

**UM PROTOCOLO PARA RECONFIGURAÇÃO DE
REDES DE SENSORES SEM FIO**

JOÃO GUILHERME MAIA DE MENEZES

**UM PROTOCOLO PARA RECONFIGURAÇÃO DE
REDES DE SENSORES SEM FIO**

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

ORIENTADOR: ANTONIO ALFREDO FERREIRA LOUREIRO

COORIENTADOR: ANDRÉ LUIZ LINS DE AQUINO

Belo Horizonte

Outubro de 2009

© 2009, João Guilherme Maia de Menezes.
Todos os direitos reservados.

Menezes, João Guilherme Maia de

M534p Um protocolo para reconfiguração de redes de
sensores sem fio / João Guilherme Maia de Menezes —
Belo Horizonte, 2009
xxii, 42 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais - Departamento de Ciência da
Computação

Orientador: Antonio Alfredo Ferreira Loureiro
Coorientador: André Luiz Lins de Aquino

1. Computação - Teses. 2. Redes de sensores sem
fio - Teses. 3. Redes de computadores - Protocolos -
Teses. I. Orientador. II. Coorientador. III. Título.

519.6*22(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Um protocolo para reconfiguração de redes de sensores sem fio

JOÃO GUILHERME MAIA DE MENEZES

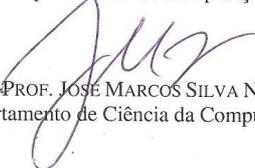
Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. ANTONIO ALFREDO FERREIRA LOUREIRO - Orientador
Departamento de Ciência da Computação - UFMG


PROF. ANDRÉ LUIZ LINS DE AQUINO - Co-orientador
Departamento de Computação - UFOP


PROFA. RAQUEL APARECIDA DE FREITAS MINI
Departamento de Ciência da Computação - PUC-MG


PROFA. ROSSANA MARIA DE CASTRO ANDRADE
Departamento de Computação - UFC


PROF. JOSÉ MARCOS SILVA NOGUEIRA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 15 de outubro de 2009.

Aos meus pais, João e Cida, e à minha irmã, Mirella, as pessoas mais importantes em minha vida.

Agradecimentos

Agradeço acima de tudo aos meus pais pelo amor, carinho, educação e formação do meu caráter. À minha irmã, a pessoa mais incrível que eu conheço.

Agradeço ao meu professor e orientador, Antonio Alfredo Ferreira Loureiro, pelos grandes ensinamentos, dedicação, honestidade, paciência e grande compreensão. Agradeço pela pessoa que é, tornando o trabalho ao seu lado muito mais divertido.

Agradeço ao meu co-orientador, Andre Luiz Lins de Aquino (Alla), pela grande dedicação, encorajamento, paciência, compreensão e amizade. Suas sugestões foram valiosas e tornaram esse trabalho possível.

Agradeço aos membros da banca, Rossana Andrade, Raquel Mini e José Marcos Nogueira pelas valiosas sugestões que tornaram esse trabalho mais interessante.

Agradeço aos melhores amigos que alguém pode ter, meus grandes amigos da UFMG e BH. Leandro, Guidoni, Sumika, Heitor, Karina, Marcelo, Lorena, Felipe, Alyson (Robinho), Bruno, Celso, Fernando, Gustavo, Holiver, Flávio, Dudu, Dudu Mucelli, Pedro, Max, DG, Verônica e Letícia. Grandes amigos de BH, nunca imaginei que conheceria pessoas tão especiais como vocês, tornando minha vida longe da minha família e amada terra menos sofrível.

Agradeço à Luana, uma das garotas mais incríveis que conheço. Estar ao seu lado é sempre motivo de pura felicidade.

Agradeço à Rose K. Bandida Master por ser tão especial. Sua amizade é única e os momentos de alegria são constantes. Às minhas amigas de Maceió, Marcelinha, Ruivinha, Sibaldinha, Ninha, Ella e Fran. Aos meus amigos Roberto, Badu, Thiago Bruno, Ícaro, Manoel, Mário e Ivo. Vocês estão distantes, mas não esquecidos.

Agradeço às bandas Radiohead, Gun's N' Roses, Pearl Jam, Audioslave, The Killers, Iron Maiden, The White Stripes e muitas outras pelas grandes composições que serviram de inspiração no desenvolvimento desse trabalho. À Tequila pelas noites mágicas em BH.

Agradeço ao DCC e à UFMG pela grande universidade.

“I don’t want to learn what I’ll need to forget”
(Audioslave)

Resumo

Reprogramação é um problema importante e desafiador para as redes de sensores sem fio já que usualmente os *softwares* dos sensores precisam ser reprogramados. Nessa direção, *Over-the-Air Programming* é um serviço fundamental que faz uso de *broadcast* confiável para a distribuição eficiente dos dados. Neste trabalho, é proposto o *Over-the-Air Programming Protocol based on Small World (OAP-SW)*, um novo protocolo de reconfiguração para redes de sensores sem fio que faz uso de conceitos de redes *small world*. Esse protocolo considera o fluxo de comunicação de dados nas redes de sensores para a criação de atalhos em direção ao nó sorvedouro, tornando dessa forma o processo de reconfiguração mais eficiente. Os nós finais desses atalhos são nós com um *hardware* diferenciado, resultando assim em uma rede de sensores heterogênea. Além disso, realizou-se a extensão do protocolo proposto para suportar *multicast*, permitindo assim que apenas um subconjunto dos nós da rede sejam reconfigurados, funcionalidade essa que não está presente nos protocolos existentes na literatura. Resultados de simulação mostram a eficiência do protocolo em relação ao número de mensagens transmitidas e o tempo gasto para realizar a reconfiguração, apresentando resultados satisfatórios.

Abstract

Reprogramming is an important and challenging problem in wireless sensor networks because it is often necessary to in-network sensor reprogramming. Thus, Over-the-Air Programming is a fundamental service that relies upon reliable broadcast for efficient distribution. In this work, it is proposed the *Over-the-Air Programming Protocol based on Small World (OAP-SW)*, a novel protocol for wireless sensor network reprogramming that uses small world features. The small world based protocol takes into account the communication workflow of sensor networks to create shortcuts toward the sink, thus improving the reprogramming process. The endpoints of these shortcuts are nodes with more powerful hardware, resulting in a heterogeneous wireless sensor network. Furthermore, the proposed protocol was extended to perform multicast communication in order to allow the reprogramming of a subset of the nodes in the sensor network. The multicast is not performed by other reprogramming protocols. Simulation results showed the feasibility of the protocol regarding the number of messages transmitted and the time to reconfigure the network, presenting satisfactory results.

Lista de Figuras

2.1	Divisão do programa em páginas e pacotes	8
2.2	Pipelining de dois segmentos em uma rede de quatro saltos.	9
3.1	Angulação em direção ao nó sorvedouro	15
3.2	Criação de atalhos considerando-se diferentes valores de p	17
3.3	Troca de mensagens do protocolo de reprogramação	18
3.4	Reprogramação sem pipelining	20
3.5	Reprogramação com uso de pipelining	20
3.6	Determinação dos vizinhos para repassar a mensagem <i>multicast</i>	23
4.1	Número de mensagens transmitidas para um cenário confiável	26
4.2	Tempo gasto na reprogramação da rede para um cenário confiável	27
4.3	Tempo gasto na reprogramação da rede para diferentes valores de densidade	28
4.4	Tempo gasto na reprogramação da rede para um cenário com perda de pacotes	30
4.5	Número de mensagens trocadas para redes com diferentes densidades . . .	31
4.6	Número de mensagens trocadas para diferentes quantidades de grupos <i>multicast</i>	32

Lista de Tabelas

2.1	Protocolos de reprogramação para RSSFs	7
4.1	Parâmetros de simulação.	26

Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Tabelas	xix
1 Introdução	1
1.1 Motivação	1
1.2 Definição do Problema	3
1.3 Objetivos	3
1.4 Organização do Trabalho	4
2 Trabalhos Relacionados	5
2.1 Desafios Para a Reprogramação em Redes de Sensores Sem Fio	5
2.2 Características dos Protocolos de Reprogramação	6
2.2.1 Pipelining	8
2.3 Deluge	10
2.3.1 Funcionamento do Protocolo Deluge	10
2.4 Comentários Finais	12
3 Solução Proposta	13
3.1 Introdução	13
3.2 Estabelecimento da Infra-estrutura	14
3.3 Protocolo de Reprogramação	17
3.3.1 Funcionamento do Protocolo OAP-SW	19
3.4 Reprogramação com Multicast	22

3.4.1	Motivação	22
3.4.2	Algoritmo Multicast	23
3.5	Comentários Finais	24
4	Resultados de Simulação	25
4.1	Introdução	25
4.2	Variando o Tamanho da Rede	26
4.3	Variando Probabilidade de Perda de Pacotes	29
4.4	Avaliação da Reprogramação com Multicast	30
4.5	Comentários Finais	33
5	Conclusões e Trabalhos Futuros	35
	Referências Bibliográficas	37

Capítulo 1

Introdução

1.1 Motivação

Nos últimos anos, evoluções tecnológicas nos sistemas microeletrônicos e na comunicação sem fio permitiram o desenvolvimento de pequenos dispositivos sensores, de baixo custo e de baixo consumo de energia. Diante de tal cenário, surgem as redes de sensores sem fio (RSSFs), redes compostas por um grande número de dispositivos com capacidade de realizar processamento, comunicação e sensoriamento distribuído [Akyildiz et al., 2002, Boukerche, 2008, Chong & Kumar, 2003, Estrin et al., 1999, Intanagonwiwat et al., 2000, Pottie & Kaiser, 2000a, Pottie & Kaiser, 2000b, Yick et al., 2008]. Essa nova tecnologia permite monitorar remotamente um ambiente e obter informações mais precisas do que as técnicas de instrumentação tradicionais, tirando vantagem do fato de ser empregada bem próxima do fenômeno de interesse.

Quando comparada às redes tradicionais, as RSSFs diferem em muitos aspectos. Por exemplo, as RSSFs possuem centenas ou mesmo milhares de nós, os quais apresentam fortes restrições de energia, já que uma bateria é normalmente utilizada como principal fonte de energia. Esses nós sensores são dispositivos autônomos com capacidade de sensoriamento, processamento e comunicação. Baseado em algum processo de decisão local, os dados sensoriados podem ser enviados para uma estação base (nó sorvedouro). Como os nós sensores possuem memória limitada e normalmente operam em regiões de difícil acesso ou mesmo inóspitas, uma interface de comunicação sem fio é utilizada para a transmissão dos dados para a estação base. Devido às limitações no alcance das interfaces desses dispositivos, a comunicação nas RSSFs é realizada em múltiplos saltos, ou seja, os nós devem cooperar no repasse de mensagens.

Uma grande vantagem das RSSFs é que estas possibilitam a coleta de informações

em ambientes onde o uso de fios ou cabeamento não são possíveis. Por exemplo, nós sensores podem ser embutidos em estruturas de prédios, pontes, tubulações, no interior de máquinas, florestas, áreas de desastres, plantações, vulcões, campos de batalha ou mesmo o próprio corpo humano. O baixo custo dos dispositivos utilizados nesse tipo de rede e o potencial desta tecnologia justificam a sua utilização em diversas áreas, tais como:

- Na área ambiental, as RSSFs podem ser utilizadas na prevenção e no auxílio às vítimas de catástrofes naturais como furacões, terremotos e vulcões, além de serem utilizadas na detecção de queimadas. Sistemas de detecção de incêndios podem auxiliar na redução dos estragos causados por queimadas. Essas redes também podem ser empregadas no rastreamento de pássaros, pequenos animais, mapeamento da bio-complexidade ambiental, dentre outras [Agre & Clare, 2000, Cerpa et al., 2001, Kahn et al., 1999, Mainwaring et al., 2002].
- Na área militar, as RSSFs podem ser utilizadas na detecção de armas químicas, como gás venenoso ou radiação. Além disso essas redes podem ser utilizadas na detecção de movimentos inimigos, monitoração de tropas e reconhecimento de terreno. Tais aplicações possuem fortes requisitos de segurança. Portanto, normalmente os dados são criptografados e submetidos a processos de assinatura digital [Simon et al., 2004, Yick et al., 2005].
- Na área industrial, as RSSFs podem ser utilizadas nos mecanismos de controle industrial. Por exemplo, sensores podem ser embutidos em máquinas em uma linha de montagem com o objetivo de detectar o mau funcionamento das mesmas. Além dessas aplicações, a detecção e vigilância de veículos e estabelecimentos, monitoração da qualidade de produtos são outros exemplos de como pode-se utilizar RSSFs na indústria [Warneke et al., 2001].
- Na área médica, as RSSFs podem ser utilizadas na monitoração de órgãos vitais como coração ou mesmo detectar a presença de substâncias que indicam a existência ou surgimento de um problema biológico, administração de drogas em hospitais e controle de doenças contagiosas [Gao et al., 2005, Lorincz et al., 2004].

