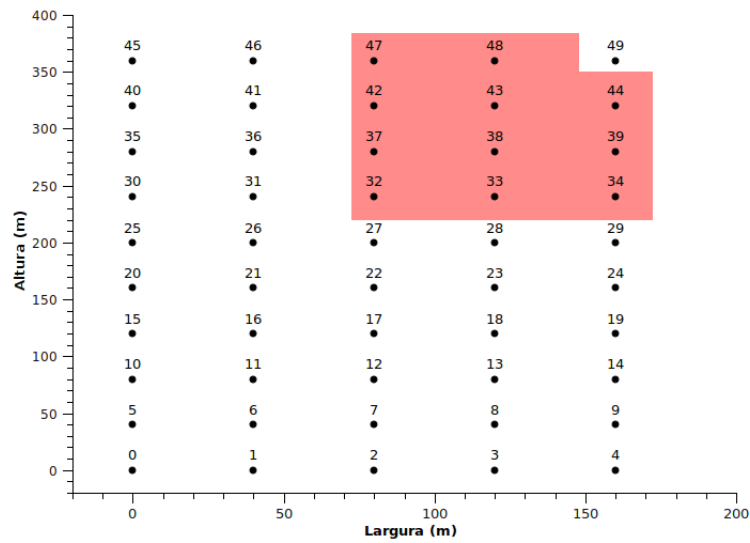


Figura 4.5: Área com maior consumo de energia.

eles são responsáveis pelo roteamento de grande quantidade de pacotes, estão mais próximos do sorvedouro e consomem maior quantidade de energia formando, portanto, a área onde o gerenciamento de rotas precisa otimizar de forma mais efetiva a escolha do caminho por onde o pacote de dados deverá ser transmitido.

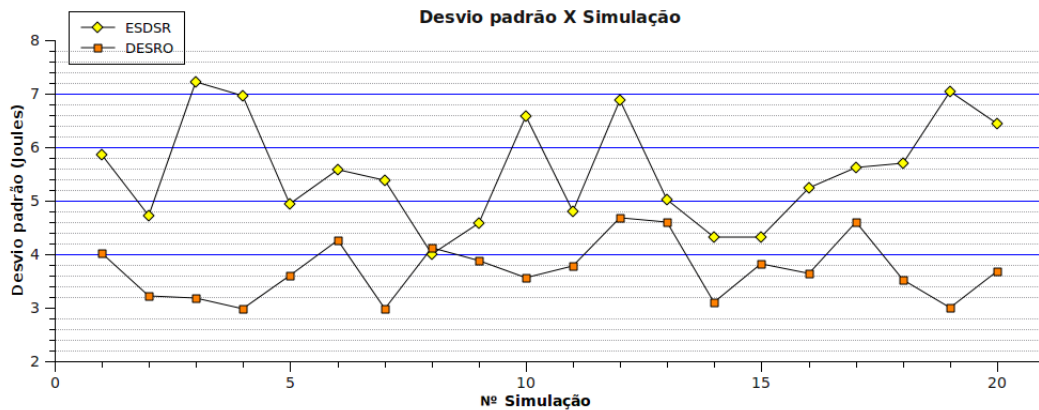


Figura 4.6: Desvio-padrão do consumo de energia nas simulações.

Conhecendo os valores de desvio-padrão para o consumo de energia nos nós, percebe-se uma clara melhoria da distribuição causada pelo mecanismo de gerenciamento de rotas. O DESRO possui um desvio menor em quase todas as simulações e a variabilidade dos valores é baixa, enquanto que no ESDSR a variabilidade é acentuada. Isso mostra que o DESRO foi capaz de manter os nós consumindo quantidades similares de energia ao alternar as rotas utilizadas. O gráfico 4.6 ilustra como o desvio-padrão para o consumo de energia obtido com o DESRO é significativamente menor (49,90%).

