



COPPE/UFRJ

PROTOCOLOS PARA REDES DE COMUNICAÇÃO AD HOC
ENDEREÇADAS POR INTERESSES

Rodrigo Souza Granja

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Cláudio Luís de Amorim

Rio de Janeiro
Junho de 2010

PROCOLOS PARA REDES DE COMUNICAÇÃO AD HOC
ENDEREÇADAS POR INTERESSES

Rodrigo Souza Granja

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Cláudio Luís de Amorim, Ph.D.

Prof. Aloysio de Castro Pinto Pedroza, D.Sc.

Prof. Luís Felipe Magalhães de Moraes, Ph.D.

Prof. Orlando Gomes Loques Filho, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2010

Granja, Rodrigo Souza

Protocolos para Redes de Comunicação Ad hoc Endereçadas por Interesses/Rodrigo Souza Granja. – Rio de Janeiro: UFRJ/COPPE, 2010.

XVII, 151 p.: il.; 29, 7cm.

Orientador: Cláudio Luís de Amorim

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2010.

Referências Bibliográficas: p. 88 – 97.

1. Protocolo de comunicação. 2. Rede Social. 3. Endereçamento por interesse. I. Amorim, Cláudio Luís de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

À minha família.

Agradecimentos

Ao longo deste trabalho, contei com a ajuda e o apoio de diversas pessoas. Gostaria de agradecer ao meu orientador, Cláudio Luís de Amorim, pelos valiosos conhecimentos transmitidos ao longo das várias disciplinas, reuniões e seminários.

Agradeço aos colegas de laboratórios pelos debates e trocas de informações. Gostaria de agradecer especialmente ao Diego Dutra, Héberte Fernandes e Renato Dutra, pela ajuda com a infra-estrutura do laboratório e todos os conhecimentos trocados.

Gostaria de agradecer também aos Professores Aloysio Pedroza, Luís Felipe de Moraes e Orlando Loques Filho por terem aceito o convite para participar da minha banca de defesa de mestrado.

Agradeço ao Professor Alexandre Sztajnberg pelas oportunidades a mim oferecidas durante a minha graduação na UERJ, que foram fundamentais para o meu ingresso no mestrado.

Agradeço aos meus pais por todo apoio e por ter me dado, ao longo de toda a minha vida, as condições para realizar este trabalho. Agradeço também ao restante da minha família, especialmente à minha tia Olena, que sempre me incentivou.

Agradeço também à Petrobras, representada pelo gerente Adelino da Silva, pela liberação, em tempo parcial, das minhas atividades profissionais para a realização das disciplinas e participação nas reuniões semanais do mestrado.

Finalmente, agradeço aos meus amigos pelos momentos de alegria que foram fundamentais para o meu equilíbrio ao longo do mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PROTOCOLOS PARA REDES DE COMUNICAÇÃO AD HOC
ENDEREÇADAS POR INTERESSES

Rodrigo Souza Granja

Junho/2010

Orientador: Cláudio Luís de Amorim

Programa: Engenharia de Sistemas e Computação

Em uma típica rede de comunicação digital, os nós da rede trocam mensagens entre si utilizando comumente o endereçamento fonte-destino, no qual uma mensagem é enviada de um nó-fonte para um nó-destino, também conhecido como endereçamento fim-a-fim. Diferentemente, Redes de Comunicação Endereçadas Por Interesse (REPIs), não requerem endereçamento fim-a-fim, pois utilizam os interesses dos usuários para roteamento e endereçamento das mensagens.

Nesta dissertação, é desenvolvida e avaliada REPI-A (REPI Ad hoc) um protocolo de comunicação para REPIs funcionarem em redes de comunicação sem fio Ad hoc. Para isso, foram implementados um middleware e uma aplicação mensageira utilizando uma rede de sensores sem fio com vinte dispositivos usando três topologias e mais de cinquenta configurações de rede, ambos monitorados em tempo real através de um sistema especialmente desenvolvido para configuração e monitoração de redes ad-hocs reais. REPI-A também foi avaliada comparativamente com algoritmos representativos probabilísticos (Gossip) e de inundação (flooding) para redes Ad hoc.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

COMMUNICATION PROTOCOLS FOR AD HOC NETWORKS ADDRESSED
BY INTERESTS

Rodrigo Souza Granja

June/2010

Advisor: Cláudio Luís de Amorim

Department: Systems Engineering and Computer Science

In a typical digital communication network, the network nodes exchange messages using commonly the source-destination addressing, in which a message is sent from a source-node to a destination-node, also known as end-to-end addressing. Differently, Communication Networks Addressed By Interests (NABIs) require no end-to-end addressing, since they utilize users' defined interests for both routing and addressing of messages.

In this dissertation, it is developed and evaluated a communication protocol for wireless ad hoc NABIs (WANABIs). To this end, a middleware and a messaging application were implemented in a 20-node wireless sensor network for three different topologies and more than fifty network settings, with both being monitored in real time using a system specially developed for monitoring and configuration of practical ad-hoc networks. Also, WANABI's performance was comparatively evaluated against that of a Gossip-type and Flooding representative algorithms for Ad hoc networks.

Sumário

Lista de Figuras	xii
Lista de Tabelas	xvii
1 Introdução	1
1.1 Motivação	3
1.2 Contribuições do Trabalho	3
2 Redes Endereçadas Por Interesse	5
2.1 Exemplo de formação de uma REPI	7
2.2 REPI-A	9
2.2.1 Tipos de mensagens da REPI-A	12
2.3 Middleware Tmote	14
2.4 Protocolo REPI-A	16
2.4.1 Implementação do Prefixo C	17
2.5 Comparação da REPI-A com outro modelos	18
2.6 Discussão	21
2.6.1 Colaboração	21
3 Uso da REPI-A	24
3.1 Utilização da REPI-A por usuários reais	24
3.1.1 Escolha das características do participante da REPI-A	28
3.2 Execuções Automatizadas	29
3.2.1 Módulo de Execução Automatizada de SAMCRA	30
3.3 Desenvolvimento de outras aplicações mensageiras para REPI-A	31
3.3.1 Formação de grupos de pesquisas	32
3.3.2 Comercio eletrônico	32
3.3.3 Apoio em situações de emergência	33
4 Avaliação Experimental	35
4.1 Ambiente Experimental	36
4.1.1 Configuração do ambiente	37

4.2	Métricas	37
4.2.1	Notação usada nas métricas	37
4.2.2	Taxa de Entrega de Mensagens (Métrica 1)	39
4.2.3	Número de saltos (Métrica 2)	40
4.2.4	Tempo médio de Entrega (Métrica 3)	40
4.2.5	Mensagens Colaborativas (Métrica 4)	41
4.2.6	Porcentagem de Nós Colaboradores (Métrica 5)	41
4.2.7	Custo da Colaboração (Métrica 6)	41
4.2.8	Numero de Nós Colaboradores (Métrica 7)	42
4.3	Procedimento experimental	42
4.3.1	Experimento 1 - Variação de mensagens na rede	42
4.3.2	Experimento 2 - Impacto da variação de destinatários na rede	42
4.3.3	Experimento 3 - Variação do número de características no prefixo	43
4.3.4	Experimento 4 - Tamanho da rede	43
4.4	Topologia 1	43
4.4.1	Definição dos parâmetros de execução	43
4.4.2	Configuração e distribuição dos Tmotes	45
4.4.3	Resultados das execuções dos experimentos na Topologia 1	45
4.5	Topologia 2	47
4.5.1	Entrega de mensagens em uma execução na REPI-A	48
4.5.2	Experimento 1	55
4.5.3	Experimento 2	58
4.5.4	Experimento 3	61
4.5.5	Experimento 4	64
4.5.6	Discussão dos resultados da Topologia 2	65
4.6	Topologia 3	66
4.6.1	Configuração do Experimento	68
4.6.2	Resultados	68
5	Trabalhos Relacionados	71
5.1	Algoritmos de roteamento para redes Ad Hoc	71
5.1.1	<i>Ad hoc On-demand Distance Vector</i>	72
5.1.2	DSR	74
5.1.3	Gossip	75
5.1.4	Resumo comparativo	76
5.2	Aplicações de redes sociais para redes Ad Hoc	77
5.2.1	MoCoTo	78
5.2.2	MobiClique	80
5.2.3	Nokia Sensor	82