Conforme apontado por Hu et. al. [Hu et al., 2004], a tendência é a produção dos nós sensores em larga escala, barateando o seu custo e o investimento no desenvolvimento tecnológico, levando a novas melhorias. Portanto, novas aplicações devem surgir, aumentando ainda mais a abrangência no uso de RSSFs. Uma característica desejável para tais aplicações é que as mesmas executem por longos períodos de tempo

de forma inatendida. No entanto, aplicações em RSSFs, assim como em qualquer outro sistema computacional, podem sofrer mudanças nos requisitos previamente imaginados, ocasionando, dessa forma, a necessidade de alterar o comportamento de tais aplicações por meio da introdução de novas funcionalidades.

1.2 Definição do Problema

Conforme destacado na seção anterior, as RSSFs costumam operar em ambientes remotos, de difícil acesso e até mesmo inóspitos. Ou seja, para que a aplicação de uma RSSF seja viável, ela deve operar de forma inatendida por longos períodos de tempo. Tal requisito introduz diversas dificuldades. Primeiramente, o ambiente pode evoluir com o tempo. Portanto, predizer todas as ações que um nó sensor deve realizar é uma tarefa muito difícil para muitas aplicações. E segundo, os requisitos das aplicações podem mudar em consequência de avanços tecnológicos ou devido ao melhor entendimento do ambiente por parte dos usuários.

Tais dificuldades, listadas anteriormente, tornam o processo de gerenciamento e manutenção das RSSFs uma tarefa desafiadora. Tornar as RSSFs reprogramáveis é uma forma de abordar tal desafio. Por exemplo, o *software* dos nós sensores pode sofrer atualizações após a instalação de uma RSSF para a correção de *bugs*, modificar atividades de nós individuais ou de todos os nós da rede, ou mesmo corrigir falhas de segurança.

Em consequência das características das RSSFs, a atividade de gerenciamento e manutenção dos nós de forma manual, visitando cada nó fisicamente, pode não ser viável ou mesmo impossível de ser realizada. Portanto, o problema tratado nesse trabalho é como desenvolver mecanismos que permitam a realização de tais atividades de gerenciamento e manutenção de forma remota.

1.3 Objetivos

Diante dos problemas supracitados, fica evidente que a forma de reprogramar manualmente as RSSFs é cara e trabalhosa, uma vez que cada nó deve ser coletado individualmente e fisicamente ser conectado a um computador para receber as modificações de código. Em consequência disso, pesquisas na área de RSSFs conceberam protocolos para reprogramação, utilizando-se da técnica de *Over-the-Air Programming*. Essa técnica permite a disseminação de código na rede, possibilitando a atualização de pro-

gramas, mudanças no funcionamento da rede ou modificar módulos de um programa de forma remota, ou seja, sem contato físico com o nó sensor.

No entanto, os protocolos para reprogramação de RSSFs existentes sofrem com problemas como grande número de mensagens trocadas durante a reprogramação, aumentando assim o consumo de energia e sacrificando o tempo de vida da rede, além de demandarem uma grande quantidade de tempo para a realização da reprogramação, inviabilizando assim a sua utilização, uma vez que os nós devem interromper sua aplicação durante a reprogramação.

Diante das limitações nas soluções existentes, o objetivo desse trabalho é o desenvolvimento de um protocolo de reprogramação para RSSFs que apresente baixo tempo de reprogramação, além de baixa quantidade no número de mensagens trocadas durante o processo de reprogramação. Nesse contexto, foi proposto o *Over-the-Air Programming Protocol based on Small World (OAP-SW)*, um novo protocolo de reprogramação para RSSFs que faz uso de conceitos de redes *small world*. Esse protocolo considera o fluxo de comunicação de dados nas RSSFs para a criação de *atalhos* em direção ao nó sorvedouro, tornando dessa forma o processo de reprogramação mais eficiente. Os nós finais desses *atalhos* são nós com um *hardware* diferenciado, resultando assim em uma RSSF heterogênea. Além disso, realizou-se a extensão do protocolo proposto para suportar *multicast*, permitindo assim que apenas um subconjunto dos nós da rede seja reprogramado, funcionalidade essa que não está presente nos protocolos existentes na literatura.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte forma. O capítulo 2 apresenta os principais protocolos para reprogramação em RSSFs existentes na literatura. O capítulo 3 apresenta o protocolo *Over-the-Air Programming Protocol based on Small World (OAP-SW)* e sua versão *multicast*. O capítulo 4 apresenta os resultados de simulação para cenários de rede confiáveis e com perda de pacotes e diferentes valores para densidade de rede. O algoritmo proposto é comparado ao protocolo estado da arte existente na literatura. Finalmente, o capítulo 5 apresenta as conclusões e os trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Este capítulo apresenta os principais trabalhos relacionados à reprogramação em RSSFs. Inicialmente, serão destacados os principais desafios relacionados à reprogramação em RSSFs. Em seguida, uma classificação dos principais protocolos para reprogramação será estabelecida e como esse protocolos encaram os desafios anteriormente apresentados. Finalmente, será apresentado de forma mais detalhada o protocolo estado da arte na literatura, Deluge, o qual é utilizado na comparação com o protocolo proposto nesse trabalho.

2.1 Desafios Para a Reprogramação em Redes de Sensores Sem Fio

Em consequência das características das RSSFs e da comunicação sem fio, o desenvolvimento de protocolos para reprogramação neste tipo de rede deve considerar diversos desafios, dentre os quais pode-se destacar:

- Todos os nós na rede devem receber o código do programa e de forma completa. Tal abordagem é diferente das aplicações tradicionais de RSSFs em que perdas ocasionais de dados podem ser toleradas.
- A complexidade de tempo e espaço dos algoritmos de reprogramação devem se adequar às restrições dos nós sensores.
- A reprogramação da rede exige alta largura de banda. Para a grande maioria de aplicações em RSSFs, os dados gerados pelos sensores é pequeno e facilmente se adequam às baixas larguras de banda dos canais sem fio.

- Escalabilidade é crucial para RSSFs de larga escala. Para que um serviço de reprogramação seja aplicável, o mesmo deve ser escalável em relação ao número de nós e à variação da densidade.
- Eficiência de energia também é importante. Como nós sensores possuem restrições de energia, a quantidade de energia consumida durante a reprogramação pode afetar o tempo de vida da rede. Dentre os problemas que podem ocasionar o desperdício de energia, destacam-se as colisões, *overhead* de mensagens de controle e o tempo ocioso pela espera de novos pacotes.
- Um protocolo simples para a disseminação de código em RSSFs é o *flooding* clássico. Neste protocolo, uma estação base realiza um *broadcast* da versão mais recente do programa para os seus vizinhos. Ao receber a versão mais recente do programa, cada nó armazena e então realiza um novo *broadcast* para os seus vizinhos. No entanto, em redes de larga escala com densidade alta, o *flooding* clássico é particularmente custoso e resulta em sérios problemas de redundância, contenção e colisão. Esse problema é conhecido como *broadcast storm problem* [Ni et al., 1999] e é causado basicamente pela redundância de dados e de remetente.

Além dos desafios citados acima, existe ainda a questão dos terminais ocultos, presentes na comunicação sem fio. Se dois nós estão fora do raio de comunicação um do outro, ao transmitirem pacotes ao mesmo tempo, pode resultar na colisão de pacotes em qualquer nó localizado na área de interseção entre os remetentes. O problema de terminal oculto degrada substancialmente o desempenho dos protocolos de acesso ao meio.

2.2 Características dos Protocolos de Reprogramação

Vários protocolos para a reprogramação de RSSFs foram propostos na literatura, conforme apresentado na tabela 2.1. Tais protocolos podem ser classificados de acordo com várias características, dentre as quais destacam-se:

- **Single-hop x multihop:** Os primeiros protocolos para reprogramação, como o XNP [Crossbow Technology, 2003], disseminam código apenas para os nós que estão dentro do raio de comunicação da estação base. No entanto, para que a reprogramação seja mais efetiva, a mesma deve ser possível para nós que estão a

Nome	Codificação	MAC	Salto	Escopo	Hierarquia	Pipelining
XNP	Programa	CSMA	Único	Rede	Não	Não
Incremental	patch	CSMA	Único	Rede	Não	Não
Trickle	script	CSMA	Múltiplo	Rede	Não	Não
MOAP	Programa	CSMA	Múltiplo	Rede	Não	Não
Deluge	Programa	CSMA	Múltiplo	Rede	Não	Sim
MNP	Programa	CSMA	Múltiplo	Rede	Não	Sim
Sprinkler	Programa	TDMA	Múltiplo	Rede	Sim	Não
Firecracker	Programa	CSMA	Múltiplo	Rede	Sim	Não

Tabela 2.1. Protocolos de reprogramação para RSSFs

múltiplos saltos da estação base. Uma característica interessante dos protocolos de reprogramação *multihop* é que tais protocolos são epidêmicos por natureza e conseqüentemente devem considerar questões importantes como eficiência e escalabilidade.

- Codificação:** A maioria dos protocolos de reprogramação disseminam a imagem do programa através da rede. No entanto, o *overhead* pode ser muito alto em situações em que apenas pequenas mudanças ocorreram entre as versões mais antigas e mais nova do programa. Abordagens que utilizam atualizações incrementais realizam a comparação entre as versões antigas e as novas do programa e transmitem apenas o que foi modificado. O protocolo *Incremental Network Programming* [Jeong & Culler, 2004] utiliza um algoritmo que realiza essa comparação especialmente otimizado para nós sensores com fortes restrições de recursos. *Trickle* [Levis et al., 2004] realiza a transmissão de *scripts* para a máquina virtual conhecida como *Maté*, e não código nativo compilado na linguagem nesC, comumente utilizada nas plataformas de sensores existentes. Apesar dos *scripts Maté* serem menores e mais simples de serem escritos que código nesC, esses *scripts* são limitados às funcionalidades da máquina virtual e não são tão flexíveis quanto a linguagem nesC. *Rateless Deluge* [Hagedorn et al., 2008] é uma modificação do protocolo *Deluge* [Hui & Culler, 2004], o qual faz uso de *Network Coding* [Shoup, 2005]. Nessa abordagem, o programa é codificado de tal maneira que uma quantidade maior de dados é enviada, no entanto, a perda de uma quantidade mínima de pacotes é tolerada.
- MAC:** Como *carrier sensing multiple access* (CSMA) MAC está presente no sistema operacional *TinyOS* [Crossbow Technology, 2003], a grande maioria dos protocolos de reprogramação utilizam CSMA. *Sprinkler* [Naik et al., 2005] utiliza *time-division multiple access* (TDMA) MAC na tentativa de reduzir contenção

e alcançar uma maior vazão. No entanto, TDMA exige um escalonamento de *slots* de tempo e sua implementação em um nó sensor é bem mais complexa que CSMA, tanto que TDMA não está disponível na maioria das plataformas de nós sensores.

- **Hierárquico:** Protocolos como o *Sprinkler* [Naik et al., 2005] e o *Firecracker* [Levis & Culler, 2004] estabelecem uma hierarquia na rede antes de iniciar o processo de reprogramação. Inicialmente a versão mais recente do programa é enviada para os super nós na hierarquia, onde esses nós são responsáveis por reprogramar os demais nós em sua área. Super nós podem ser *cluster-heads* ou um conjunto dominante de nós conectados capazes de cobrir toda a rede. Protocolos como *Deluge* [Hui & Culler, 2004] e MNP [Kulkarni & Wang, 2009, Wang, 2004] inicializam o processo de reprogramação a partir de uma estação base e não assumem nenhuma hierarquia.
- **Escopo:** Os protocolos de reprogramação existentes não possuem seleção de escopo e são capazes de disseminar novos programas apenas para todos os nós na rede, ou seja, nós individuais não podem ser reprogramados. Tal limitação é contornada no protocolo proposto nesse trabalho.
- **Pipelining:** Protocolos como o *Deluge* e MNP utilizam *pipelining* de forma a acelerar o processo de reprogramação, enquanto que protocolos como MOAP [Stathopoulos et al., 2003] utilizam o conceito de janela deslizante.