Simulação	ESDSR-Desvio	DESRO-Desvio	ESDSR-Variância	DESRO-Variância
1	5,8536	4,0257	34,2642	16,2064
2	4,7187	3,2153	22,2663	10,3381
3	7,2116	3,1739	52,0069	10,0739
4	6,9687	2,9785	48,5632	8,8714
5	4,9363	3,6053	24,3670	12,9985
6	5,5833	4,2611	31,1733	18,1567
7	5,3727	2,9794	28,8654	8,8770
8	3,9991	4,1122	15,9931	16,9098
9	4,5826	3,8739	20,9999	15,0068
10	6,5752	3,5573	43,2327	12,6543
11	4,8030	3,7707	23,0688	14,2183
12	6,8770	4,6746	47,2930	21,8522
13	5,0247	4,5916	25,2478	21,0827
14	4,3187	3,0944	18,6513	9,5752
15	4,3126	3,8286	18,5985	14,6580
16	5,2373	3,6340	27,4297	13,2059
17	5,6120	4,5909	31,4945	21,0766
18	5,7074	3,5200	32,5746	12,3904
19	7,0494	3,0091	49,6933	9,0544
20	6,4439	3,6802	41,5244	13,5441
Média	5,5594	3,7088	31,8654	14,0375

Tabela 4.2: Desvio-padrão e variância em cada simulação.

Pela Figura 4.1, percebe-se que qualquer pacote de dados deve passar necessariamente pelo nó 44 ou pelo nó 48 para chegar ao sorvedouro, já que são os únicos nós cujo raio de transmissão alcança o nó 49. A análise do consumo de energia desses nós fornece uma idéia mais clara do importante papel desempenhado pelo gerenciamento de rotas.

No ESDSR uma rota é usada até que um nó participante da rota esgote sua bateria e só então outra rota é usada, a menos que uma nova rota seja descoberta e sua energia estimada seja maior que a energia inicialmente estimada para a rota em uso. Isso fez com que o balanceamento de carga de roteamento e de consumo de energia entre os nós 44 e 48 fosse inexistente. Isso pode ser verificado na Figura 4.7.

Devido à falta de gerenciamento de rotas, o ESDSR foi incapaz de impedir a morte de nós durante as simulações. No total de 20 simulações, houve 13 mortes devido ao esgotamento de bateria. Por outro lado o DESRO foi plenamente capaz de impedir a morte dos nós da rede, não havendo morte por esgotamento de bateria em nenhuma das simulações. O maior consumo verificado ao fim das simulações em um nó usando o DESRO foi de 18,94 Joules, consumindo assim 94,7% da energia disponível.

O mecanismo de gerenciamento de rotas se mostrou eficaz no balanceamento de carga de

roteamento e de consumo de energia. O DESRO conseguiu que o consumo de energia dos nós 44 e 48 se mantivessem semelhantes estendendo, dessa forma, a vida da rede, além de aumentar o taxa de entrega de pacotes. Isso pode ser verificado na Figura 4.8.

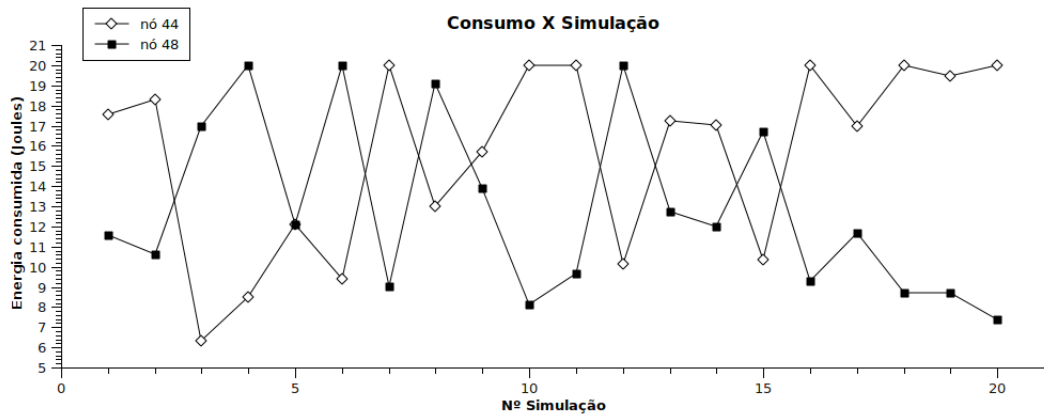


Figura 4.7: Consumo de energia dos nós 44 e 48 no ESDSR.