5.2.4	Comparativo das aplicações de redes sociais para redes Ad Hoc	84
6	Conclusão	85
6.1	Trabalho Futuro	86
	Referências Bibliográficas	88
A	Resultados dos Experimentos da Topologia 1	98
A.1	Experimento 1	98
A.1.1	Taxa de Entrega de Mensagens em função do número de mensagens na rede	98
A.1.2	Número de saltos	100
A.2	Experimento 2	101
A.2.1	Taxa de Entrega de Mensagens em função do número de destinatários na rede	101
A.2.2	Porcentagem de Mensagens Encaminhadas de forma colaborativa	103
A.2.3	Porcentagem de nós colaboradores	103
A.2.4	Custo Colaboração	104
A.3	Experimento 3	104
A.3.1	Número de nós colaboradores	105
A.3.2	Taxa de Entrega de Mensagens em função do algoritmo de roteamento	105
B	Manual da Aplicação Mensageira	108
B.1	Parâmetros de inicialização	108
B.2	A interface gráfica da aplicação	110
B.3	Tela de escolha de motes	110
B.4	Tela de Configuração Inicial	111
B.5	Modos padrão e administrador	111
B.6	Mapa de usuários da rede	114
B.7	Administração pelo usuário	115
B.8	Árvore de informações dos participantes	116
B.9	Controle de Tráfego	118
B.10	Configurações da Rede	119
B.11	Execuções Automatizadas	119
B.12	Módulo de contagem de tempo de envio de mensagens	120
C	Redes Ad Hoc Sem Fio	122

D	Padrão 802.15.4 (<i>Zigbee</i>)	124
D.1	Dispositivos de uma rede 802.15.4/ZigBee	125
D.2	Topologias de uma rede 802.15.4/ZigBee	125
D.2.1	Topologia em Estrela	126
D.2.2	Topologia em Malha	126
D.2.3	Topologia de múltiplas árvores	127
D.3	A camada física (PHY) do padrão 802.15.4	127
D.4	Camada de Controle e Acesso ao Meio (MAC) do padrão 802.15.4	129
D.5	Comparação 802.15.4/ZigBee e BlueTooth	131
E	SAMCRA	133
E.1	Módulo de Coleta de dados	133
E.1.1	Tabela Dados de participação por Tmote	133
E.1.2	Tabelas com detalhamento de mensagem recebidas por Tmote	135
E.1.3	Tabela Quantidade de Campos coincidentes por Tmote	136
E.1.4	Tabela com rastreamento de mensagens (<i>trace</i>)	137
E.2	Módulos de Configuração e Execução Automatizada	138
E.2.1	Configuração dos parâmetros da REPI-A	139
E.2.2	Nome dos canais públicos	140
E.2.3	Configuração da execução automática	140
F	Sistema Operacional TinyOS	144
F.1	Arquitetura do TinyOS 1.1	144
F.2	Modelo de concorrência do TinyOS	145
F.3	Operações divididas em fase	146
F.4	Modelo de memória do TinyOS	147
G	Tmote Sky	148
G.1	Rádio do Tmote Sky	149
G.2	A antena do Tmote Sky	150

Lista de Figuras

2.1	Faixas de interesses representadas em eixos cartesianos. Figura adaptada de [1]	6
2.2	Raio de comunicação dos nós utilizados no exemplo de formação da REPI.	8
2.3	Diagrama de implantação da REPI-A	11
2.4	Detalhe dos Leds utilizados para depuração do middleware Tmote . .	14
2.5	Representação da pilha de protocolos utilizados para a formação da REPI-A.	16
2.6	Exemplo da formação de uma REPI.	20
3.1	Tela de troca de mensagens da aplicação mensageira.	25
3.2	Abas de seleção de canais.	26
3.3	Painel de escolha de canal público.	27
3.4	Canais customizado e reservado.	27
3.5	Tela de Configuração Inicial da aplicação.	29
3.6	Tela de ativação da execução automática, que faz a aplicação enviar mensagens na rede.	29
3.7	Tela com as opções de parametrização de SAMCRA para a realização de uma execução automatizada.	31
4.1	Mapa com a distribuição espacial dos Tmotes na topologia 1.	46
4.2	Cones utilizados para suspender os Tmotes e evitar interferências do solo na transmissão na topologia 1.	46
4.3	Mapa com a distribuição espacial dos Tmotes na topologia 2.	47
4.4	Cones utilizados para suspender os Tmotes e evitar interferências do solo na transmissão na topologia 2.	48
4.5	Análise da entrega de mensagens em apenas um salto na configuração com apenas um Tmote enviando mensagens.	49
4.6	Padrão de propagação de rádio-frequência da antena do Tmote Sky, adaptado de [2]	50

4.7	Análise do número do total de mensagens, mensagens aceitas pelo destinatário e mensagens descartadas em função do prefixo C.	51
4.8	Análise do número de vizinhos de Rádio Frequência e da REPI por nó.	51
4.9	Gráfico que apresenta a relação entre o número de vizinhos de rádio frequência e o Total de Mensagens Recebidas por cada nó.	52
4.10	Análise das mensagens entregues ao nó 3 na execução estudada	53
4.11	Análise das mensagens entregues ao nó 9 na execução estudada.	54
4.12	Análise das mensagens entregues ao nó 9 em média nas vinte execuções realizadas.	54
4.13	Gráfico que mostra o comparativo de dois experimentos realizados com o mesmo intervalo de tempo entre o envio de duas mensagens consecutivas, variando apenas o número de mensagens entregues.	56
4.14	Gráfico que mostra a evolução da Taxa Entrega de Mensagens (\overline{TEM}) em função do número de Nós Fonte e do intervalo de tempo entre o envio de duas mensagens consecutivas.	57
4.15	Gráfico que mostra a evolução da Taxa de Entrega de Mensagens (\overline{TEM}) em função do número de Nós Destinatários, com três configurações de intervalo de tempo.	59
4.16	Gráfico que mostra a evolução do Custo de Colaboração (\overline{CCOL}) em função do número de Nós Destinatários, com três configurações de intervalo de tempo.	60
4.17	Gráfico que mostra a evolução do custo de encaminhamento das mensagens não colaborativas em função do número de Nós Destinatários, com três configurações de intervalo de tempo.	60
4.18	Gráfico que mostra a relação entre o Custo de Envio de Mensagens e a Taxa de Entrega de Mensagens (\overline{TEM}) de acordo com o algoritmo utilizado.	61
4.19	Gráfico que mostra a relação entre o Custo de Envio de Mensagens e a Taxa de Entrega de Mensagens (\overline{TEM}) de acordo com o algoritmo utilizado.	62
4.20	Gráfico que mostra o número médio de saltos de todas as mensagens e apenas das mensagens entregues aos destinatários, de acordo com o algoritmo utilizado.	63
4.21	Representação de quatro configurações do Experimento 4. Na configuração mostrada em 1 há quatro nós na rede, em 2 há oito nós, doze na 3 e dezesseis nós na 4.	64
4.22	Gráfico que mostra a variação do Tempo Médio de Entrega de mensagens na REPI-A em função do número de nós, em medições com e sem o <i>overhead</i> da aplicação mensageira.	65