2.2.1 Pipelining

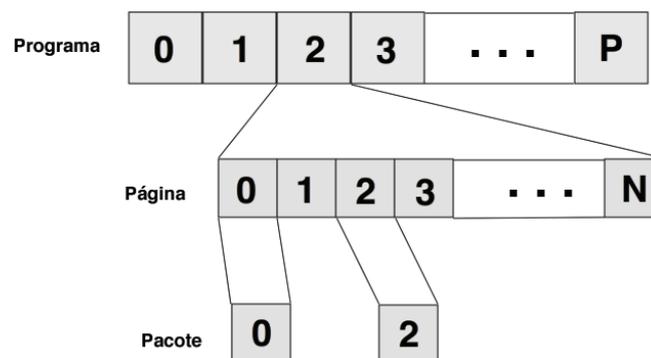


Figura 2.1. Divisão do programa em páginas e pacotes

Vários protocolos de reprogramação fazem uso de *pipelining* de forma a permitir a transferência paralela de dados na rede, possibilitando assim a multiplexação espa-

cial. *Pipelining* é realizado por meio de segmentação, ou seja, o programa é dividido em vários segmentos (chamados de páginas no Deluge), onde cada segmento contém um conjunto fixo de pacotes, conforme pode ser visto na figura 2.1. Os pacotes são as unidades que serão enviadas pela rede. Na ausência de *pipelining*, um nó necessita receber o programa completamente antes de iniciar a disseminação para os seus vizinhos. Por outro lado, ao se utilizar *pipelining*, um nó está apto a disseminar após receber o primeiro segmento completo. Para a transmissão de uma grande quantidade de dados, a utilização de *pipelining* pode aumentar a vazão total da rede de forma significativa. Além disso, ao não se utilizar segmentação, o nó necessita de um grande conjunto de estados para identificar as informações dos pacotes (recebido ou não), para grandes programas.

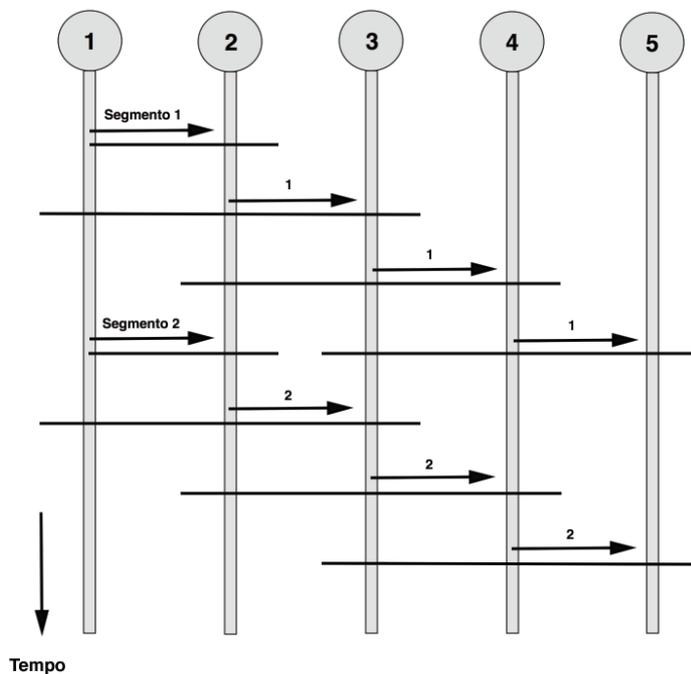


Figura 2.2. Pipelining de dois segmentos em uma rede de quatro saltos.

Várias questões precisam ser consideradas para que todos os benefícios do *pipelining* sejam alcançados. A primeira delas é garantir que a transferência paralela de dados não interfira uma na outra. A figura 2.2, representando uma rede simples, onde as linhas na horizontal representam o raio de interferência e as setas representam a comunicação confiável. Em consequência do problema de terminal oculto, a transferência simultânea de dados do nó 1 para o nó 2 irá colidir com a transferência de dados do nó 3 para o nó 4. A transferência paralela de dados deveria ocorrer com ao menos um

espaçamento de três saltos. Em uma rede com n saltos, sem *pipelining*, a transferência de x segmentos necessita de um tempo total de $O(nx)$. Com *pipelining*, o tempo total é reduzido para $O(n + 3(m - 1))$. Uma outra questão é que a implementação de *pipelining* pode necessitar que todos os nós na rede permaneçam com os rádios ligados, como ocorre com o protocolo Deluge. Para contornar tal problema, o protocolo *MNP* adota um esquema de dormência adaptativo. Esse protocolo reduz o tempo com que o nó fica inativo, colocando-o para dormir quando seus vizinhos estão transmitindo pacotes que não são de interesse para o nó. O tempo de dormência é cuidadosamente estabelecido de acordo com o tempo necessário para a transmissão de dados.

2.3 Deluge

Deluge é o protocolo estado da arte para reprogramação de RSSFs existente na literatura. Tal protocolo opera de acordo com uma máquina de estados em que cada nó obedece um conjunto de regras específicas de forma a atingir um estado global desejável. Na sua forma mais simples, ocasionalmente, cada nó realiza um anúncio da versão mais recente do programa para todos os nós que estão dentro do seu raio de comunicação. Se o nó S receber um anúncio de uma versão mais antiga de outro nó, R , S responderá com um anúncio da sua versão mais recente do programa. Por meio do anúncio, R determina quais as porções do programa precisam ser atualizadas e realiza uma requisição de tais porções para qualquer nó vizinho que tenha anunciado a disponibilidade de tais porções, inclusive S . Nós que recebem requisições, respondem com os dados requisitados. Além disso, *Deluge* utiliza *pipelining* de forma a permitir a transferência paralela de dados e possibilitar a multiplexação espacial dos dados. A seguir, é apresentado detalhadamente o funcionamento do protocolo *Deluge*.

2.3.1 Funcionamento do Protocolo Deluge

Um nó no protocolo *Deluge* pode operar em um de três estados possíveis em um dado momento no tempo: *MAINTAIN*, *RX* ou *TX*. O conjunto de regras que um dado nó deve seguir é uma função do seu estado atual e especifica quais as ações e transições de estados devem ser realizadas em resposta a eventos. O conjunto de regras locais e uma discussão de como cada uma dessas regras contribui para o comportamento global do protocolo é apresentada na sequência.

2.3.1.1 Maintenance

Um nó que está no estado *MAINTAIN* é responsável por assegurar que todos os nós dentro do seu raio de comunicação possuem a versão mais recente do programa. Para assegurar tal propriedade, cada nó ocasionalmente anuncia um sumário com as informações da versão mais recente do programa que ele possui e o conjunto de páginas que estão disponíveis para transmissão.

Idealmente, a frequência com que um nó realiza anúncios dentro de sua vizinhança deve ser independente da densidade, permitir a rápida propagação do programa e consumir poucos recursos enquanto estiver nesse estado. Essa propriedade deve ser válida inclusive quando milhares de nós estão dentro do raio de comunicação, cenário que pode ser representado pela programação dos nós antes dos mesmos serem depositados no campo de sensoriamento. De forma a controlar a transmissão de mensagens redundantes, Deluge divide o tempo em uma série de *rounds*, em que cada nó deve determinar se transmite ou não um anúncio nesses *rounds*. Com esse serviço de anúncios, um nó pode determinar se algum de seus vizinhos possui uma versão antiga do programa. Caso afirmativo, o nó deve realizar um anúncio da versão mais recente.

Uma contribuição significativa do protocolo Deluge é a sua ênfase na multiplexação espacial. Deluge realiza anúncios com a disponibilidade de novas páginas antes mesmo que todas as páginas do programa estejam completas, permitindo a disseminação das novas páginas recentemente recebidas, aumentando dessa forma a vazão na rede. Uma desvantagem é que todos os nós da rede devem permanecer com os rádios ligados de forma a alcançar todos os benefícios da multiplexação espacial.

2.3.1.2 Request

Um nó no estado *RX* é responsável por requisitar os pacotes restantes para completar uma determinada página. Deluge utiliza períodos de *backoff* aleatórios para minimizar o número de colisões entre requisições de outros nós, além de permitir a supressão de mensagens caso um vizinho realize requisição das mesmas páginas que o nó está requisitando.

2.3.1.3 Transmit

Um nó que está no estado *TX* é responsável por realizar um *broadcast* de todos os pacotes requisitados em uma dada página, até que todos os pacotes tenham sido transmitidos, momento no qual o nó retorna ao estado *MAINTAIN*. Deluge realiza a união de

novas requisições com requisições antigas que não foram atendidas, diminuindo assim o número de transmissões de pacotes.

2.4 Comentários Finais

Neste capítulo foram apresentados os principais desafios existentes na reprogramação de RSSFs. Em seguida, foi apresentada uma classificação dos principais protocolos existentes na literatura de acordo com as características desejáveis para um protocolo de reprogramação. Finalmente, foi apresentada uma descrição do protocolo Deluge, com destaque para o seu funcionamento. Por apresentar o melhor desempenho dentre os protocolos existentes, tal protocolo é destacado pela literatura como o estado da arte na reprogramação de RSSFs, fato que levou a utilização do protocolo Deluge na comparação com o protocolo proposto nesse trabalho.

Capítulo 3

Solução Proposta

Este capítulo apresenta o protocolo *Over-the-Air Programming based on Small World* (OAP-SW) [Maia et al., 2009] para RSSFs. A descrição desse protocolo se dará em três etapas: infra-estrutura, reprogramação e *multicast*. A etapa de infra-estrutura é responsável por agregar as características de redes *small world* à RSSF. Na etapa seguinte, reprogramação, serão especificadas as mensagens que possibilitam a mobilidade de código propriamente dita. Finalmente, na etapa de *multicast*, será apresentado o processo de reprogramação parcial da RSSF, característica essa que não está presente nos protocolos de reprogramação existentes.

3.1 Introdução

Como apresentado na seção 1.2, o problema aqui estudado é o de reprogramar os programas em execução presentes nos nós sensores de forma remota. Assumindo-se que a rede é conexa, todos os nós sensores devem receber a imagem exata do programa, além de serem atualizados com a versão correta do programa. Neste trabalho, apenas RSSFs com nós estacionários serão consideradas, deixando as redes com nós móveis como um trabalho futuro. O projeto de protocolos de reprogramação para RSSFs deve levar em consideração os seguintes requisitos:

- **Tempo de reprogramação:** O programa deve ser propagado e instalado o mais rápido possível, já que a interrupção da funcionalidade da rede deve ser a mínima possível.
- **Eficiência de energia:** RSSFs são fortemente limitadas em relação a sua capacidade de energia, portanto o consumo de energia decorrente do processo de

reprogramação deve ser o mínimo possível de forma a não comprometer o tempo de vida da rede.

- **Consumo de memória:** RSSFs são fortemente limitadas em relação a quantidade de memória disponível para as aplicações presentes na RSSF.
- **Confiabilidade:** A imagem exata do programa deve ser recebida pelos nós sensores e eventualmente, todos os nós da RSSF são reprogramados.
- **Autonomia:** O processo de reprogramação deve ser realizado sem intervenção humana.

Apesar do tempo de reprogramação e eficiência de energia não serem requisitos essenciais para o funcionamento correto dos protocolos de reprogramação, eles são importantes e não podem ser desprezados. No entanto, em consequência das dificuldades relacionadas ao desenvolvimentos de novos protocolos para RSSFs, nem sempre é possível atender todos os requisitos necessários, portanto escolhas devem ser feitas de forma a garantir a viabilidade do projeto da RSSF.

3.2 Estabelecimento da Infra-estrutura

Enquanto a comunicação de dados nas redes tradicionais se dá entre entidades de comunicação arbitrárias, em RSSFs a comunicação de dados tende a ocorrer de forma que o nó sorvedouro é o destino ou a origem de uma mensagem. Normalmente, RSSFs são compostas por um conjunto homogêneo de nós sensores, no entanto Gupta et. al [Gupta & Kumar, 2000] e Yarvis et. al [Yarvis et al., 2005] mostraram que redes *ad-hoc* homogêneas sofrem problemas severos de desempenho e funcionalidade. Em consequência disso, RSSFs heterogêneas estão ganhando espaço na comunidade científica. No trabalho aqui apresentado, assume-se a presença de uma grande quantidade de nós de *hardware* limitado (*L-sensors*) e uma pequena quantidade de nós com *hardware* diferenciado (*H-sensors*). O maior poder de comunicação dos nós *H-sensors* resulta em uma estrutura topológica de rede não trivial.

Conforme apresentado em [Kleinberg, 2008, Newman, 2003, Watts, 2007], modelos de redes complexas são importantes na caracterização de redes com características topológicas não triviais. Um desses modelos, conhecido como *small world*, apresenta características desejáveis para as RSSFs, como por exemplo, caminho mínimo médio pequeno. A criação de redes com características das redes *small world* se dá a partir da adição de um pequeno número de ligações de longo alcance, chamados de atalhos.