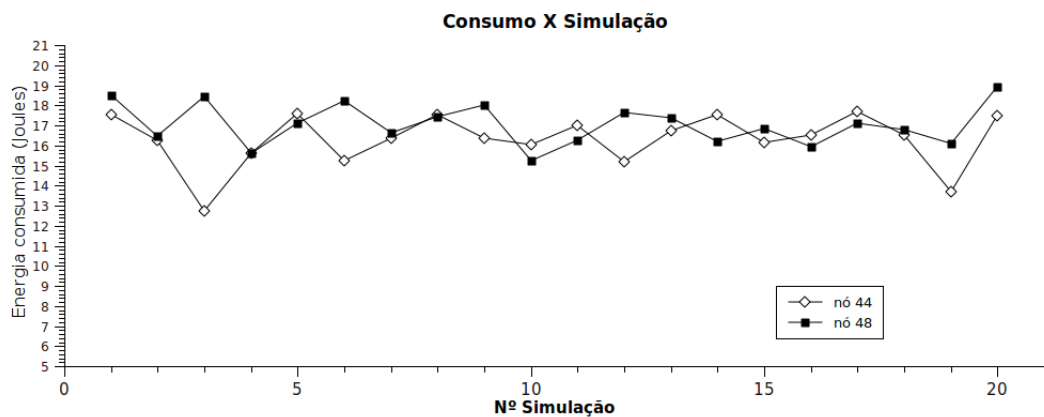


Figura 4.8: Consumo de energia dos nós 44 e 48 no DESRO.

O consumo médio de energia gasta com pacotes de controle foi de 12,34 Joules para o ESDSR e 9,59 Joules para o DESRO, mostrando que o DESRO consumiu 28,68% menos energia com controle. Isso pode ser explicado pelo fato de que quando um nó morre o ESDSR e DESRO geram pacotes de controle a fim de que a nova topologia seja conhecida pelos outros nós. As rotas que continham o nó morto são destruídas e novas rotas precisam ser descobertas através de mensagens *broadcast* usando algoritmo de inundação, consumindo assim muita energia. Como o DESRO evitou a morte dos nós o tráfego de pacotes de controle foi menor, mesmo computando os pacotes de controle usados no gerenciamento das rotas.

O consumo do DESRO com pacotes de dados foi cerca de 30% maior (71,52 Joules). Este maior consumo é natural e causado por dois fatores: o DESRO gerou 10.121 pacotes de dados

a mais do que o ESDSR; o total de mensagens de dados repetidas recebidas pelo sorvedouro foi de 453.667 para o DESRO e 229.755 para o ESDSR, uma diferença de 223.912 mensagens. O maior tráfego causado pelo gerenciamento de rotas fez com que o protocolo MAC tivesse mais trabalho durante as transmissões, o que resultou um maior número de retransmissões.

Deve-se observar que os nós iniciam uma transmissão de dados ao sorvedouro a cada 20 segundos, o equivalente a 10 minutos na escala de tempo usada, e que o tempo médio de resposta dos protocolos ficou em torno de 3 segundos, sendo este último valor em tempo real, sem escala. Ou seja, o maior tráfego gerado pelo DESRO não alteraria o número de retransmissões numa aplicação real, já que o tempo de 3 segundos na aplicação real equivaleria a 0,5% do tempo entre novas transmissões de dados, enquanto que nas simulações representou 15%.

4.3.2 Distribuição uniforme aleatória

No cenário com distribuição uniforme aleatória, a posição dos nós foi definida de forma aleatória pelo simulador e o sorvedouro (nó 49) foi posicionado no centro da área a fim de aumentar o número de nós capazes de alcançá-lo. Foram realizadas 20 simulações usando a topologia uniforme aleatória ilustrada na Figura 4.9