4.23	Mapa com a distribuição espacial dos Tmotes na topologia 3.	67
4.24	Gráfico que mostra a comparação da variação da Taxa de Entrega de Mensagens e do Custo de Entrega de Mensagens com três configurações diferentes da REPI-A, duas configurações do protocolo <i>Gossip</i> e o protocolo de inundação.	69
4.25	Gráfico que mostra a quantidade de mensagens entregues em todas as execuções para as três configurações analisadas.	69
5.1	Exemplo da aplicação MoCoTo para realizar avaliações de aprendizado. Na figura da esquerda o menu principal da aplicação, no da direita uma das questões do teste, retirada de [3].	79
5.2	Diagrama que exibe a formação da rede MobiClique, adaptado de [4].	81
5.3	Exemplo de das informações que o usuário insere no seu perfil do Nokia Sensor, chamado Folio. Adaptado de [5].	83
5.4	Três telas do Nokia Sensor: à esquerda a tela do resultado do processo de Scan, como cinco usuários encontrados; na Figura do meio, a pasta com os arquivos compartilhados na rede Nokia Sensor; e a direita um alerta avisando que um usuário marcado como favorito está ao alcance do rádio Bluetooth.	83
A.1	Gráfico que mostra a evolução da Taxa de entrega de Mensagens (\overline{TEM}) em função do número de mensagens enviadas, com duas configurações da REPI-A.	99
A.2	Média do número de saltos em função do número de mensagens geradas e com duas configurações da REPI-A.	100
A.3	Gráfico que mostra a evolução da Taxa de Entrega de Mensagens (\overline{TEM}) em função do número de destinatários com três configurações da REPI-A.	102
A.4	Gráfico que mostra a quantidade de mensagens descartadas por ausência de casamento de prefixo C com três configurações da REPI-A.	102
A.5	Gráfico que mostra a Porcentagem de Mensagens Encaminhadas de forma colaborativa (\overline{MC}) em função do número de Nós Destinatários, com duas configuração da REPI-A.	103
A.6	Gráfico que mostra a Porcentagem de nós colaboradores (\overline{NC}) em função do número de Nós Destinatários com duas configurações da REPI-A.	104
A.7	Gráfico que mostra o custo de Colaboração (\overline{CCOL}) em função do número de Nós Destinatários com quatro Configurações da REPI-A. .	105
A.8	Gráfico que mostra o número de nós colaboradores (\overline{NC}) de acordo com a variação do número de campos no Prefixo C.	106

A.9	Gráfico que mostra a Taxa de Entrega de Mensagens (\overline{TEM}) de acordo com a variação do número de campos no prefixo C e comparação com algoritmo de inundação.	106
A.10	Gráfico que mostra a evolução da relação entre o Total de Mensagens Recebidas (TMR) e de mensagens entregues ao destinatário final em função do número de campos do prefixo C e em comparação com algoritmo de inundação.	107
B.1	Tela de escolha do Tmote em uma máquina com 3 Tmotes conectados.	111
B.2	Tela de Configuração Inicial da aplicação mensageira.	112
B.3	Tela de Configuração Inicial da aplicação mensageira sem o painel de escolha do prefixo.	112
B.4	Tela de Configuração Inicial da aplicação.	112
B.5	Mapa de usuários da rede.	114
B.6	Configurações da aplicação.	115
B.7	Janela de alerta visual que notifica a chegada de novas mensagens. . .	116
B.8	Painel da árvore de informações dos participantes da rede, mostrando dados do prefixo local.	117
B.9	Painel da Árvore de Informações dos Participantes da Rede, mostrando dados do prefixo de outro participante da rede.	117
B.10	Tabela de informações das mensagens recebidas na rede, de acordo com o tipo.	118
B.11	Descrição das configurações obtidas no servidor de configuração da rede.	119
B.12	Tela de ativação da execução automática, que faz a aplicação enviar mensagens na rede.	120
B.13	Painel de contagem de tempo de envio de pacotes, com e sem <i>overhead</i> da aplicação.	121
D.1	Exemplo de topologia em estrela.	126
D.2	Exemplo de topologia em Malha.	127
D.3	Exemplo de topologia em Múltiplas Árvores.	128
D.4	Canais da camada PHY do padrão 802.15.4, adaptado de [6].	129
E.1	Visão parcial da tabela que consolida os dados sobre as mensagens trocados por cada Tmote.	134
E.2	Uma das tabelas que detalham o número de mensagens recebidas por Tmote.	136
E.3	Visão parcial da tabela que consolida o número de campos coincidentes nos prefixos das mensagens recebidas	137

E.4	Tabela com o trace de uma execução na rede REPI.	138
E.5	Tela com as opções de parametrização e execução automatizada na REPI-A.	139
G.1	Dispositivo de comunicação Tmote Sky, adaptado de [2].	149
G.2	Padrão de propagação de rádio-frequência da antena do Tmote Sky, adaptado de [2]	150

Lista de Tabelas

2.1	Tabela que mostra as característica e interesses dos participantes da REPI do exemplo apresentado.	7
2.2	Tabela de códigos dos Leds de Depuração dos Tmotes	15
2.3	Tabela com os códigos de erros do middleware Tmote	15
2.4	Tabela que apresenta como as características do Prefixo C foram modeladas na REPI-A.	18
4.1	Comparação da Taxa de Entrega de Mensagens na Topologias 1 e 2 com dois intervalos de tempo diferentes.	58
5.1	Tabela de comparação dos protocolos de roteamento para redes Ad Hoc	77
5.2	Quadro com o resumo comparativo dos trabalhos relacionados.	84
D.1	Quadro com o resumo do comparativo entre o 802.15.4/ZigBee e o BlueTooth.	132

Capítulo 1

Introdução

Ao longos das últimas três décadas acompanhamos uma grande evolução nas indústrias de telecomunicações e informática (telemática), que, através da popularização de equipamentos de comunicação móvel e da internet, trouxe profundas mudanças sociais e econômicas.

Um bom exemplo de como essa situação se reflete no Brasil é obtido a partir da análise dos relatórios anuais produzidos pela Anatel (Agência Nacional de Telecomunicações)[7]. Eles mostram que entre 1998 e 2008 houve um crescimento de mais 2000% no número de aparelhos celulares, com 150 milhões de aparelhos em 2008, contra 7,4 milhões em 1998. Já entre o ano 2000 e 2008 houve um crescimento de mais de 1000% no número de pontos de acesso a internet, com um salto de 124,5 mil para 11,4 milhões.

Esse rápido crescimento associado com a facilidade de desenvolvimento e hospedagem de sítios e distribuição de aplicações na internet, estabelece um campo fértil para a criação e popularização de muitos softwares e serviços na internet e telefonia móvel, que a cada dia estão mais integrados. Atualmente, é comum que serviços como: bancos, compras, entretenimento, notícias e comunicação pessoal sejam realizados através de celulares e da internet, em transações inteiramente digitais.

Nos últimos dez anos dois tipos de aplicações para internet tem chamado a atenção mundial e tiveram grande destaque na mídia, sendo também objeto de estudos acadêmicos: as aplicações com arquitetura par-a-par (*peer-to-peer*) e as aplicações de redes sociais (*Social Networks*).

As aplicações de redes sociais, como o Orkut[8], Facebook [9], Twitter [10] e MySpace [11], permitem que seus usuários criem páginas customizadas, chamadas de perfis, com seus dados pessoais e preferências. Além disso, também permitem a criação de comunidades, que são páginas onde os usuários podem trocar mensagens através de um fórum virtual. Além das comunidades, os usuários também podem interagir diretamente, através de mensagens para outros usuários, escritas em seus respectivos perfis.

Em suas versões mais recentes, as redes sociais permitem uma série funcionalidades extras, como o APIs (*Application Programming Interface*) para o desenvolvimento de aplicações distribuídas e colaborativas e compartilhamento de conteúdo multimídia, como álbuns de fotos e vídeos, além de opções avançadas de privacidade dos usuários.

Segundo os relatórios do sítio Alexa [12], especializado em estatísticas de tráfego de dados em sítios na Internet, em dezembro de 2009, o Facebook era o segundo sítio com mais tráfego de acessos no mundo, enquanto o MySpace era o décimo terceiro, o Twitter o décimo quarto e o orkut o nonagésimo primeiro.

Em contraste com as redes sociais, as aplicações com arquitetura par-a-par em geral tem um foco diferente: seu principal objetivo é a troca de arquivos entre os seus usuários. E, em virtude desta mudança de foco, o anonimato costuma ser uma funcionalidade desejável e há pouco ou nenhum espaço para que os usuários divulguem suas informações pessoais.