Como a comunicação em RSSFs se dá do nó sorvedouro para os demais nós da rede e vice-versa, os atalhos a serem adicionados, no trabalho aqui apresentado, levam em consideração esse padrão de comunicação de forma a otimizar a comunicação de dados entre o nó sorvedouro e os demais nós da rede. A seguir, é apresentado um modelo no qual todos os atalhos são gerados utilizando a angulação com relação ao nó sorvedouro. Este modelo é utilizado na fase de infra-estrutura do protocolo *OAP-SW*.

RSSFs são tipicamente grafos regulares. Portanto, é possível diminuir significativamente o caminho mínimo médio entre par de nós pela adição de atalhos, ou seja, pela introdução de um conjunto de nós com *hardware* diferenciado, os *H-sensors*. No entanto, em RSSFs a comunicação se dá por *broadcasting*, ou seja, todos os vizinhos que estão dentro do raio de comunicação de um dado nó, recebem os pacotes transmitidos pelo mesmo. Como o modelo de Watts e Strogatz [Watts & Strogatz, 1998] consiste da adição de ligações entre pares de nós, para que seja possível a introdução da propriedade de redes *small world* às RSSFs, é necessária a utilização de ligações *unicast* para a criação dos atalhos. Os nós finais desses atalhos devem operar com duas frequências de comunicação diferentes, uma para a comunicação entre *H-sensors* e outra para a comunicação entre *L-sensors*. Isso é feito para que a comunicação de longa distância entre os nós *H-sensors* não cause interferência na comunicação entre os nós *L-sensors*.

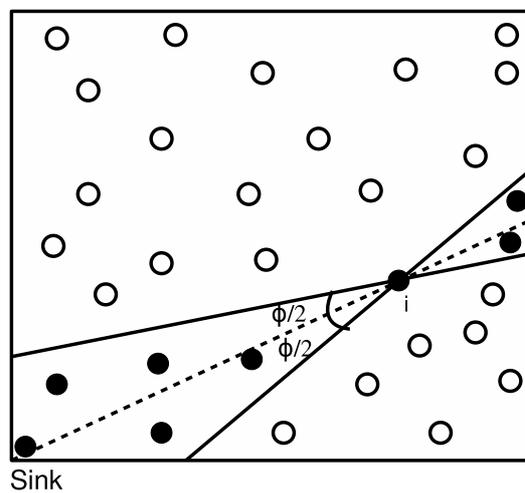


Figura 3.1. Angulação em direção ao nó sorvedouro

No modelo *Directed Angulation Toward the Sink Node* [Guidoni et al., 2008b, Guidoni et al., 2008a], cada nó escolhe aleatoriamente um outro nó para a criação de um atalho entre eles. Esse atalho é criado de acordo com uma probabilidade p e seus

pontos finais são nós *H-sensors*. Os atalhos são criados considerando-se a reta

$$y = m x + b$$

a qual passa na posição geográfica de cada nó e na posição geográfica do nó sorvedouro. Essa reta é a bissetriz do ângulo ϕ , o qual define uma região angular direcionada ao nó sorvedouro e oposto ao mesmo (ver figura 3.1). Os nós que estão na direção do fluxo de dados entre os nós i e o nó sorvedouro, podem ser escolhidos para a criação do atalho. O espaço de busca inclui a direção oposta ao nó sorvedouro já que tal direção também auxiliará no processo de comunicação de dados.

Para verificar se um outro nó sensor está dentro da região angular, um nó sensor realiza os seguintes passos. Primeiramente, o nó i calcula a equação da reta

$$y_1 = m_1 x + b_1$$

a qual passa por sua posição geográfica e a posição geográfica do nó sorvedouro. Em seguida, o nó i calcula a equação da reta

$$y_2 = m_2 x + b_2$$

que passa em sua posição geográfica e a posição geográfica do nó k , onde $k \neq i$ e o nó k é o nó que se quer verificar se o mesmo está dentro da angulação direcionado ao nó i . A tangente do ângulo formado pelas duas retas é dada por

$$\tan \theta = \frac{m_1 - m_2}{1 + m_1 m_2}.$$

Portanto, o nó i pode determinar se um dado nó está dentro da região angular do ângulo ϕ , fazendo-se

$$\tan \theta < \tan \frac{\phi}{2}.$$

Utilizando-se desses atalhos, o protocolo *OAP-SW* cria a infra-estrutura *small world* como segue. Considere um grafo regular $G(V, E)$. Dado uma aresta $e \in E \mid e = (v_i, v_j)$, onde $v_i, v_j \in V$, é necessário adicionar uma aresta e' entre os nós v_i e $v_k \in V$ de forma que $e' = (v_i, v_k)$, onde

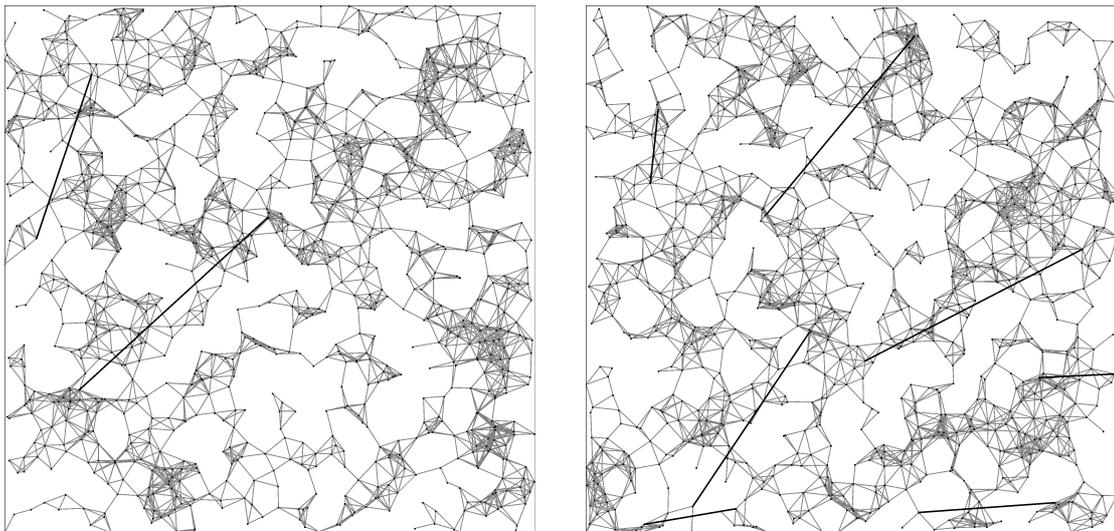
$$\tan \theta = \frac{m_i - m_k}{1 + m_i m_k},$$

portanto

$$\frac{m_i - m_k}{1 + m_i m_k} < \tan \frac{\phi}{2},$$

com $v_i, v_k \in V$, e m_i, m_k os coeficientes angulares das retas que passam pelos nós i e o nó sorvedouro, e pelos nós i e o nó k . A adição dessa aresta é realizada utilizando-se da mesma probabilidade p determinada para cada aresta de G . Esses passos são repetidos para todas as arestas de G .

A figura 3.2 ilustra a criação de atalhos no modelo pela adição de atalhos de acordo com uma dada probabilidade p , considerando apenas a comunicação unicast. Nesta figura, o nó sorvedouro está localizado no canto inferior esquerdo da rede. Vale ressaltar que para todos os valores de probabilidade, todos os atalhos são criados em direção ao nó sorvedouro. Quando a probabilidade de adicionar um atalho é igual a 0, a RSSF é um grafo regular. A partir do momento em que p começa a aumentar, a RSSF começa a exibir as características de redes *small world*.



(a) $p = 0.0001$

(b) $p = 0.001$

Figura 3.2. Criação de atalhos considerando-se diferentes valores de p

3.3 Protocolo de Reprogramação

O protocolo *OAP-SW* é um protocolo baseado em mensagens *NACK* e que faz uso de anúncios periódicos para manter seus vizinhos informados sobre o estado atual. O ciclo básico de comunicação é apresentado na figura 3.3 e detalhado a seguir:

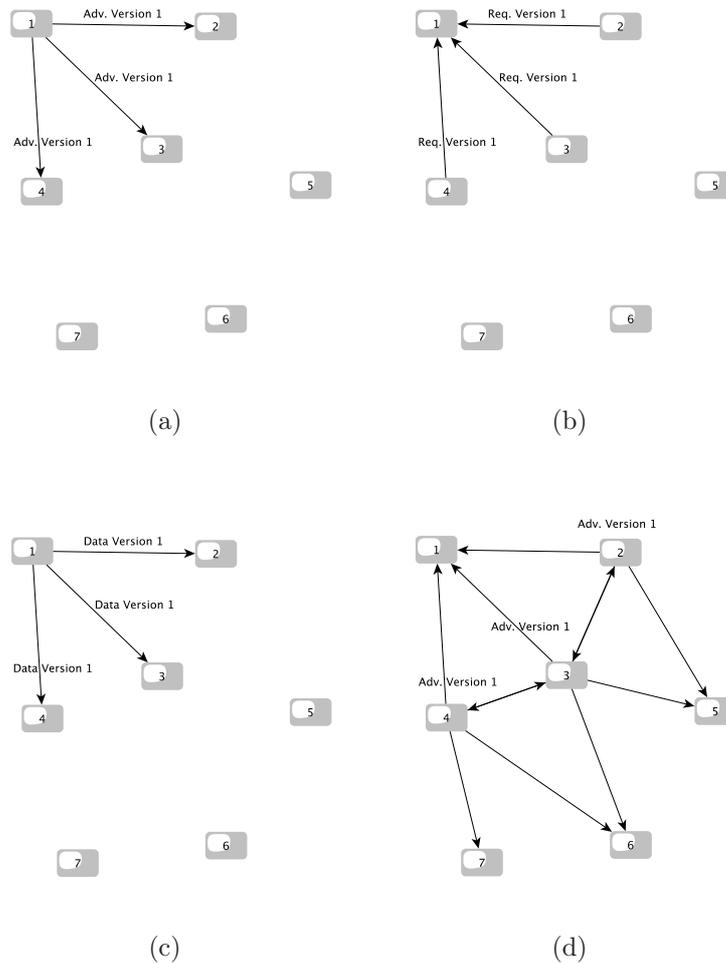


Figura 3.3. Troca de mensagens do protocolo de reprogramação

Passo 1 Todos os nós na rede periodicamente realizam um *broadcast* com um anúncio (figura 3.3). Esse anúncio contém um vetor que descreve as páginas que o anunciante já recebeu por completo e que estão disponíveis para serem repassadas. Os anúncios podem ser realizados com uma frequência maior no momento em que a reprogramação está sendo realizada.

Passo 2 Ao receber um anúncio, o nó verifica qual é o valor da menor página que ele necessita e realiza uma requisição para o primeiro nó a anunciar aquela página. Essa requisição indica qual a página e os pacotes necessários (ver figura 3.3).

Passo 3 O nó anunciante inicia um período de espera. Quando esse período termina, o anunciante realiza um *broadcast* de cada pacote requisitado da página selecionada (figura 3.3).

Passo 4 Quando um nó recebe o último pacote de que ele precisa para completar uma página, ele realiza um anúncio antes de requisitar novos pacotes de que ele necessita, possibilitando assim multiplexação espacial (figura 3.3).

É possível perceber que o protocolo evita manter qualquer informação sobre o estado de seus vizinhos ou de qualquer outro nó da rede, resultando em uma implementação simples e robusta. Outra característica do protocolo *OAP-SW* é que ele possibilita *pipelining* ao dividir o programa em um tamanho fixo de páginas, as quais são divididas em pacotes. Tal abordagem resulta em uma característica desejável para os protocolos de reprogramação, que é a multiplexação espacial, ou seja, permite que partes diferentes da rede realize o processo de reprogramação ao mesmo tempo. No entanto, esse protocolo toma proveito das características *small world*, apresentada anteriormente, para tornar a mobilidade de código mais eficiente.

Para entender o comportamento do protocolo *OAP-SW*, considere a figura 3.4. Nos protocolos tradicionais de reprogramação, para uma rede com quatro nós (figura 3.3), se o nó 1 possuir uma nova versão do programa para disseminar para a rede, então o mesmo iria propagar todo o programa sem *pipelining* (figura 3.3). Apenas após receber o programa completo do nó 1, é que o nó 2 iria iniciar a disseminação do programa para os seus vizinhos (figura 3.3). O mesmo acontece com o nó 3, ele pode iniciar a disseminação do novo programa, somente após receber a nova versão completa do nó 2 (figura 3.3).

No protocolo *OAP-SW* o processo é mais eficiente já que um nó não precisa receber o programa completo para poder dar início a disseminação para os seus vizinhos, graças ao uso do *pipelining*, conforme apresentado na figura 3.5. Inicialmente, o nó 1 iria disseminar a primeira página do programa para o nó 2 (figura 3.3). Após receber parte do programa do nó 1, o nó 2 iria iniciar a disseminação da parte do programa que ele já possui, antes de requisitar pela página seguinte (figura 3.3), possibilitando assim a multiplexação espacial. No entanto, o uso de ligações de longo alcance torna o processo de multiplexação espacial ainda mais eficiente, conforme será observado na seção de Resultados de Simulação.