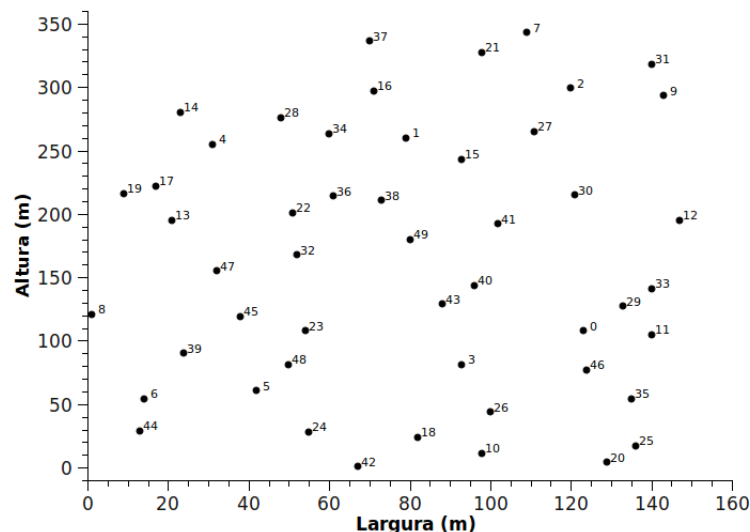


Figura 4.9: Topologia uniforme aleatória

O DESRO gerou, no total de 20 simulações, 4.232.961 mensagens de dados; o ESDSR gerou 4.213.059. A explicação para essa diferença de 19.902 mensagens é, mais uma vez, a morte de nós por esgotamento de bateria. Uma vez mortos, os nós deixam de gerar pacotes de dados.

As taxas de entrega dos protocolos foram semelhantes entre si na distribuição uniforme

aleatória, com uma pequena vantagem para o DESRO. O DESRO foi capaz de entregar 99,50% dos pacotes de dados criados contra 99,19% do ESDSR, não computando mensagens repetidas. A Figura 4.10 ilustra a taxa de entrega para cada simulação.

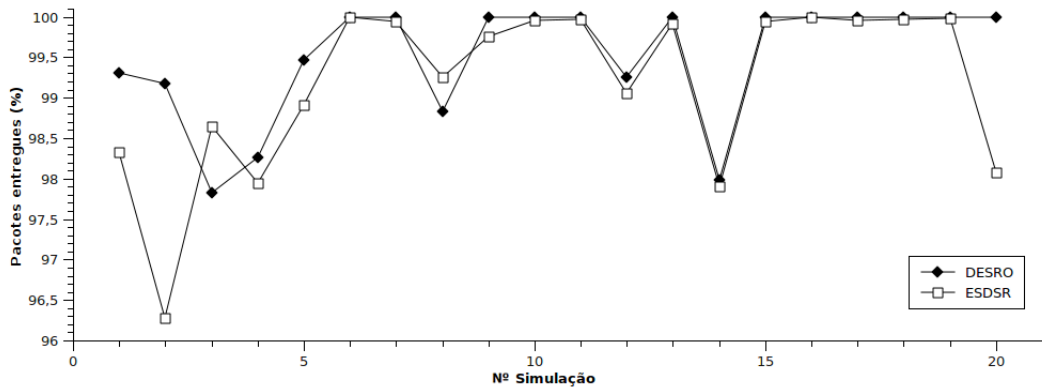


Figura 4.10: Taxa de entrega em cada simulação

Mais uma vez o gerenciamento de rotas do DESRO influenciou positivamente a taxa de entrega de dados. O ESDSR teve sua taxa de entrega prejudicada pela morte de nós, assim como nas simulações com distribuição uniforme. Foram, no total de 20 simulações, 25 mortes de nós quando o ESDSR estava em uso contra apenas 2 mortes do DESRO.

Como já foi visto, o gerenciamento de rotas consome tempo e por isso o tempo de resposta médio do DESRO, 2,56 segundos, foi 20,2% maior que o do ESDSR, cujo tempo foi de 2,13 segundos.

A distribuição média de energia encontrada no DESRO e ESDSR são semelhantes, com uma ligeira vantagem para o DESRO. As Figuras 4.13 e 4.12 ilustram as distribuições média de energia obtidas para o DESRO e ESDSR, respectivamente. Entretanto, como já visto, a média aritmética pode camuflar uma alta variabilidade dos valores, o que de fato ocorreu.

Os valores de desvio-padrão e variância para o consumo de energia em cada simulação, que podem ser vistos na Tabela 4.3, mostram que a distribuição de energia gerada pelo DESRO sofreu uma menor variância. Os valores médios de desvio-padrão são de 3,4520 Joules para o ESDSR e 2,1091 Joules para o DESRO. Desse modo o desvio-padrão do DESRO foi 38,9% menor.

Para o cálculo do desvio-padrão e variância, foram considerados os nós contidos em um círculo com raio de 45 metros (alcance máximo do rádio) e centro no sorvedouro (nó 49) - Figura 4.11; esses são os nós capazes de transmitir diretamente ao sorvedouro.