Neste tipo de rede, as aplicações participantes exercem funções tanto de servidor quanto de cliente, criando uma rede descentralizada e que muitas vezes dependem de servidores centralizados apenas para o seu processo de inicialização. Alguns exemplos de programas que utilizam esta arquitetura são o LimeWire[13], o Kazaa [14], o emule [15] e o Bit Torrent [16].

A principal vantagem deste modelo é que é possível criar uma rede com muitos usuários, com grande quantidade de arquivos e altas taxas de transmissão de dados sem que seja necessário a manutenção e instalação de servidores que centralizem todo o trabalho: a maior parte dele é distribuída pelos próprios usuários. Com isso, essas redes demonstram grande capacidade de escalabilidade e se beneficiam dela.

Embora as redes sociais e a arquitetura P2P estejam consolidadas na Internet, o seu uso em aparelhos móveis, em especial os celulares, ainda é incipiente. Em sua maioria, as aplicações de redes sociais para celulares são adaptações das aplicações da Internet para navegadores de dispositivos móveis, que dispõe de menos recursos e uma tela reduzida. Porém, funcionam da mesma forma: o usuário acessa o serviço em um servidor remoto através de um navegador Internet e a aplicação não leva em conta a sua localidade geográfica e outros possíveis usuários que possam estar em sua proximidade.

Além disso, a grande maioria das formas de interação oferecidas pelos aparelhos de telefones celulares, como, por exemplo, serviços de voz, serviços de mensagens curtas (SMS) e serviços de mensagens de multimídia (MMS) são centralizadas e precisam da infra-estrutura paga para funcionar. E há pouca utilização de modelos descentralizados que se aproveitem das vantagens oferecidas pelas aplicações P2P para a criação de grandes redes.

Esse cenário se deve a vários fatores, mas dois podem ser destacados: a ausência

de uma padrão tecnológico amplamente difundido para a criação de redes distribuídas de grande porte em celulares; e a falta de interesse das operadoras neste tipo de modelo, uma vez que seus planos de negócio são voltadas para a venda de serviços centralizados, que só elas são capazes de oferecer.

Apesar da grande usabilidade dos serviços disponíveis, ainda existe uma gama de aplicações que não são atendidas por essas tecnologias. Especialmente aquelas onde existe a necessidade de comunicação e formação de redes sociais e grupos de interesses, mas não há infra-estrutura fixa disponível ou em que a localidade física seja um fator determinante para a formação dos grupos.

Através da tecnologia atual, é possível, por exemplo, encontrar e integrar pessoas ao redor do mundo interessadas nos mais variados assuntos. Porém, ainda não há uma forma eficiente de fazer o mesmo com pessoas que estão fisicamente próximas, como em um transporte público, um parque ou em um congresso.

1.1 Motivação

A junção do conceito de aplicações de redes sociais e dispositivos móveis é uma tendência tecnológica que é objeto de estudo de vários trabalhos [3–5, 17–28].

Apesar dos diversos estudos, esta ainda é uma área nova, em que ainda existem questões em aberto em relação ao hardware, protocolos de comunicação, estratégias de roteamento e modelagem de aplicações.

O trabalho proposto em [1] apresenta um novo modelo de comunicação para dispositivos móveis, chamado de MOI - Modelo Orientado a Interesses. Através deste modelo é possível construir redes sociais em dispositivos móveis criando Redes Endereçadas Por Interesse (REPI). A principal diferença desta proposta é que as características das redes sociais são usadas não só para modelar aplicações, mas também para realizar o roteamento e endereçamento da rede, que é feito através de comparação de palavras com interesses dos participantes.

Em uma REPI o roteamento e endereçamento da rede são feito de acordo com características e interesses do usuário, ao invés de características da rede, como outras propostas da literatura. Neste trabalho, apresentamos a arquitetura REPI-A, uma implementação para redes Ad Hoc sem fio¹ da REPI.

1.2 Contribuições do Trabalho

As principais contribuições deste trabalho são os seguintes:

¹Para melhorar a organização deste texto, as redes Ad Hoc sem fio serão chamadas apenas de redes Ad hoc. Mais detalhes sobre este tipo de rede são apresentados no Apêndice C.

Modelar uma REPI

Estabelecer as bases necessárias para implementar o MOI em uma arquitetura computacional distribuída. Esta modelagem é genérica o suficiente para ser a base para a implementação do modelo em qualquer tipo de dispositivo móvel sem fio capaz de criar uma rede Ad Hoc;

Implementar uma REPI

Através da modelagem do MOI, criar uma implementação da Rede Endereçada Por Interesses para redes Ad Hoc. Ela é reconfigurável e permite que vários parâmetros sejam alterados e avaliados;

Avaliar a implementação de uma REPI

Criar a instrumentação computacional necessária para avaliar a implementação da REPI e, através de métricas que avaliam a capacidade, escalabilidade e eficiência, comparar seu desempenho com o de um protocolo de roteamento por inundação (*flooding*) e um algoritmo com transmissão probabilística (*Gossip*);

SAMCRA

Como parte da instrumentação desenvolvida para avaliar a REPI-A, foi desenvolvido o Sistema de Automação, Monitoração e Configuração de Redes Ad Hoc [29]. Este sistema auxilia o processo de avaliação de protocolos de comunicação para redes Ad Hoc, com módulos para parametrizar a rede, coletar dados e realizar execuções automatizadas. E, embora tenha sido desenvolvido para avaliar a REPI-A, é flexível o suficiente para avaliar outros protocolos em outros ambientes de redes Ad Hoc.

O restante deste texto está organizado da seguinte forma: O Capítulo 2 apresenta as Redes Endereçadas Por Interesse. O Capítulo 3 apresenta as formas de utilização da REPI-A. O Capítulo 4 apresenta a avaliação experimental da REPI-A e sua comparação com outros protocolos da literatura. O Capítulo 5 descreve os trabalhos relacionados, mostrando as similaridades, diferenças e contribuições. Finalmente, o Capítulo 6 apresenta as conclusões e possibilidades de trabalho futuro.

Capítulo 2

Redes Endereçadas Por Interesse

As Redes Endereçadas Por Interesses (REPI) foram propostas originalmente em [1]. A REPI usa como arcabouço teórico o Modelo de Comunicação Distribuída Orientada ao Interesse (MOI), um modelo de fila de mensagens (*Message Queue*) assíncrono, em que não existe armazenamento de mensagens.

Neste modelo, tanto o roteamento da rede quanto o seu endereçamento são determinados por termos (sequência de caracteres) que tenham um significado para o usuário. Com isso, a rede passa a ser focada nos usuários e não nos equipamentos, como acontece em redes convencionais.

Uma mensagem REPI é composta de duas partes: uma é o cabeçalho da mensagem, que é chamado de prefixo e contém os termos escolhidos por cada usuário; e a outra são os dados que serão transmitidos (*Payload*).

O Prefixo também é dividido em duas partes: uma com as características do usuário e outra com os seus interesses. Essas duas partes são descritas a seguir.

Características

As características do usuário são termos cuja probabilidade de ocorrência seguem preferencialmente a distribuição normal¹ e que são usados para a formação de grupos e roteamento de mensagens na rede. Alguns exemplos de características previstas são cor do cabelo, faixa etária e faixa de peso. Neste texto, este conjunto de características também é chamado de Prefixo C.

É importante ressaltar que a adequação da escolha destas características não faz parte do escopo deste trabalho, que busca apenas avaliar a aplicabilidade da distribuição normal. Por isso, nos experimentos realizados, esses campos foram preenchidos randomicamente através desta distribuição.

¹É importante notar que a distribuição normal é a usada na proposta atual da REPI e que outras distribuições probabilísticas podem ser avaliadas. Porém, esta avaliação está fora do escopo deste trabalho.

Quando uma mensagem é recebida por um nó, as suas características são comparadas uma a uma e, seguindo uma política de encaminhamento pré-definida, ela pode ser encaminhada ou descartada.

É possível ter políticas restritivas, onde uma mensagem só é encaminhada se todas as características de ambos forem iguais, ou mais relaxadas, em que ela é encaminhada se, por exemplo, pelo menos uma ou duas características forem iguais.