3.3.1 Funcionamento do Protocolo OAP-SW

Nesta seção será apresentado uma descrição detalhada do protocolo proposto. Apesar da descrição básica do protocolo apresentada anteriormente, *OAP-SW* considera várias questões sutis. Primeiramente, o protocolo é ciente de densidade, de tal forma que anúncios e requisições redundantes são suprimidas, diminuindo assim a contenção do

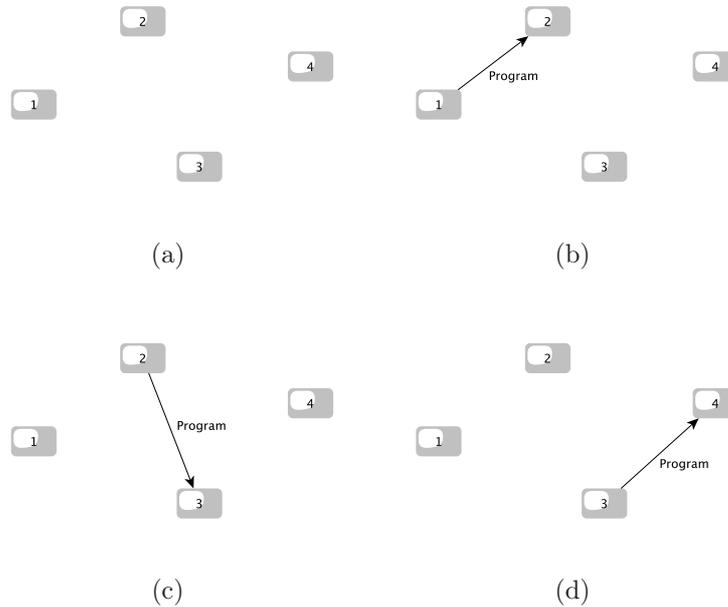


Figura 3.4. Reprogramação sem pipelining

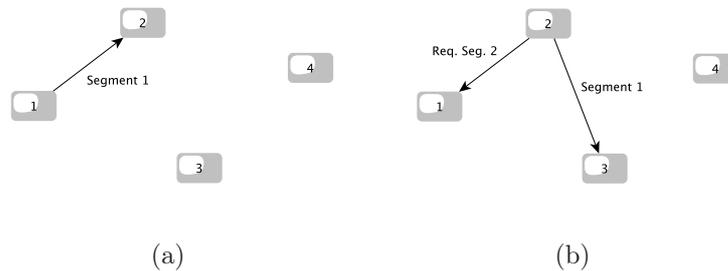


Figura 3.5. Reprogramação com uso de pipelining

meio. É importante destacar que enquanto supressão de mensagens pode melhorar o desempenho do protocolo, os períodos de *back off* introduzidos podem aumentar a latência. Segundo, é que protocolos para RSSFs devem ser robustos às ligações assimétricas. A comunicação de três vias empregada pelo protocolo OAP-SW ajuda a garantir que uma conexão bi-direcional existe antes da transmissão de dados. Além disso, se um nó não receber todos os dados após um número fixo de requisições, ele escolherá um outro vizinho capaz de atender suas requisições. Finalmente, o protocolo OAP-SW enfatiza o uso de multiplexação espacial de forma a permitir a transferência paralela de dados.

Assim como o protocolo Deluge, um nó no protocolo OAP-SW pode estar em um de três estados possíveis, *IDLE*, *REQUESTING* e *TRANSMITTING*:

3.3.1.1 IDLE

Um nó que se encontra em estado *IDLE* deve garantir que todos os nós em sua vizinhança possuem a versão mais recente do programa. Para garantir tal propriedade, cada nó da rede realiza um anúncio periodicamente. Esse anúncio contém a versão mais recente do programa que o nó possui e o conjunto de páginas que o nó tem disponível para transmitir aos seus vizinhos. Define-se a página i como uma página **completa** se todos os pacotes para essa página foram recebidos corretamente. Uma página i está **disponível** somente se ela estiver completa e todas as páginas no intervalo $[0, i)$ também estiverem completas. Dessa forma, o anúncio contém apenas dois valores, v e x . v é o número da versão e x é o número da maior página disponível para transferência. Um nó ao receber um anúncio com $v' = v$ e $x' > x$ muda para o estado *REQUESTING*, a não ser que uma requisição para uma página $p \leq x$ foi previamente recebida. Ao receber uma requisição de uma página $p \leq x$ com versão v , muda para o estado *TRANSMITTING*.

3.3.1.2 REQUESTING

Um nó que está no estado *REQUESTING* é responsável por requisitar os pacotes restantes para completar a página $p = x + 1$. Cada requisição é composta por um vetor de *bits* que identifica os pacotes da página que são necessários. Como o meio de comunicação é sem fio, respostas para as requisições podem ser compartilhadas pelos receptores. Após realizar uma requisição, o nó aguarda por uma resposta e então realiza novas requisições para os dados que foram perdidos. OAP-SW permite a comunicação mesmo na presença de ligações assimétricas. Um nó R que está recebendo anúncios de um nó S pode não conseguir realizar requisições diretamente para S . Dessa forma, o número de requisições que um nó R realiza a um nó S é limitada em um valor máximo r .

3.3.1.3 TRANSMITTING

Um nó no estado *TRANSMITTING* é responsável por disseminar todos os pacotes de uma dada página que foram requisitados, até que todos os pacotes os pacotes tenham sido disseminados, momento no qual o nó retorna ao estado *IDLE*. O nó S ao receber uma requisição de um conjunto de pacotes P_1 de um vizinho R_1 não inicia imediatamente a transmissão dos pacotes requisitados. O nó S realiza um período de espera, possibilitando assim que novas requisições possam ser realizadas por outros vizinhos, por exemplo, o vizinho R_2 requisitando os pacotes P_2 . Com isso, o nó S responderá às

requisições com mensagens de resposta compostas pelos pacotes $P_1 \cup P_2$. Ao final da transmissão, retorna ao estado *IDLE*.

3.4 Reprogramação com Multicast

3.4.1 Motivação

Os protocolos de reprogramação para RSSFs exigem que todos os nós da RSSF devam receber a nova versão do programa que está sendo disseminada. No entanto, em algumas situações pode ser desejável reprogramar apenas alguns dos nós sensores presentes na rede. Para ilustrar tal situação, imagine o seguinte cenário. Suponhamos a existência de uma RSSF na floresta Amazônica, a qual é composta por nós com capacidade de monitorar grandezas como temperatura, umidade e pressão. No entanto, suponhamos que para regiões diferentes da floresta, diferentes grandezas estarão sendo monitoradas, ou seja, em uma região *A* da floresta os nós ali presentes estarão executando uma aplicação responsável em monitorar a temperatura e para uma região *B* da floresta, os nós estarão executando uma aplicação responsável em monitorar a umidade e assim por diante. Imaginemos agora que uma nova versão da aplicação responsável em monitorar a temperatura do ambiente foi disponibilizada e apenas os nós sensores que estão executando tal aplicação precisam ser reprogramados. Neste cenário, ao se utilizar os protocolos existentes, a reprogramação não será possível já que todos os nós da rede deverão ser reprogramados.

Diante de tal limitação, neste trabalho foi proposta uma extensão do OAP-SW com *multicast* de forma a viabilizar a reprogramação de um subconjunto dos nós da rede. No problema aqui trabalhado, o que se quer é reprogramar os nós pertencentes a uma determinada região, portanto uma solução de *multicast* geográfico [Basagni et al., 2001, Gui & Mohapatra, 2004, Lee et al., 2002, Sanchez et al., 2006, Sanchez et al., 2007, Wu & Candan, 2006] foi proposta. Assim como nos trabalhos existentes em *multicast* geográfico, assume-se que o nó que está iniciando a reprogramação, nó sorvedouro no nosso caso, conhece a localização de todos os nós que irão receber a nova versão do programa a ser disseminado na rede por intermédio de algum mecanismo de gerenciamento de grupos *multicast*. Assume-se também que todos os nós conhecem a sua própria localização e a de seus vizinhos.

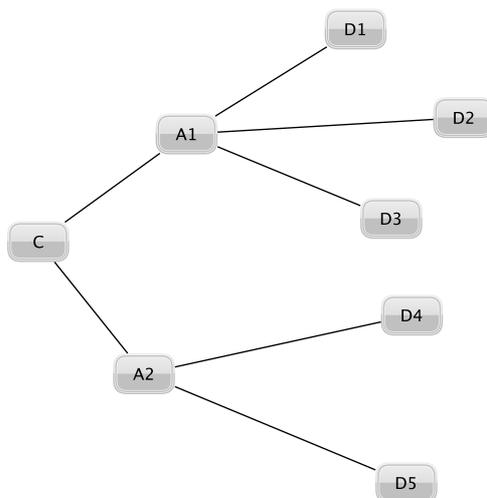


Figura 3.6. Determinação dos vizinhos para repassar a mensagem *multicast*.

3.4.2 Algoritmo Multicast

Várias heurísticas foram propostas na literatura para a obtenção de uma árvore *multicast*, problema que já foi demonstrado como sendo NP-completo [Ruiz & Gomez-Skarmeta, 2005]. No entanto, optou-se na utilização de uma heurística simples, de forma apenas a contornar as limitações impostas pelos protocolos de reprogramação existentes na literatura. Dados k nós destino, o algoritmo para a seleção dos vizinhos realiza os seguintes passos: inicia-se com o conjunto de vizinhos $\{D_1, D_2, \dots, D_k\}$ para os quais existe um vizinho do nó corrente que provê avanço. A heurística consiste simplesmente em agrupar, em um mesmo subconjunto, os nós destino para os quais o vizinho que provê o maior avanço é o mesmo. Para o melhor entendimento da heurística, considere a figura 3.6. Neste exemplo temos que o nó C recebeu uma mensagem *multicast* e precisa determinar quais de seus vizinhos ficarão responsáveis por quais nós destino. Para esse caso teríamos $\{\{D_1, D_2, D_3\}, \{D_4, D_5\}\}$, onde A_1 ficaria responsável pelos nós destino D_1, D_2, D_3 , e o nó A_2 ficaria responsável pelos nós D_4 e D_5 . Quando um nó C não possuir nenhum vizinho que provê avanço em direção a algum dos nós destino, o algoritmo entra na fase de perímetro e executa o algoritmo *Greedy-face-Greedy* proposto em [Bose et al., 2001].

Quando o dado *multicast* está sendo repassado, apenas os vizinhos selecionados pelo vizinho atual necessitam processar a mensagem. Adicionou-se um cabeçalho às mensagens de reprogramação para permitir que os vizinhos percebam que eles foram selecionados como nós intermediários. Tal cabeçalho também é utilizado na identificação dos nós destino que necessitam ser roteados em modo perímetro. Esse cabeçalho

contém a posição do nó que está enviando a mensagem e uma lista de campos, onde cada campo está associado a cada vizinho selecionado como nó intermediário. Uma mensagem transmitida por um nó é recebida por todos os seus vizinhos em consequência da natureza da comunicação sem fio. Isso significa que para transmitir uma mensagem *multicast* para vários nós intermediários só é necessário transmitir apenas uma mensagem com todas as informações necessárias. Tal característica reduz o *overhead* do protocolo. Cada campo do cabeçalho contém as seguintes informações:

- **Identificação do vizinho:** Identificação única do nó selecionado como nó intermediário.
- **Informações de perímetro:** Esse campo é incluído apenas se existe algum nó destino que necessita ser roteado em modo perímetro. Contém informações sobre o nó no qual iniciou o modo perímetro e a lista de nós destino que estão sendo roteados nesse modo.
- **Lista de destinos:** Lista com as posições dos nós destino.

3.5 Comentários Finais

Este capítulo apresentou um protocolo para reprogramação de RSSFs chamado *Over-the-Air Programming based on Small World* (OAP-SW) que considera o fluxo de comunicação nestas redes para a adição de atalhos entre nós com *hardware* diferenciado, obtendo assim uma topologia de rede não trivial que pode ser caracterizada utilizando-se modelos de redes *small world*. Esse protocolo assume que a rede é heterogênea, ou seja, existe uma grande quantidade de nós sensores com *hardware* limitado e uma pequena quantidade de nós com *hardware* diferenciado, os quais possuem maior raio de comunicação, possibilitando a adição de ligações de longo alcance entre os mesmos. Além disso, foi proposta a extensão do protocolo de reprogramação para suportar *multicast*, de forma a possibilitar a reprogramação de um subconjunto dos nós da rede.