O gerenciamento de rotas foi capaz de diminuir o número de morte de nós devido ao con-

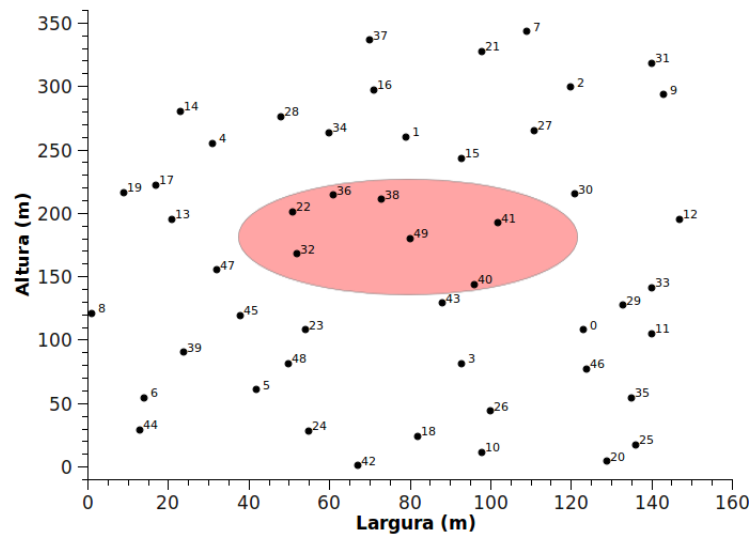


Figura 4.11: Nós capazes de transmitir diretamente ao sorvedouro

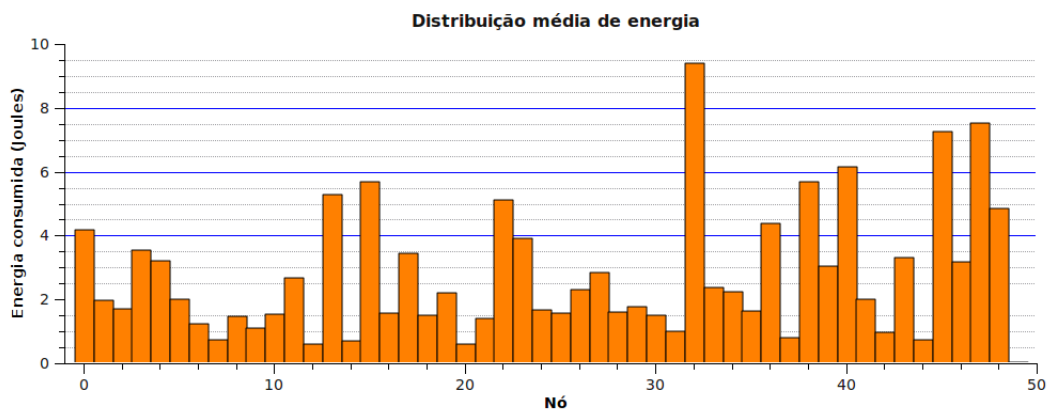


Figura 4.12: Distribuição média de energia no ESDSR

sumo de energia porém aumentou o consumo total de energia na rede. A energia média gasta com controle no ESDSR foi de 1,12 Joules enquanto que o DESRO gastou em média 7,02 Joules com pacotes de controle, o que representa um aumento de mais de 600%. O aumento se deu também com os pacotes de dados; o ESDSR gastou 134,9 Joules com dados enquanto que o DESRO teve um gasto de 165,13 Joules, um aumento de cerca de 22,41%.