Com isso, são criados grupos na rede com participantes que possuem características em comum que atendem a política de encaminhamento da rede, formando a REPI, que é uma rede sobreposta (*Overlay Network*) à rede Ad Hoc.

Interesses

Os interesses são termos cuja probabilidade de ocorrência seguem preferencialmente a distribuição Zipf. O objetivo desta abordagem é que a distribuição Zipf promova o endereçamento e, no limite, a identificação de um usuário na rede, devido à probabilidade de ocorrência de termos pouco usados ser pequena.

Os interesses de uma pessoa ou organização costumam variar muito, podendo ser categorizados em faixas de interesses, como por exemplo: faixa pessoal, social ou profissional, como em [8, 9, 30].

Para melhor compreender esta proposta, as faixas e interesses podem ser dispostas em eixos cartesianos, como mostra a Figura 2.1, onde existem várias categorias de interesses, chamadas de faixas e, nelas são distribuídos os interesses, chamados de canais.

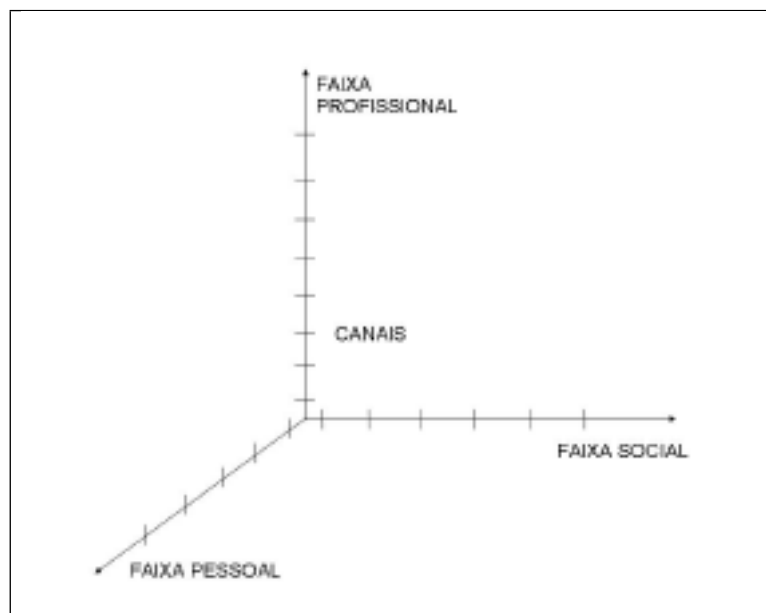


Figura 2.1: Faixas de interesses representadas em eixos cartesianos. Figura adaptada de [1]

Na faixa profissional, por exemplo, temos canais de interesses como: empregos, cursos, idiomas e concursos. Já na faixa social, temos canais de interesses como: festas, viagens e eventos musicais. Finalmente, na faixa pessoal, temos canais de interesses como: relacionamentos, hobbies e literatura.

Em uma analogia com modelos de comunicação em rádio comercial, podemos pensar nas faixas como as várias faixas de frequência disponíveis (AM, FM, OM) e as várias frequências específicas como os canais. Mas a REPI vai além deste modelo, pois permite que um usuário sintonize vários canais ao mesmo tempo, como se pudesse “ouvir” várias estações de rádios simultaneamente.

Seguindo com esta analogia, em uma REPI também existe um sintonizador, que na verdade é o critério que determina se as mensagens de um usuário devem ser entregues aos outros, de acordo com os interesses de ambos.

Mais detalhes sobre a REPI podem ser obtidos em [1, 31].

2.1 Exemplo de formação de uma REPI

Para exemplificar a formação de uma REPI, mostramos a seguir como é feita a entrega de uma mensagem em uma REPI com quatro nós, chamados de A, B, C e D. Nesta rede, os prefixos dos nós são formados por duas características, que são usadas para o roteamento das mensagens, e um interesse, que é usado para o endereçamento.

Tabela 2.1: Tabela que mostra as características e interesses dos participantes da REPI do exemplo apresentado.

Nome do nó	Prefixo		
	Características		Interesses
	Cor do cabelo	Faixa de Altura	
A	Castanho	1,50-1,60	Defesa Civil
B	Castanho	1,70-1,80	Esportes
C	Louro	1,50-1,60	Cinema
D	Preto	1,80-,190	Defesa Civil

A Tabela 2.1 apresenta os prefixos dos quatro nós da rede. Neste exemplo, o nó A, que tem interesse em Defesa Civil, precisa se comunicar com o nó D, que também possui este interesse.

Porém, como é possível notar na Figura 2.2, que apresenta os raios de transmissão dos nós da rede, D não está no alcance de comunicação do transceptor de A. Por isso, a rede precisa da Colaboração¹ dos outros dois nós, que não possuem interesse

¹Para melhor organizar o texto, a palavra colaboração será grafada com a primeira letra

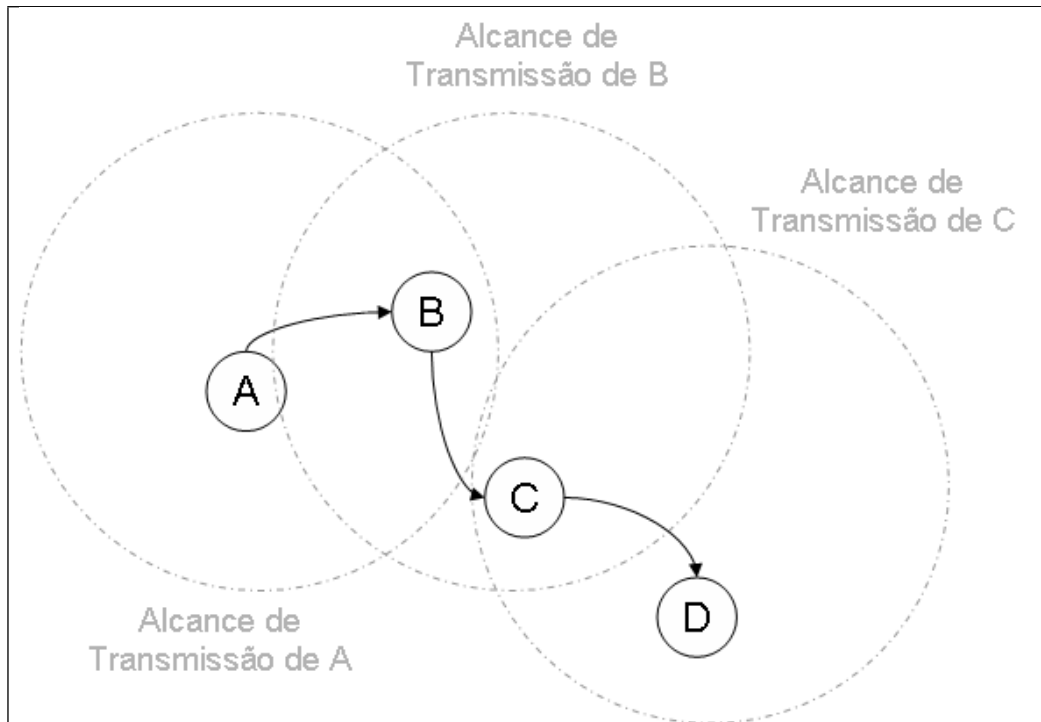


Figura 2.2: Raio de comunicação dos nós utilizados no exemplo de formação da REPI.

em defesa civil, para ser formada.

Neste exemplo, a comunicação de A para D segue cinco etapas:

1. O nó A inicia a comunicação e envia uma mensagem para a rede, buscando outros nós interessados em defesa civil;
2. O nó B é o único no alcance de comunicação de A e somente ele recebe a mensagem enviada. Como B não possui interesse em defesa civil, ele não aproveita a mensagem, mas verifica as características do seu prefixo para determinar se ela deve ser encaminhada;
3. Como o nó A e B possuem uma característica em comum, a cor do cabelo, o nó B encaminha a mensagem. Nesta situação, consideramos que o B foi um colaborador para a comunicação de A e D, pois encaminhou a mensagem mesmo não tendo interesse nela;
4. Por ser o único nó no raio de transmissão de B, o nó C recebe a mensagem e a encaminha também, pois possui o campo faixa de altura coincidente com o do nó origem, tornando-se um colaborador da rede também;
5. Finalmente, a mensagem é entregue ao nó D que a aproveita por ser um dos seus destinatários, em função de ter o mesmo interesse que o nó origem. Neste

maíuscula quando se referir ao conceito de colaboração em uma REPI

ponto a propagação da mensagem é interrompida, pois o nó D não possui nenhuma característica em comum com o nó A e por isso a mensagem não é encaminhada por ele.