Capítulo 4

Resultados de Simulação

4.1 Introdução

Neste capítulo são apresentados os resultados de simulação do protocolo *OAP-SW*. As avaliações foram realizadas levando-se em conta as seguintes pressuposições:

- **Simulação:** Realizou-se avaliações por intermédio de simulações e o simulador utilizado foi o simulador de redes Sinalgo, versão 0.75.3 [Group, 2008]. Todos os resultados correspondem a média aritmética de n simulações, onde n é definido de acordo com o intervalo de confiança desejado. Seja $n = (100 z s / e w)^2$ onde s é o desvio padrão, w é a média da amostra inicial de 10 simulações e z é uma distribuição uniforme adequada para $n > 30$. Em todas as simulações, foi utilizado um intervalo de confiança de 95% ($e = 0.05$).
- **Parâmetros avaliados:** Foram variados o número de nós, probabilidade de perda de pacotes, densidade da rede e para o caso do *multicast*, o número de grupos *multicast*. Quando a densidade não foi variada, área do campo de sensoriamento foi ajustada de forma a manter a densidade na rede de aproximadamente 16 vizinhos. Foi considerado também que a densidade da rede é $n \pi r^2 / A$, onde A é a área sensoriada. Os principais parâmetros de simulação são apresentados na tabela 4.1.
- **Métricas avaliadas:** As métricas avaliadas foram o número de mensagens e o tempo de reprogramação da rede. O objetivo é avaliar o comportamento do protocolo *OAP-SW* em redes heterogêneas em cenários confiáveis e com perda de pacotes.

Tabela 4.1. Parâmetros de simulação.

Parâmetro	Valor
Nó sorvedouro	Inferior esquerdo
Tamanho da rede	(121, 256, 529, 1024, 2048)
Campo de sensoriamento	700 x 700 m ²
Densidade da rede	16 nós
Tamanho do programa	16 KB
Probabilidade de perda	0.0
Raio de comunicação dos L-sensor	50 m
Raio de comunicação dos H-sensor	500 m
Probabilidade de nós H-sensor	0.01

4.2 Variando o Tamanho da Rede

Neste cenário, foi variado o número de nós na rede (121, 256, 529, 1024, 2048) e os demais valores presentes na tabela 4.1 foram mantidos. Nas figuras apresentadas nesta seção, *Flooding* representa a implementação *naive* de um protocolo de reprogramação, *Deluge* é protocolo estado-da-arte da literatura e *OAP-SW* representa o protocolo proposto.

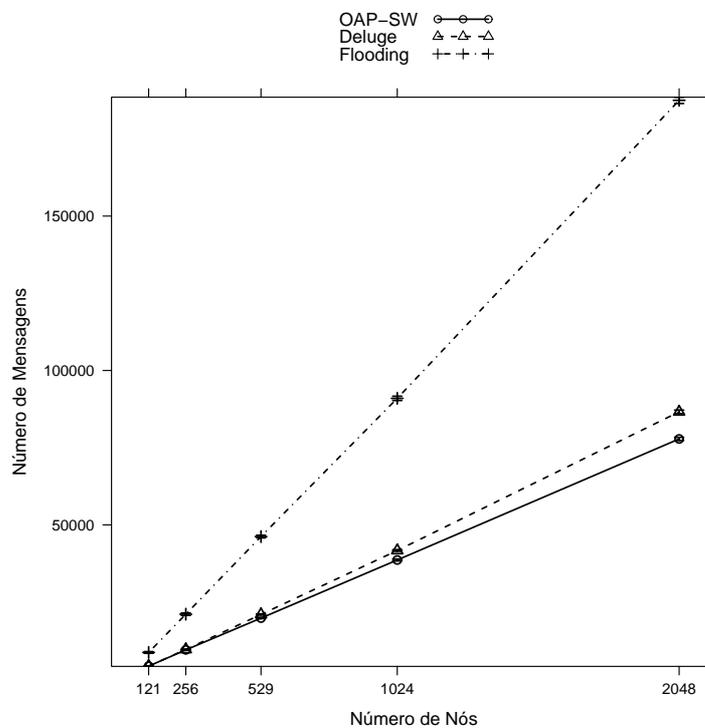


Figura 4.1. Número de mensagens transmitidas para um cenário confiável

A figura 4.1 apresenta o número de mensagens transmitidas durante o processo de reprogramação para um cenário confiável. Por confiável entende-se uma rede sem colisões, interferência e perda de pacotes. Conforme pode ser observado, para os cenários com 121, 256 e 529 nós, o número de mensagens transmitidas é praticamente o mesmo tanto para o *Deluge* quanto para o *OAP-SW*. No entanto, para os cenários com 1024 e 2048 nós a diferença no número de mensagens transmitidas torna-se mais evidente. Por exemplo, para 2048 nós, o protocolo *OAP-SW* transmite cerca de 10% menos mensagens que o protocolo *Deluge*. Esse resultado pode ser explicado pelas características de rede *small world* presente no protocolo *OAP-SW*, a qual reduz o caminho mínimo médio da rede e consequentemente o número de mensagens. Essa diferença é mais evidente para um número maior de nós, já que a probabilidade de nós *H-sensors* é proporcional ao número de nós na rede. Apesar do consumo de energia não ter sido avaliado, o número de mensagens transmitidas durante a reprogramação é uma forte estimativa, já que nas RSSFs a comunicação é a principal responsável pelo consumo de energia.

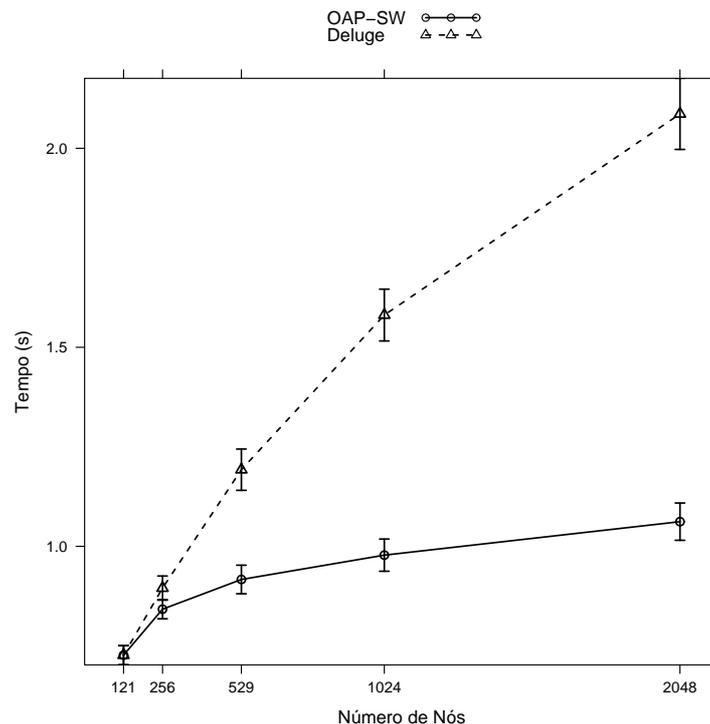


Figura 4.2. Tempo gasto na reprogramação da rede para um cenário confiável

Uma métrica importante dos protocolos de reprogramação para RSSFs é o tempo necessário para reconfigurar os nós sensores da rede, já que pode não ser interessante para algumas aplicações interromper suas operações por longos períodos de tempo

enquanto a reprogramação é realizada. Portanto, qualquer avanço que visa diminuir o tempo gasto na reprogramação é muito bem apreciada. A figura 4.2 mostra o tempo gasto pelos protocolos *Deluge* e *OAP-SW* para reconfigurar uma rede com diferentes quantidades de nós sensores. Para redes com 121 e 256 nós, o comportamento de ambos os protocolos é bastante semelhante. No entanto, para redes com 529, 1024 e 2048 nós a diferença de comportamento entre os protocolos é considerável. Por exemplo, para 529 nós o protocolo *OAP-SW* reduz o tempo de reprogramação da rede cerca de 23% quando comparado ao protocolo *Deluge*. Para 1024 nós a redução é de 38% e finalmente para 2048 nós a redução chega a quase 50%. A explicação para essa diferença de comportamento é simples. Se olharmos a figura 3.2, é possível perceber que quando um nó *H-sensor* envia seu programa para outro nó *H-sensor* situado longe do primeiro, partes diferentes da rede estarão realizando o processo de reprogramação, acelerando assim a disseminação do programa. Tal característica não é observada no protocolo *Deluge* já que o processo de reprogramação ocorre salto a salto, vizinho por vizinho.

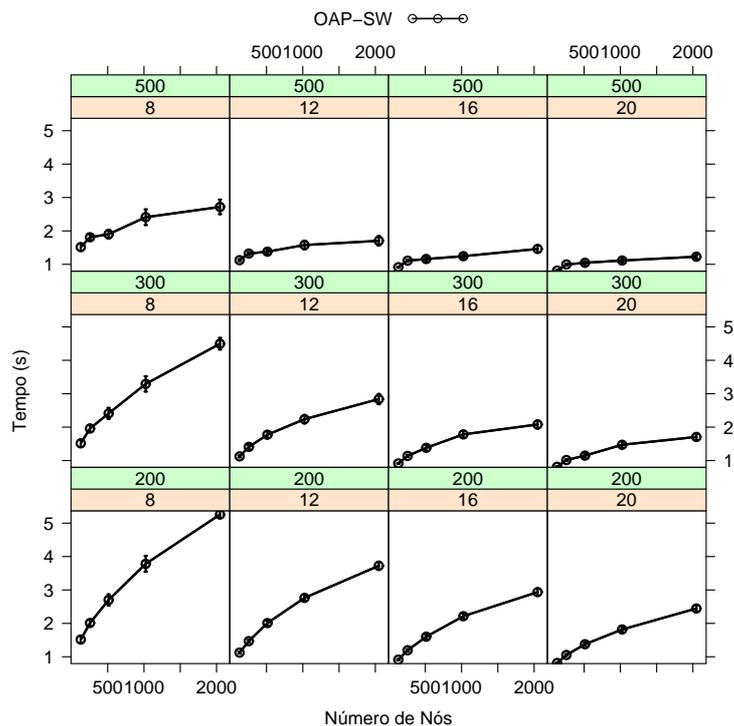


Figura 4.3. Tempo gasto na reprogramação da rede para diferentes valores de densidade

De forma a melhor avaliar o comportamento do protocolo proposto, foi avaliado um cenário em que tanto a densidade da rede (8, 12, 16 e 20) quanto o raio de co-

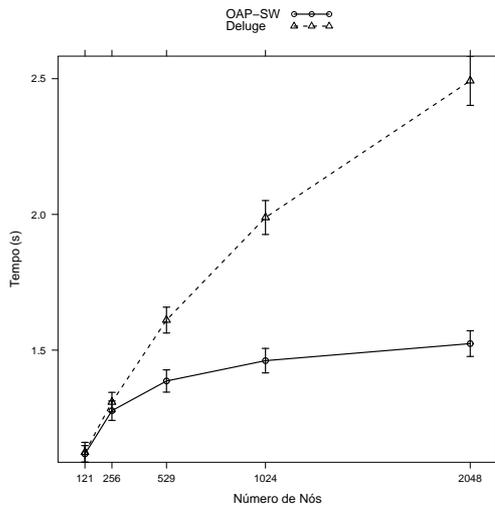
municação dos nós *H-sensors* (200, 300 e 500), conforme pode ser visto na figura 4.3. O aumento no valor da densidade resulta em uma redução no tempo gasto durante a reprogramação. Os mecanismos de supressão de mensagens em uma vizinhança utilizados pelo algoritmo ocasionam uma redução no número de mensagens trocadas pelo algoritmo e conseqüentemente uma redução no tempo de reprogramação da rede. Assim como a densidade, o aumento no raio de comunicação dos nós *H-sensors* também ocasiona uma redução no tempo de reprogramação da rede. Essa redução era esperada, já que ao se aumentar o alcance dos atalhos de longo alcance, partes mais distantes da rede podem executar o processo de reprogramação ao mesmo tempo, acelerando assim o processo global.