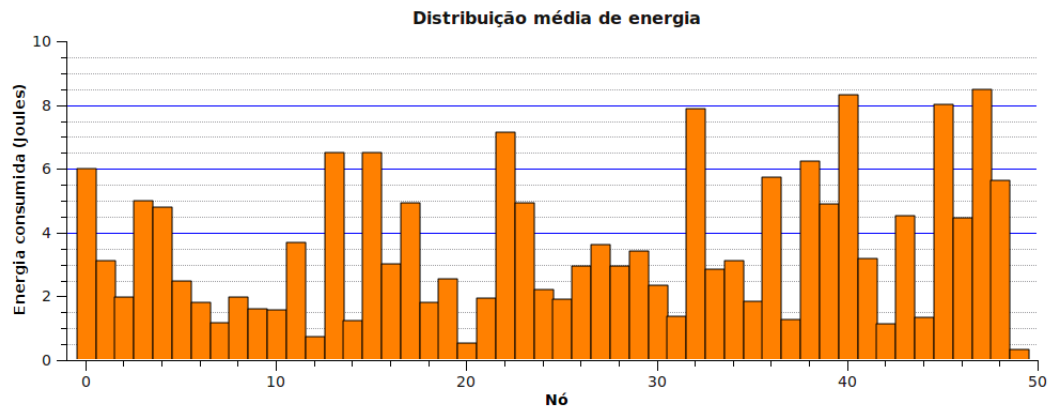


Figura 4.13: Distribuição média de energia no DESRO

Simulação	ESDSR-Desvio	DESRO-Desvio	ESDSR-Variância	DESRO-Variância
1	3,3251	2,2450	11,0560	5,0401
2	3,6205	2,6933	13,1078	7,2541
3	2,3501	1,0080	5,5236	1,0151
4	3,6798	1,7842	13,5410	3,1834
5	3,4263	2,1832	11,7392	4,7662
6	4,0339	2,2079	16,2725	4,8747
7	4,0104	2,5718	16,0833	6,6139
8	3,0656	1,2240	9,3976	1,4981
9	3,2382	2,2821	10,4859	5,2078
10	3,7596	1,5633	14,1342	2,4439
11	4,1528	2,9108	17,2454	8,4728
12	3,6942	1,9616	13,6468	3,8478
13	3,2616	1,8697	10,6380	3,4959
14	4,1766	2,7791	17,4442	7,7236
15	3,6281	2,1488	13,1632	4,6174
16	3,1342	1,8205	9,8233	3,3140
17	3,3098	1,1409	10,9163	1,3016
18	3,1723	2,3873	10,0634	5,6991
19	3,3307	2,8122	11,0936	7,9085
20	2,6692	2,5875	7,1249	6,6952
Média	3,4520	2,1091	12,1250	4,7487

Tabela 4.3: Desvio-padrão e variância em cada simulação.

5 *Conclusões e Perspectivas Futuras*

Neste trabalho foi apresentado o protocolo DESRO, que consiste de alterações do protocolo ESDSR para a inclusão de funções de gerenciamento de energia na definição do roteamento. A inclusão de gerenciamento de rotas ao ESDSR teve por objetivo postergar a morte de nós através de um balanceamento de carga de roteamento e produzir uma melhor distribuição de energia na rede. Pode-se inferir, através das simulações realizadas, que o objetivo foi alcançado.

Os resultados das simulações mostram que o DESRO obteve taxas de entrega melhores que as taxas obtidas com o ESDSR. O DESRO evitou completamente a morte de nós na distribuição uniforme não-aleatória, não havendo nenhum nó inativo por falta de energia ao final das 20 simulações, tendo apenas dois nós mortos nas simulações com distribuição uniforme aleatória. O ESDSR por sua vez teve 13 nós mortos na distribuição uniforme não-aleatória e 25 na distribuição uniforme aleatória.

O gerenciamento de rotas consome mais energia do ponto de vista de controle mas, por outro lado, economiza ao evitar o envio de novos pacotes de descoberta de rotas. Em algumas simulações a economia de energia alcançada por não enviar pacotes de descoberta de rotas compensou o consumo com pacotes de gerenciamento, em até 100%.

O DESRO consumiu mais energia que o ESDSR com a transmissão de pacotes de dados e esse aumento no consumo é explicado por dois motivos principais: o DESRO gerou mais mensagens de dados, tanto na distribuição uniforme não-aleatória quanto na distribuição uniforme aleatória; a escala de tempo usada na simulação fez com que o tempo gasto com o gerenciamento de rotas fosse desproporcional ao período de envio de mensagens de dados, havendo assim mais retransmissões de pacotes. É de se esperar que em uma aplicação real de monitoramento, usando os mesmos parâmetros do presente estudo, com exceção da escala de tempo, o DESRO consuma, com pacotes de dados, a mesma quantidade de energia que o ESDSR.