Uma das limitações deste modelo é a possibilidade de ocorrência de isolamento de nós, em função da ausência de características em comum nos nós encaminhadores. No Capítulo 4 são realizados experimentos para verificar em quais configurações da REPI-A este problema é observado e qual o impacto dele no desempenho da REPI-A nas topologias avaliadas.

2.2 REPI-A

Em função das características da REPI, em especial a capacidade de formação de grupos de acordo com os interesses dos seus participantes e a utilização de características destes para a formação da rede, ela possui grande potencial para ser um arcabouço para a criação de aplicações para a formação de redes sociais.

Além disso, a junção destes conceitos com as tecnologias de comunicação sem fio pessoais torna possível a formação destas redes em ambientes reais, através da interação dos participantes com dispositivos de fácil acesso, como celulares e PDAs, para facilitar a interação social em situações do dia-a-dia.

Nesta seção, apresentamos a REPI-A, uma implementação da REPI em um ambiente de redes Ad Hoc.

A REPI-A foi concebida com os seguintes objetivos:

- Implementar o modelo de Comunicação Distribuído Endereçado Por Interesses proposto em [1], criando aplicações para avaliá-lo em uma rede sem fio totalmente distribuída. Estes experimentos podem ser repetidos de duas formas, ambas utilizando ferramentas desenvolvidas neste trabalho: por usuários reais, interagindo através de uma aplicação mensageira; ou de forma automatizada, sem interação humana, através de ferramentas de execução automatizada. As duas formas são descritas no Capítulo 3;
- Estudar o comportamento de uma implementação real de uma REPI e, a partir deste estudo, fornecer os dados necessários para a realização de simulações com milhares de nós, que serão realizadas em trabalhos futuros;
- Oferecer ferramentas para parametrizar a rede e avaliar o seu funcionamento com várias configurações diferentes, que são detalhadas na Seção E.1;
- A partir de experimentos reais, avaliar a aplicabilidade do modelo e propor aplicações que explorem suas características.

Para cumprir esses objetivos, a REPI-A integra elementos de hardware, software e comunicação, descritos a seguir:

Dispositivo de Comunicação Tmote Sky

Dispositivo de comunicação sem fio que utiliza o padrão 802.15.4/ZigBee, descrito em detalhes no Apêndice D, para comunicação através de rádio frequência. Mais detalhes sobre o Tmote Sky são apresentados no Apêndice G;

Middleware Tmote

Desenvolvido na linguagem NesC, um dialeto do C para programação de sistemas embarcados, este Middleware é o responsável pelo controle básico do hardware do Tmote Sky. Ele realiza a troca de mensagens entre a aplicação mensageira e o transceptor de rádio frequência. Ele é integrado ao sistema operacional para sistemas embarcados TinyOS 1.1, que é descrito no Apêndice F. Mais detalhes sobre Middleware Tmote são apresentados na Seção 2.3.

SAMCRA

O Sistema de Automação, Monitoração e Configuração de Redes Ad Hoc (SAMCRA) [29] foi desenvolvido para permitir a automatização dos processos de configuração, execução de experimentos e coleta de dados em uma rede Ad Hoc. Ele realiza essa tarefa através de três módulos de software que são descritos em detalhes no Apêndice E;

Aplicação Mensageira

Aplicação proposta como prova de conceito para a REPI-A. Ela é desenvolvida em Java 1.5 e permite que os participantes da REPI-A troquem mensagens instantaneamente em canais de interesses que podem ser pré-definidos ou criados pelo próprio participante, através da comunicação por rádio frequência;

Protocolo REPI-A

Integrado a aplicação mensageira, ele também é implementado em Java e executa nos nós distribuídos. Sua função é aplicar as políticas de roteamento, endereçamento e controle de duplicação de mensagens de acordo com as várias configurações possíveis da REPI-A.

A Figura 2.3 apresenta o diagrama de implantação da REPI-A, mostrando como se dá a interação dos elementos descritos anteriormente. Nela é possível notar que existem três canais de comunicação na rede, descritos a seguir:

Comunicação dos PCs da rede com SAMCRA

Esta comunicação, que é feita através do protocolo TCP/IP, é realizada para

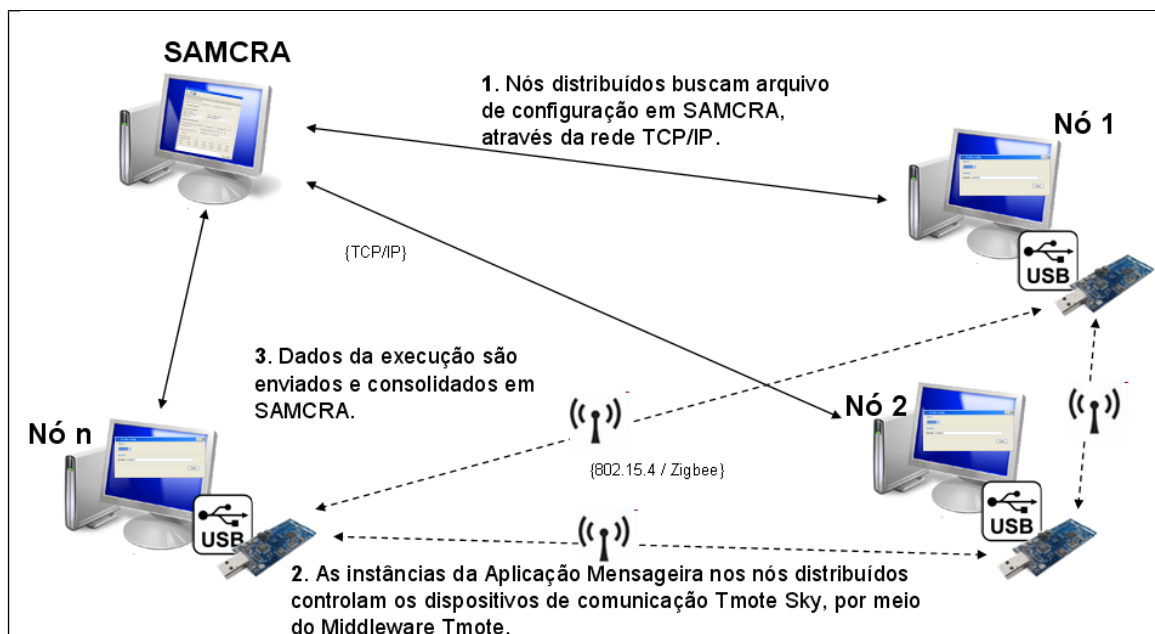


Figura 2.3: Diagrama de implantação da REPI-A

a comunicação dos nós remotos com SAMCRA, que controla a parametrização da rede e consolida os dados coletados nas suas execuções. Ela é necessária apenas na etapa de avaliação da REPI-A. Em um ambiente de produção, os dispositivos já serão entregues aos usuários finais com as devidas configurações e não será necessário realizar a coleta centralizada de dados na rede.

Para divulgação dos dados coletados pela aplicação, foi utilizada a tecnologia *Remote Method Invocation* (RMI) [32] da linguagem de programação Java [33]. Para a configuração da rede, foi usada a ferramenta JConfig [34], desenvolvida em Java e que oferece serviços de distribuição de arquivos de configuração escritos em XML [35];

Comunicação Middleware Tmote com Protocolo REPI-A

Realizada através da porta USB, esta comunicação faz com que as mensagens trocadas através do dispositivo Tmote Sky sejam sempre processadas pelo protocolo REPI-A. Ou seja, as mensagens geradas pelo protocolo REPI-A são enviadas, através da porta USB, para o middleware Tmote e este as repassa para os outros dispositivos da rede através de rádio frequência, utilizando o padrão 802.15.4/ZigBee. Ao receber uma nova mensagem, o middleware Tmote a encaminha pela porta USB para que ela seja avaliada pelo protocolo REPI-A, que decidirá se a mensagem deve ser reencaminhada ou descartada, de acordo com os critérios de encaminhamento estabelecidos;

Comunicação entre Tmotes

Realizada com base no padrão 802.15.4/ZigBee, é o principal canal de comu-

nicação para formação da REPI-A, através desta comunicação que a rede Ad Hoc é efetivamente formada.