4.3 Variando Probabilidade de Perda de Pacotes

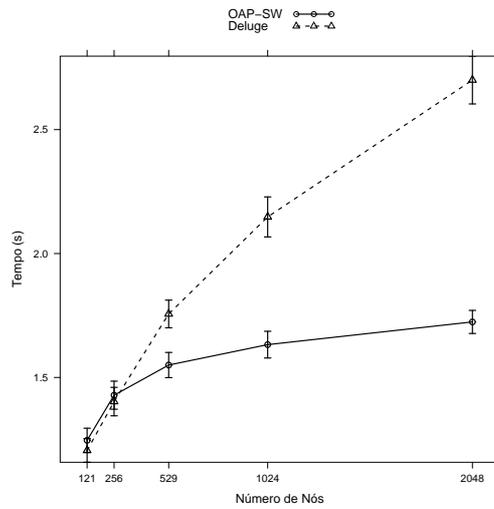
Nesta seção é avaliado o comportamento dos protocolos *Deluge* e *OAP-SW* para um cenário com perda de pacotes. A intenção é verificar o quão sensível os protocolos são em relação à perda de pacotes. No entanto, na seção 4.2 foi observado que apesar de existir uma diferença na quantidade de mensagens transmitidas por ambos os protocolos, essa diferença não foi tão significativa quanto o tempo necessário para reconfigurar a rede. Como consequência, aqui apresentaremos apenas os resultados do tempo gasto para reconfigurar a rede.

A figura 4.3 apresenta o tempo necessário para reconfigurar a rede com uma taxa de perda de pacotes igual a 10%. Assim como no cenário confiável, a partir de 529 nós, o protocolo *OAP-SW* começa a apresentar um tempo de reprogramação menor que o protocolo *Deluge*. Para 2048 nós, o protocolo *OAP-SW* reduz em cerca de 49% o tempo gasto na reprogramação da rede. As figuras 4.3, 4.3, e 4.3 apresentam o tempo de reprogramação para probabilidades de perda de pacotes iguais a 20%, 30% e 40%. Os resultados mostram que o tempo aumenta com o aumento da taxa de perda, no entanto o comportamento é semelhante ao cenário sem perda de pacotes, evidenciando a efetividade do protocolo.

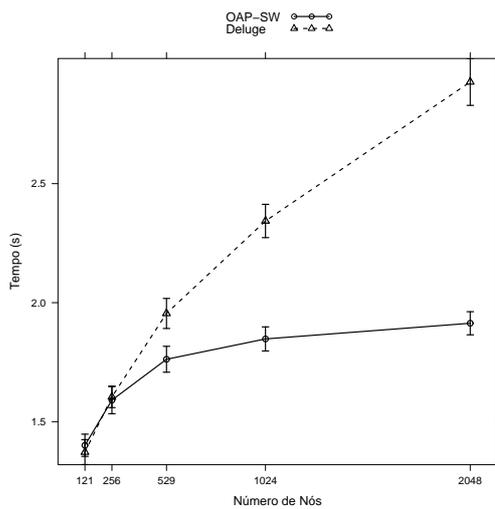
Vale ressaltar que não existem mecanismos para a detecção de perda de pacotes. A perda é contornada, pois os protocolos aqui apresentados utilizam comunicação *broadcast*, ou seja, caso alguns pacotes sejam perdidos ao longo do caminho o código a ser reprogramado chegará ao destino através de outros nós, só que, eventualmente, gastando um tempo maior.



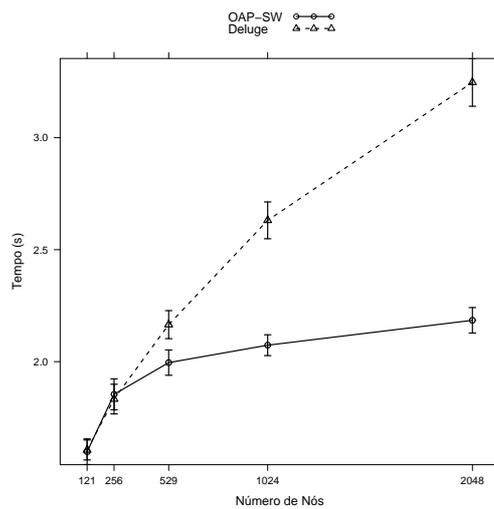
(a) Probabilidade de 0.1



(b) Probabilidade de 0.2



(c) Probabilidade de 0.3



(d) Probabilidade de 0.4

Figura 4.4. Tempo gasto na reprogramação da rede para um cenário com perda de pacotes

4.4 Avaliação da Reprogramação com Multicast

Nesta seção, são apresentados os resultados da avaliação do protocolo *OAP-SW* na reprogramação de um subconjunto de nós da rede. Como as soluções presentes na literatura não permitem a reprogramação de apenas parte da rede, nesta seção serão comparados os protocolos *OAP-SW-Multi* e *OAP-Multi*, ou seja, o protocolo de repro-

gramação *multicast* com a infra-estrutura *small world* e o protocolo de reprogramação *multicast* sem a infra-estrutura *small world*. A intenção da avaliação aqui realizada é a de demonstrar que a combinação da solução de reprogramação com *multicast* é funcional. Para as nossas avaliações, um novo parâmetro é utilizado, o de Grupos Multicast. A definição dos Grupos Multicast é realizada da seguinte forma: inicialmente, o campo de sensoriamento é dividido em quadrantes; os nós que estiverem dentro do mesmo quadrante, pertencerão ao mesmo grupo *multicast*.

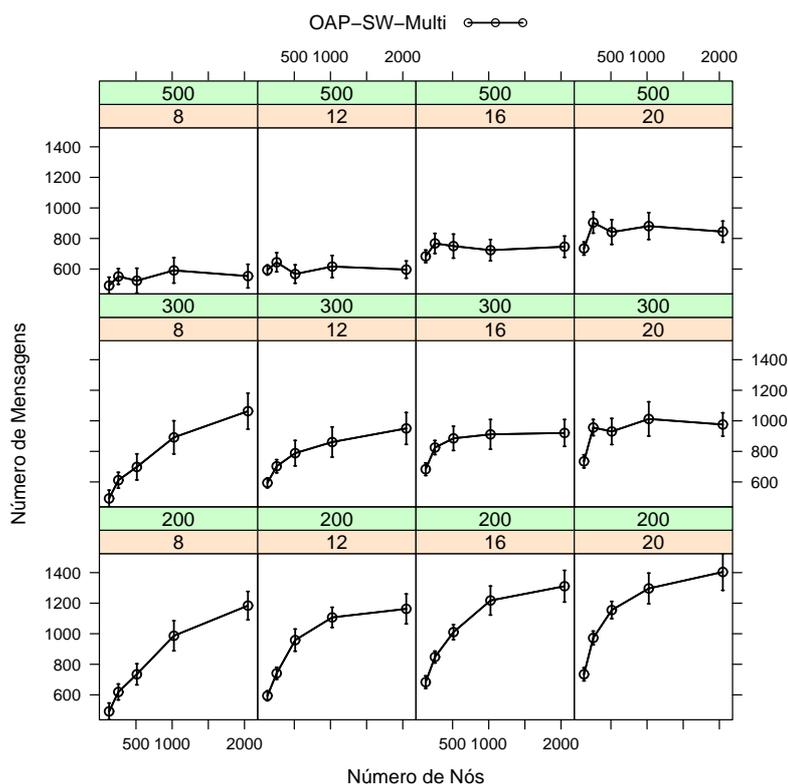


Figura 4.5. Número de mensagens trocadas para redes com diferentes densidades

A figura 4.5 apresenta o número de mensagens trocadas para reconfigurar os nós de um grupo *multicast* para diferentes valores da densidade de rede (4, 8, 12, 16, 20 e 24) e para diferentes valores do raio de comunicação dos *H-sensors* (200, 300 e 500). Nesta avaliação, assumiu-se a existência de apenas um grupo *multicast*, onde apenas os nós pertencentes a esse grupo tiveram seu *software* atualizado. Conforme pode ser observado, o aumento no valor da densidade resulta em um aumento no número de mensagens trocadas durante a reprogramação. Apesar do protocolo *OAP-SW-Multi* empregar mecanismos de supressão de mensagens em uma vizinhança, o aumento no número de mensagens deve-se ao fato de que o aumento da densidade ocasiona um aumento no número de nós pertencentes ao grupo *multicast*, ou seja,

uma maior quantidade de nós a serem reprogramados e conseqüentemente um maior número de mensagens trocadas durante o processo. Por outro lado, o efeito do aumento do raio de comunicação dos nós *H-sensors* é exatamente o contrário, ou seja, ocorre uma redução no número de mensagens trocadas durante o processo com o aumento do raio de comunicação. A explicação para tal redução é que neste cenário, o grupo *multicast* utilizado são os nós localizados no quadrante mais afastado do nó sorvedouro, quanto menor o raio de comunicação dos nós *H-sensors*, maior é a utilização de nós intermediários entre o nó sorvedouro e os nós do grupo *multicast* para conduzir as mensagens até os nós a serem reprogramados. Com o aumento no raio de comunicação dos nós *H-sensors*, a presença das ligações de longo alcance diminui a necessidade da utilização desses nós intermediários, já que tais ligações permitem alcançar os nós do grupo *multicast* com uma quantidade menor de saltos e conseqüentemente, um número menor de mensagens.

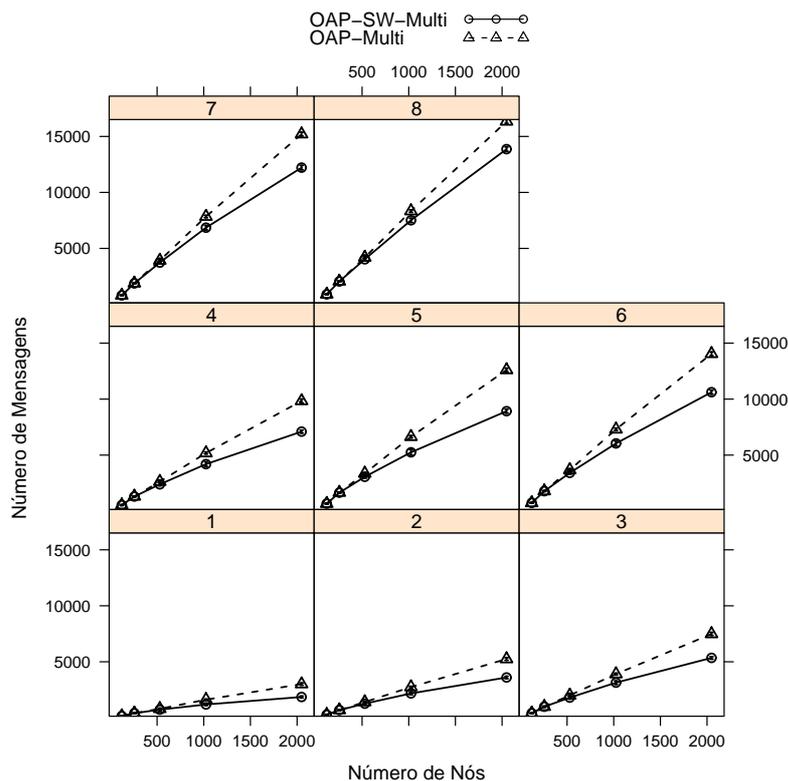


Figura 4.6. Número de mensagens trocadas para diferentes quantidades de grupos *multicast*

Avaliou-se também o comportamento de ambos os protocolos para diferentes quantidades de grupos *multicast*. A figura 4.6 apresenta o número de mensagens trocadas durante a reprogramação para diferentes quantidades de grupos *multicast* (1, 2, 3,

4, 5, 6, 7 e 8). Se observarmos a quantidade de mensagens trocadas para 2048 nós em todos os cenários avaliados, pode-se perceber que a quantidade de mensagens trocadas no protocolo *OAP-Multi* aumenta mais rapidamente com o aumento do número de grupos *multicast* do que o protocolo *OAP-SW-Multi*. Mais uma vez a explicação para tal comportamento está na presença das ligações de longo alcance presentes no protocolo *OAP-SW-Multi*, o qual reduz a quantidade de saltos para se alcançar os grupos *multicast* e consequentemente o número de mensagens trocadas. Apesar da utilização de *multicast* com um protocolo de reprogramação já representar um avanço em relação as propostas presentes na literatura, os resultados aqui apresentados demonstram mais uma vez que a utilização da infra-estrutura *small world* incorpora características às RSSFs que tornam o funcionamento dos protocolos de reprogramação mais eficientes, também para o cenário com *multicast*.

4.5 Comentários Finais

Este capítulo apresentou os resultados de simulação do protocolo proposto quando comparado ao protocolo estado da arte presente na literatura. Os resultados aqui apresentados evidenciaram que o protocolo proposto apresenta resultados mais satisfatórios que o protocolo estado da arte nos cenários avaliados, principalmente com relação ao tempo gasto na reprogramação da rede. A explicação para tal resultado deve-se a utilização da estrutura de rede *small world*, que reduz o caminho mínimo médio na rede, acelerando o processo de reprogramação. A estrutura *small world* foi também a principal responsável pela diminuição do número de mensagens trocadas durante a reprogramação de apenas um subconjunto dos nós, cenário em que é utilizado o nosso protocolo de reprogramação com *multicast*.