Apesar dos bons resultados apresentados o DESRO pode se tornar ainda mais eficiente. Alterações no gerenciamento de rotas podem gerar um melhor balanceamento de carga consumindo os mesmos recursos utilizados pelo gerenciamento atual. O DESRO utiliza uma estima-

tiva de carga simples, atualizando o contador de uso da rota apenas para os pacotes de dados gerados pelo próprio nó. Uma abordagem mais precisa, contando também os pacotes roteados que utilizaram a rota, pode realizar uma melhor distribuição de energia por possuir uma melhor estimativa de carga.

A atualização de energia estimada na tabela de rotas interfere apenas na escolha de rotas para pacotes de dados gerados pelo próprio nó, ou seja, não interfere no roteamento de outros pacotes de dados. É possível que uma alteração na decisão de roteamento, agora usando informações da sua própria tabela de energia de rotas, leve a uma economia e melhor distribuição de energia, por permitir que a rota especificada no cabeçalho do pacote de dados seja alterada por um nó cuja tabela de energia estimada seja mais atual do que a tabela do nó gerador do pacote.

A distribuição de energia alcançada pelo gerenciamento de rotas varia com a posição do sorvedouro. Se houver apenas um caminho para o sorvedouro, o gerenciamento de rotas poderá ter uma influência ruim sobre a rede porque não conseguirá balancear a carga de roteamento, além de aumentar o consumo de energia com pacotes de controle. Por outro lado, se houver vários caminhos ou ainda vários sorvedouros, o gerenciamento de rotas poderá trazer um grande benefício ao distribuir a carga por várias rotas. Assim, o impacto do uso de mais de um sorvedouro numa rede de sensores usando o DESRO como protocolo de roteamento deve ser estudado.

As simulações realizadas neste trabalho consideraram os nós ativos em qualquer instante após o seu despertar. A utilização de um ciclo de trabalho diferente, onde os nós alternam estados de atividade e inatividade, pode trazer uma redução no consumo de energia da rede. Um estudo acerca do impacto do ciclo de trabalho dos nós no DESRO poderia revelar uma melhoria na eficiência do protocolo.

Por fim, vale notar que o DESRO lista em seu cabeçalho todos os identificadores dos nós da rota a ser trafegada. Isso limita o número de nós na rede, seu raio e a distância máxima (em saltos) entre o nó mais distante e o sorvedouro. Se o número de saltos entre um nó e o sorvedouro for grande, o *overhead* do protocolo se torna impraticável; portanto, essa característica do DESRO deve ser melhorada. O DESRO foi criado com o objetivo de trazer para as RSSFs um eficiente protocolo das MANETs; com a correção da falta de gerenciamento de rotas do ESDSR, o DESRO já pode ser utilizado satisfatoriamente em RSSFs. Um próximo passo é a alteração do seu modo de listagem de rota no pacote de dados, diminuindo assim o *overhead* no roteamento.

Referências Bibliográficas

- AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, Elsevier, v. 3, n. 3, p. 325–349, May 2005.
- AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. *Computer Networks*, Elsevier, v. 38, n. 4, p. 393–422, March 2002.
- APL. *Academic Public License*. 2009. [Online; acessado 31-Outubro-2009]. Disponível em: <<http://www.omnetpp.org/home/license>>.
- CASTALIA. *National ICT Australia – Castalia*. 2009. [Online; acessado 31-Outubro-2009]. Disponível em: <<http://castalia.npc.nicta.com.au/>>.
- DAM, T. van; LANGENDOEN, K. An adaptive energy-efficient mac protocol for wireless sensor networks. In: *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. [S.l.]: ACM, 2003. p. 171–180.
- DOSHI, S.; BHANDARE, S.; BROWN, T. X. An on-demand minimum energy routing protocol for a wireless ad hoc network. In: *ACM SIGMOBILE Mobile Computing and Communications Review*. [S.l.]: IEEE, 2002. v. 6, n. 3, p. 50–66.
- FIGUEIREDO, C. M. S.; LOUREIRO, A. A. F.; NAKAMURA, E. F. Protocolo adaptativo híbrido para disseminação de dados em redes de sensores sem fio auto-organizáveis. In: *22^o Simpósio Brasileiro de Redes de Computadores*. [S.l.: s.n.], 2004.
- GONÇALVES, R. T. et al. Multi-k: Um protocolo de roteamento para redes de sensores sem fio usando Árvores de espalhamento parciais. In: *XXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. [S.l.: s.n.], 2008. p. 1–14.
- HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*. [S.l.]: IEEE, 2000. p. 908–918.
- HEINZELMAN, W. R.; KULIK, J.; BALAKRISHNAN, H. Adaptive protocols for information dissemination in wireless sensor networks. In: *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’99)*. [S.l.]: ACM, 1999. p. 174–185.
- INTANAGONWIWAT, C.; GOVINDAN, R.; ESTRIN, D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’00)*. [S.l.]: ACM, 2000. p. 56–67.