2.2.1 Tipos de mensagens da REPI-A

A REPI-A está preparada para lidar com 3 tipos de mensagens, descritas a seguir:

Mensagem REPI-A

Mensagem principal da REPI-A, que carrega as características e interesses dos participantes da rede. Esta mensagem é composta pelos seguintes campos:

chatText

Mensagem de texto da aplicação mensageira, representada como um vetor de cinquenta e cinco caracteres ASCII;

chatID

Campo com um apelido que o usuário pode especificar para se identificar na aplicação mensageira, representado como um vetor de dez caracteres ASCII. Se este campo não for preenchido, o usuário será tratado como um usuário anônimo;

CustomInt

Campo com o interesse customizado do usuário, representado como um vetor de oito caracteres ASCII;

source

Campo com o endereço TinyOS de origem do tmote que enviou a mensagem, representado como um inteiro de dezesseis bits. É importante notar que o endereço do Tmote não é usado pelo protocolo REPI-A e é usado apenas para facilitar a contabilidade dos dados coletados;

lastHop

Campo com o endereço TinyOS do último Tmote por onde o pacote passou antes de chegar ao destino, representado como um inteiro de dezesseis bits. Este campo também só é usado para facilitar a contabilidade dos dados da execução dos experimentos;

HTL

Indica quantos Tmotes podem encaminhar a mensagem antes que ela seja considerada como expirada. A sigla HTL deriva da expressão *Hops to Live*, que indica quantos “saltos” o pacote ainda pode executar antes de ser descartado. Este campo é representado como um inteiro de oito bits;

seqno

Número sequencial, que é usado como um identificador único do pacote;

prefixo

Conjunto de características e interesses de um participante da rede. É representado como um vetor de trinta e dois valores booleanos.

Mensagem de Status do Tmote

Mensagem usada para tarefas de comunicação entre o protocolo REPI-A e o middleware Tmote. Seus objetivos principais são a comunicação de erros e a identificação do Tmote para o protocolo REPI-A. É composta dos seguintes campos:

errCode

Campo que identifica o código do erro ocorrido no middleware Tmote. É representado como um inteiro de oito bits;

sourceaddr

Campo que identifica o endereço TinyOS do Tmote de origem do pacote. É representado como um inteiro de dezesseis bits;

isErr

Campo que identifica se o pacote atual está notificando um erro. É representado como um booleano.

Mensagem de contagem de tempo

Este é o pacote usado pelo Módulo de Contagem de Tempo, que conta o tempo de entrega de pacotes na rede e é descrito na Seção B.12. Esta mensagem possui os seguintes campos em comum com a mensagem da REPI: source, lastHop, HTL, seqno. Além desses campos, ela possui os seguintes:

timeLeft

Usado para marcar o instante em que o pacote deixou o Tmote original. Representado como um inteiro de trinta e dois bits;

timeArrive

Usado para marcar o instante em que o pacote retornou ao Tmote original. Representado como um inteiro de trinta e dois bits;

hops

Contador de saltos do pacote, indica por quantos Tmotes a mensagem passou antes de chegar ao destinatário. Representado como um inteiro de oito bits;

2.3 Middleware Tmote

O principal dispositivo de comunicação dentro da REPI-A são os Tmotes Sky, apresentados no Apêndice G. Através deles, as mensagens da REPI-A são trocadas na rede Ad Hoc, utilizando Rádio Frequência (RF) em uma implementação do padrão 802.15.4/ZigBee.

Embora os Tmotes sejam os principais elementos comunicadores, o middleware que os controla é extremamente simples e realiza basicamente a troca de mensagens entre a porta USB e o transmissor RF, sem controlar nenhum detalhe do protocolo REPI-A. Essa escolha de projeto foi influenciada pelas severas limitações de recursos computacionais que este tipo de dispositivo impõe.

O Middleware Tmote tem as seguintes funções:

- Enviar e receber mensagens da REPI-A através do protocolo Sensornet Protocol, proposto em [36]. Nesta aplicação só são usadas as capacidades de envio e recebimento de pacotes deste protocolo, visto que não é feito nenhum gerenciamento dos nós vizinhos. O Sensornet Protocol e o chip CC2420 do Tmote Sky são os responsáveis pelo controle das camadas 1 e 2 (Física e Enlace), segundo o modelo OSI da ISO. Ao middleware Tmote cabe o controle da camada de transporte;
- Identificar quais mensagens devem ser transmitidas através da porta USB e quais devem ser transmitidas através de RF. Todas as mensagens transmitidas por RF são enviadas por *broadcast*;
- Gerenciar uma fila circular com as mensagens que devem ser entregues para evitar a perda de mensagens em situações de tráfego intenso;



Figura 2.4: Detalhe dos Leds utilizados para depuração do middleware Tmote

- Controlar o funcionamento do Tmote, verificando sua operacionalidade. Em caso de falha do dispositivo, a aplicação o reinicia através das funções de temporizador cão de guarda (*WatchDog Timer*) do microcontrolador MSP430.

Neste método, a cada novecentos milissegundos é iniciado um temporizador, chamado de cão de guarda, que é programado para reiniciar o dispositivo depois de um segundo. Quando o Tmote trava, o contador não é reconfigurado e ele é reiniciado automaticamente;

Tabela 2.2: Tabela de códigos dos Leds de Depuração dos Tmotes

Código	Tipo de Mensagem	Etapa da execução
000	SUCESSO	Recebimento de mensagem da REPI-A
001	ERRO	Erro ao obter posição fila de envio
010	ERRO	Erro ao enviar mensagem
011	SUCESSO	Primeira etapa de envio de mensagem concluída
100	SUCESSO	Confirmação do envio da mensagem
101	ERRO	Envio adiado porque outra mensagem já está sendo enviada
110	ERRO	Função de envio de mensagem foi chamada, mas fila de envio está vazia
111	ERRO	Erro na alocação de espaço na fila de envio ao enviar mensagem de Status

- Exibir detalhes da execução através dos Leds de depuração, apresentados na Figura 2.4. Para facilitar esta tarefa, foi criada a tabela 2.2, que mapeia os principais estados da sua execução para um número binário de três dígitos, apresentado na primeira coluna da tabela. Para representar, por exemplo, o código 100, o Tmote acenderá o Led 0 e deixará o Led 1 e o Led 2 apagados.
- Reportar para o protocolo REPI-A a ocorrência de erros ocorridos durante a execução do middleware Tmote. Para este fim, também foi criada a tabela 2.3, que mapeia os 7 principais tipos de erros que podem ocorrer na execução do middleware. Esses dados são enviados para o protocolo REPI-A e consolidados em SAMCRA.

Tabela 2.3: Tabela com os códigos de erros do middleware Tmote

Número do Erro	Significado
0	Fila de envio sem posições.
1	Erro ao enviar uma mensagem.
2	Tentativa de envio de mensagem enquanto o rádio está sendo utilizado em outro envio.
3	Envio foi solicitado, mas fila de mensagens está vazia.
4	Fila está lotada e há um envio sendo realizado no momento.
5	Erro na fase de conclusão de envio de mensagem.
6	Erro ao receber uma mensagem na rede.

2.4 Protocolo REPI-A

O Protocolo REPI-A é implementado em Java e é integrado a aplicação mensageira. Esta integração acontece porque, como mostra a Figura 2.5, eles interagem diretamente na arquitetura proposta. Porém, em função da orientação a objetos oferecida pela linguagem Java, eles podem ser desassociados e utilizados com outras aplicações e protocolos.