Capítulo 5

Conclusões e Trabalhos Futuros

Reprogramação é um serviço fundamental para as RSSFs já que gerenciamento e manutenção são atividades desafiadoras neste tipo de rede. Neste trabalho, foi proposto um novo protocolo de reconfiguração, chamado *OAP-SW*, o qual reconfigura redes com características *small world*. Neste tipo de rede, um pequeno número de atalhos é criado entre os nós *H-sensors*, resultando em uma rede com características desejáveis para as RSSFs, como por exemplo, caminho médio mínimo pequeno. Além disso, foi proposta uma extensão ao protocolo de forma a permitir que apenas parte dos nós da rede sejam reconfigurados, solução essa, obtida através da utilização de *multicast* para RSSFs.

Os resultados aqui apresentados, demonstraram que o protocolo *OAP-SW* é mais eficiente nos cenários avaliados que a solução estado-da-arte presente na literatura. Foi mostrado que a combinação do protocolo de reconfiguração com um algoritmo para o estabelecimento de uma infra-estrutura de rede *small world*, resultou na diminuição do tempo gasto na reconfiguração da rede, métrica fundamental aos protocolos de reconfiguração já que muitas aplicações para RSSFs são altamente sensíveis ao tempo gasto durante o processo, ou seja, longos tempos gasto na reconfiguração podem inviabilizar o funcionamento de algumas aplicações. Foi mostrado que o número de mensagens trocadas também sofreu uma diminuição notável, apesar de não ser tão significativa quanto a redução do tempo de reconfiguração. Para os cenários com *multicast* foi mostrado que a utilização da infra-estrutura *small world* também resultou na melhora do comportamento do algoritmo. Tais resultados são apenas mais uma evidência de que a utilização de modelos de redes complexas podem gerar soluções simples e robustas para problemas em RSSFs.

Como trabalhos futuros, pretende-se:

- Avaliar o protocolo proposto levando-se em conta outros parâmetros importantes

no processo de reconfiguração, como por exemplo, o impacto do tamanho do programa no comportamento do algoritmo.

- Pretende-se verificar qual é o consumo de energia dentro do nó durante o processo de reconfiguração ao se utilizar uma pequena quantidade de nós *H-sensors*. Avaliar tal métrica é importante, já que uma possível crítica à solução proposta é que a utilização de nós com raio de comunicação de longa distância aumentaria o consumo de energia da rede. No entanto, é importante destacar que apesar dos nós *H-sensors* possuírem um maior raio de comunicação e consumirem maior quantidade de energia, eles ainda possuem um *hardware* diferenciado e consequentemente uma fonte de energia diferente da dos nós *L-sensors*.
- Como os nós devem receber o código do programa de forma completa, pretende-se implementar um mecanismo de detecção e correção de falhas integrado ao OAP-SW, que garanta que os nós alvo sejam reconfigurados com sucesso.
- Em relação à proposta de *multicast*, pretende-se avaliar a qualidade da árvore *multicast* que está sendo gerada e propor uma heurística mais eficiente que a utilizada neste trabalho.
- Como muitas vezes o código a ser propagado na rede é grande, pretende-se estudar e propor técnicas de compactação de dados que podem ser empregadas de forma satisfatória para a reprogramação das RSSFs.

Referências Bibliográficas

- [Agre & Clare, 2000] Agre, J. & Clare, L. (2000). An integrated architecture for cooperative sensing networks. *Computer*, 33(5):106--108. ISSN 0018-9162.
- [Akyildiz et al., 2002] Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102--114.
- [Basagni et al., 2001] Basagni, S.; Chlamtac, I. & Syrotiuk, V. (2001). Location aware, dependable multicast for mobile ad hoc networks. *Computer Networks*, 36(5-6):659--670.
- [Bose et al., 2001] Bose, P.; Morin, P.; Stojmenović, I. & Urrutia, J. (2001). Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609--616.
- [Boukerche, 2008] Boukerche, A. (2008). *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press.
- [Cerpa et al., 2001] Cerpa, A.; Elson, J.; Hamilton, M.; Zhao, J.; Estrin, D. & Girod, L. (2001). Habitat monitoring: application driver for wireless communications technology. Em *SIGCOMM LA '01: Workshop on Data communication in Latin America and the Caribbean*, pp. 20--41, New York, NY, USA. ACM.
- [Chong & Kumar, 2003] Chong, C. & Kumar, S. (2003). Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247--1256.
- [Crossbow Technology, 2003] Crossbow Technology, I. (2003). Mote in-network programming user reference. www.tinyos.net/tinyos-1.x/doc/Xnp.pdf.
- [Estrin et al., 1999] Estrin, D.; Govindan, R.; Heidemann, J. & Kumar, S. (1999). Next century challenges: scalable coordination in sensor networks. Em *MobiCom*

- '99: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 263--270, New York, NY, USA. ACM.
- [Gao et al., 2005] Gao, T.; Greenspan, D.; Welsh, M.; Juang, R. & Alm, A. (2005). Vital signs monitoring and patient tracking over a wireless network. *Proceedings of the 27th IEEE EMBS Annual International Conference*, 1(1):102--105.
- [Group, 2008] Group, E. D. C. (2008). Sinalgo - simulator for network algorithms. <http://dcg.ethz.ch/projects/sinalgo/>.
- [Gui & Mohapatra, 2004] Gui, C. & Mohapatra, P. (2004). Scalable multicasting in mobile ad hoc networks. Em *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pp. 2119--2129. ISSN 0743-166X.
- [Guidoni et al., 2008a] Guidoni, D. L.; Mini, R. A. & Loureiro, A. A. (2008a). On the design of heterogeneous sensor networks based on small world concepts. Em *MSWiM '08: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 309--314, New York, NY, USA. ACM.
- [Guidoni et al., 2008b] Guidoni, D. L.; Mini, R. A. F. & Loureiro, A. A. F. (2008b). Creating small-world models in wireless sensor networks. Em *19th International Symposium on Personal, Indoor and Mobile Radio Communications. (PIMRC'08)*, pp. 1--6.
- [Gupta & Kumar, 2000] Gupta, P. & Kumar, P. (2000). The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388--404.
- [Hagedorn et al., 2008] Hagedorn, A.; Starobinski, D. & Trachtenberg, A. (2008). Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes. *Information Processing in Sensor Networks*, 22(24):457--466.
- [Hu et al., 2004] Hu, W.; Bulusu, N. & Jha, S. (2004). A communication paradigm for hybrid sensor/actuator networks. Em *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, volume 1, pp. 201--205.
- [Hui & Culler, 2004] Hui, J. W. & Culler, D. (2004). The dynamic behavior of a data dissemination protocol for network programming at scale. Em *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 81--94, New York, NY, USA. ACM.

- [Intanagonwiwat et al., 2000] Intanagonwiwat, C.; Govindan, R. & Estrin, D. (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks. Em *In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00)*, pp. 56--67.
- [Jeong & Culler, 2004] Jeong, J. & Culler, D. (2004). Incremental network programming for wireless sensors. Em *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 25--33.
- [Kahn et al., 1999] Kahn, J. M.; Katz, R. H. & Pister, K. S. J. (1999). Next century challenges: mobile networking for "smart dust". Em *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 271--278, New York, NY, USA. ACM.
- [Kleinberg, 2008] Kleinberg, J. (2008). The convergence of social and technological networks. *Commun. ACM*, 51(11):66--72. ISSN 0001-0782.
- [Kulkarni & Wang, 2009] Kulkarni, S. & Wang, L. (2009). Energy-efficient multihop reprogramming for sensor networks. *ACM Transactions on Sensor Networks*, 5(2):1-40.
- [Lee et al., 2002] Lee, S.; Su, W. & Gerla, M. (2002). On-demand multicast routing protocol in multihop wireless mobile networks. *Mobile Networks and Applications*, 7(6):441--453.
- [Levis & Culler, 2004] Levis, P. & Culler, D. (2004). The firecracker protocol. Em *EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, p. 3, New York, NY, USA. ACM.
- [Levis et al., 2004] Levis, P.; Patel, N.; Culler, D. & Shenker, S. (2004). Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. Em *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, pp. 2--2, Berkeley, CA, USA. USENIX Association.
- [Lorincz et al., 2004] Lorincz, K.; Malan, D. J.; Fulford-Jones, T. R. F.; Nawoj, A.; Clavel, A.; Shnayder, V.; Mainland, G.; Welsh, M. & Moulton, S. (2004). Sensor networks for emergency response: Challenges and opportunities. *IEEE Pervasive Computing*, 3(4):16--23. ISSN 1536-1268.

- [Maia et al., 2009] Maia, G.; Guidoni, D. L.; Aquino, A. L. & Loureiro, A. A. (2009). Improving an over-the-air programming protocol for wireless sensor networks based on small world concepts. Em *MSWiM '09: Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp. 261--267, New York, NY, USA. ACM.
- [Mainwaring et al., 2002] Mainwaring, A.; Culler, D.; Polastre, J.; Szewczyk, R. & Anderson, J. (2002). Wireless sensor networks for habitat monitoring. Em *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 88--97, New York, NY, USA. ACM.
- [Naik et al., 2005] Naik, V.; Arora, A.; Sinha, P. & Zhang, H. (2005). Sprinkler: A reliable and energy efficient data dissemination service for wireless embedded devices. Em *RTSS '05: Proceedings of the 26th IEEE International Real-Time Systems Symposium*, pp. 277--286, Washington, DC, USA. IEEE Computer Society.
- [Newman, 2003] Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45:167--256.
- [Ni et al., 1999] Ni, S.-Y.; Tseng, Y.-C.; Chen, Y.-S. & Sheu, J.-P. (1999). The broadcast storm problem in a mobile ad hoc network. Em *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 151--162, New York, NY, USA. ACM.
- [Pottie & Kaiser, 2000a] Pottie, G. J. & Kaiser, W. J. (2000a). Embedding the Internet: wireless integrated network sensors. *Communications of the ACM*, 43(5):51--51. ISSN 0001-0782.
- [Pottie & Kaiser, 2000b] Pottie, G. J. & Kaiser, W. J. (2000b). Wireless integrated network sensors. *Communications of the ACM*, 43(5):51--58. ISSN 0001-0782.
- [Ruiz & Gomez-Skarmeta, 2005] Ruiz, P. & Gomez-Skarmeta, A. (2005). Approximating optimal multicast trees in wireless multihop networks. Em *10th IEEE Symposium on Computers and Communications, 2005. ISCC 2005. Proceedings*, pp. 686--691.
- [Sanchez et al., 2006] Sanchez, J.; Ruiz, P. & Stojmenovic, I. (2006). GMR: Geographic multicast routing for wireless sensor networks. *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, 1.

- [Sanchez et al., 2007] Sanchez, J. A.; Ruiz, P. M. & Stojmenovic, I. (2007). Energy-efficient geographic multicast routing for sensor and actuator networks. *Computer Communications*, 30(13):2519--2531. ISSN 0140-3664.
- [Shoup, 2005] Shoup, V. (2005). *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press.
- [Simon et al., 2004] Simon, G.; Maróti, M.; Ákos Lédeczi; Balogh, G.; Kusy, B.; Nádas, A.; Pap, G.; Sallai, J. & Frampton, K. (2004). Sensor network-based counter-sniper system. Em *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 1--12, New York, NY, USA. ACM.
- [Stathopoulos et al., 2003] Stathopoulos, T.; Heidemann, J. & Estrin, D. (2003). A remote code update mechanism for wireless sensor networks. Relatório técnico, University of California, LA.
- [Wang, 2004] Wang, L. (2004). Mnp: multihop network reprogramming service for sensor networks. Em *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 285--286, New York, NY, USA. ACM.
- [Warneke et al., 2001] Warneke, B.; Last, M.; Liebowitz, B. & Pister, K. S. J. (2001). Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44-51.
- [Watts, 2007] Watts, D. J. (2007). A twenty-first century science. *Nature*, 445(2):489.
- [Watts & Strogatz, 1998] Watts, D. J. & Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684):440--442.
- [Wu & Candan, 2006] Wu, S. & Candan, K. S. (2006). Gmp: Distributed geographic multicast routing in wireless sensor networks. Em *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, pp. 49--49, Washington, DC, USA. IEEE Computer Society.
- [Yarvis et al., 2005] Yarvis, M.; Kushalnagar, N.; Singh, H.; Rangarajan, A.; Liu, Y. & Singh, S. (2005). Exploiting heterogeneity in sensor networks. Em *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pp. 878--890. ISSN 0743-166X.
- [Yick et al., 2005] Yick, J.; Mukherjee, B. & Ghosal, D. (2005). Analysis of a prediction-based mobility adaptive tracking algorithm. Em *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, pp. 753--760 Vol. 1.

- [Yick et al., 2008] Yick, J.; Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292--2330.