- JOHNSON, D. B.; MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. In: IMIELINSKI, T.; KORTH, H. F. (Ed.). *Mobile Computing*. [S.l.]: Kluwer Academic Publishers, 1996, (The International Series in Engineering and Computer Science, v. 353). p. 153–181.
- KARL, H.; WILLIG, A. *Protocols and architectures for wireless sensor networks*. [S.l.]: Wiley, 2007. 524 p. ISBN 978-0-470-51923-3.
- LOUREIRO, A. A. F. et al. Redes de sensores sem fio. *21º Simpósio Brasileiro de Redes de Computadores, Natal, RN, Brasil, 2003*.
- LUA. *Lua Programming Language*. 2010. [Online; acessado 13-Janeiro-2010]. Disponível em: <<http://www.lua.org/>>.
- MANJESHWAR, A.; AGRAWAL, D. P. Teen: A protocol for enhanced efficiency in wireless sensor networks. In: *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*. [S.l.]: IEEE, 2001. p. 1567–1576.
- NS-2. *The network simulator - NS-2*. 2009. [Online; acessado 08-Dezembro-2009]. Disponível em: <<http://www.isi.edu/nsnam/ns/>>.
- OMNET++. *OMNeT++ Community Site*. 2009. [Online; acessado 31-Outubro-2009]. Disponível em: <<http://www.omnetpp.org/>>.
- SARANGAPANI, J. *Wireless Ad Hoc and Sensor Networks - Protocols, Performance, and Control*. Boca Raton, FL: CRC Press, 2007. 536 p. (Control Engineering). ISBN 0-8247-2675-8.
- SHEE, S. H.; WANG, K.; HSIEH, Y. L. Color-theory-based dynamic localization in mobile wireless sensors networks. In: *Proceedings of Workshop on Wireless, Ad Hoc and Sensor Networks (WASN05)*. [S.l.: s.n.], 2005. p. 73–78.
- SILVA, A. P.; GOULART, C. de C. Multi-geo: um protocolo de roteamento hierárquico para redes de sensores sem fio. In: *XXXV Conferência Latinoamericana de Informática*. [S.l.: s.n.], 2009. p. 1–10.
- SOHRABY, K.; MINOLI, D.; ZNATI, T. *Wireless sensor networks: technology, protocols and applications*. [S.l.]: Wiley, 2007. 328 p. ISBN 978-0-471-74300-2.
- SRINIVASAN, K.; LEVIS, P. Rssi is under appreciated. In: *Proceedings of the 3rd Workshop on Embedded Networked Sensors (EmNet's 2006)*. [S.l.: s.n.], 2006.
- TARIQUE, M.; TEPE, K. E.; NASERIAN, M. Energy saving dynamic source routing for ad hoc wireless networks. In: *Proceedings of 3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*. [S.l.]: IEEE, 2005. p. 305–310.
- WU, S.-L.; TSENG, Y.-C. *Wireless Ad Hoc Networking: Personal-Area, Local-Area, and the Sensory-Area Networks*. [S.l.]: CRC Press, 2007. 664 p. ISBN 978-0-849-39254-2.
- XU, Y.; HEIDEMANN, J.; ESTRIN, D. Geography-informed energy conservation for ad hoc routing. In: *Proceedings of the 7th annual international conference on Mobile computing and networking*. [S.l.]: ACM, 2001. p. 70–84.

YE, W.; HEIDEMANN, J.; ESTRIN, D. An energy-efficient mac protocol for wireless sensor networks. In: *Proceedings of the 21st Conference of the IEEE Computer and Communications Society (INFOCOM)*. [S.l.]: IEEE, 2002. p. 1567–1576.