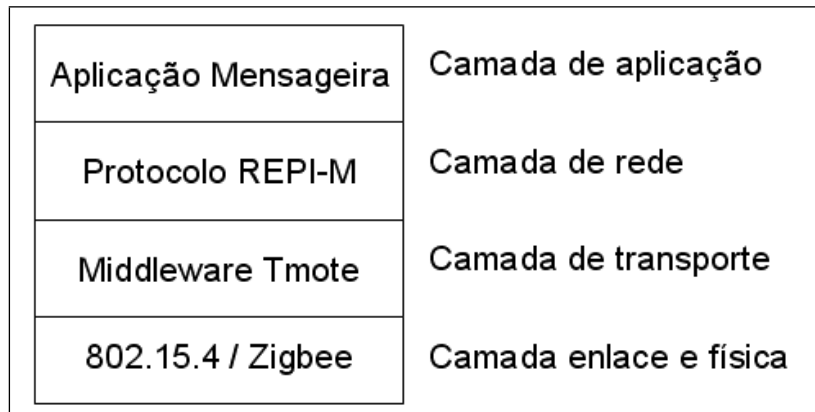


Figura 2.5: Representação da pilha de protocolos utilizados para a formação da REPI-A.

Quando uma mensagem é recebida por um dos Tmotes, ela inicialmente é tratada de acordo com as especificações do padrão 802.15.4 pelo próprio Tmote, que implementa as camadas físicas e de enlace em hardware, com descrito no Apêndice G.

Em seguida, a mensagem é encaminhado para o middleware Tmote que é o responsável pela sua transmissão até o Protocolo REPI-A, implementado como uma classe Java. Essa comunicação é feita através da porta USB onde o Tmote está conectado.

Ao receber a mensagem, o protocolo REPI-A aplica as políticas vigentes na rede para determinar se a mensagem será encaminhada ou descartada, de acordo com a forma como a REPI-A foi parametrizada.

Na implementação atual, os seguintes parâmetros do Protocolo REPI-A podem ser variados:

Número de Saltos para entrega de mensagens

Número de saltos (HTL - *Hops-To-Live*) que a mensagem pode percorrer antes de ser descartada;

Número de características no prefixo

Número de características que serão usadas no prefixo das mensagens para realizar o encaminhamento das mensagens. Este valor pode ser variado de uma até oito características;

Política de encaminhamento

Determina qual será a política de encaminhamento de mensagens na REPI-A. Possui duas opções: encaminhamento se pelo menos uma característica for coincidente ou encaminhamento apenas se todas as características forem coincidentes;

Controle de propagação de mensagens

Este parâmetro define qual será o mecanismos de controle de propagação de mensagens duplicadas usado pelo protocolo. Ele pode ter três valores distintos:

Sem Memória

Com esta configuração os Tmotes não possuem memória dos pacotes que já encaminharam e podem encaminhar a mesma mensagem várias vezes;

Número sequencial

Com esta configuração o controle de propagação de mensagens é feito através de um número sequencial, estratégia utilizada em outros tipos de protocolos. Neste algoritmo, todas as mensagens são marcados com um número sequencial que é incrementado de uma unidade a cada nova mensagem. Ao receber uma mensagem, o protocolo REPI-A detecta qual foi o maior número sequencial de uma mensagem encaminhada e, caso o número sequencial da mensagem atual seja maior que ele, a mensagem é encaminhada e este valor é armazenado como o novo número sequencial de referência para futuras comparações. Se o número sequencial for menor, a mensagem é descartada;

Hash

Com esta configuração o controle de propagação de mensagens é feito através de uma tabela de dispersão (*hash*) que registra todas as mensagens entregues. Ao receber uma nova mensagem, o protocolo REPI-A verifica se ela já foi entregue para decidir se ela é encaminhada ou não. As entradas da tabela de dispersão são criadas a partir da concatenação de informações do prefixo do usuário. A tabela de dispersão é implementada com as funções da classe *HashSet* do Java;

Esses valores são configurados através do Sistema SAMCRA, que possui uma interface para a parametrização de todos os nós da rede de forma centralizada, conforme é descrito na Seção E.2.

2.4.1 Implementação do Prefixo C

O prefixo C foi implementado na REPI-A com um número configurável de características. Este número pode variar de um até oito, de acordo com a parametrização

da REPI-A.

O prefixo C foi implementado com um vetor de bits em que cada campo possui um determinado número de bits, que indicavam quantos valores distintos aquele campo poderia assumir. A tabela 2.4 apresenta o nome das características utilizadas, o número de bits usados para representá-las e o número de valores distintos que elas podem assumir.

Tabela 2.4: Tabela que apresenta como as características do Prefixo C foram modeladas na REPI-A.

Nome Campo	Número de Bits	Valores distintos
Gênero	2	4
Faixa Etária	3	8
Cor do Cabelo	3	8
Faixa de Altura	3	8
Cor dos Olhos	2	4
Etnia	2	4
Faixa de Peso	3	8
Porte Físico	3	8

Essas características foram escolhidas por serem comuns a todos os seres humanos e por seguirem, preferencialmente, a distribuição normal para grandes populações. Porém, vale ressaltar que esse conjunto de características pode ser alterado e adaptado para outros tipos de contextos.

Além disso, é importante que os tipos de características escolhidas sejam correlacionados aos interesses, de acordo com o contexto. O objetivo desta abordagem é aumentar a probabilidade de encaminhamento de mensagens dentro dos diversos grupos de interesse da rede.

Neste trabalho, as características foram preenchidas de forma aleatória, através da geração de numéricos randômicos usando a distribuição normal. Estes números eram gerados como números reais e depois eram convertidos em números discretos, através de faixas de valores. A quantidade dessas faixas foi determinada pelo número de valores possíveis de cada característica.

Mais detalhes sobre a implementação das características são apresentados no Apêndice B.

2.5 Comparação da REPI-A com outros modelos

A proposta da REPI-A permite que ela seja usada como arcabouço para a construção de aplicações que mesclam características de aplicações de comunicação instantânea como Msn Messenger [37], Icq [38] e Yahoo Messenger [39], e aplicações de redes sociais, como Orkut[8], Facebook[9], Octopop[30].

Um exemplo de aplicação desenvolvida para a REPI-A que se assemelha as aplicações de troca instantânea de mensagens é a aplicação mensageira proposta neste trabalho e apresentada no Capítulo 3. Nesta aplicação, o usuário pode enviar e receber mensagens em uma rede de forma instantânea, se comunicando com um ou mais participantes.

Porém, ao contrário das aplicações tradicionais, onde o usuário precisa ter uma participante da rede específico para se comunicar, em uma REPI-A o interlocutor é encontrado e determinado pelos seus interesses e características. Este modelo também está presente nas redes sociais, onde os usuários criam comunidades com interesses comuns e podem se relacionar através de fóruns, inclusive de forma anônima.

A seguir, apresentamos algumas propriedades da REPI-A que a diferenciam destes dois modelos e que criam a possibilidade para a criação de novas aplicações que explorem estas propriedades:

Formação de grupos de longo alcance

O padrão 802.15.4/ZigBee oferece a capacidade de comunicação através de múltiplos saltos, permitindo que a rede possa ter comunicação entre usuários que não estão no alcance do rádio de comunicação um do outro, através do encaminhamento de mensagens promovido por outros dispositivos da rede. Com isso, a rede pode se expandir até um tamanho X , onde $X = \text{MAIOR}(n, \text{HTL})$, onde n é o número de nós da rede e HTL é o número de nós que uma mensagem pode percorrer antes de ser descartada;

Ausência de servidores

Com a utilização de redes móveis Ad Hoc, a aplicação funciona sem a utilização de servidores centralizados. Com isso, usuários podem formar redes e se comunicar com pessoas ao seu redor, se valendo da sua localização para interação social de acordo com seus interesses. Esta abordagem cria a possibilidade do surgimento de uma nova classe de aplicações, conforme é mostrado na Seção 3.3;

Ausência de mensagens de controle

Uma REPI-A só é formada enquanto existem mensagens circulando, pois não há mensagens de controle (*beacons*) para controlar a entrada e saída de nós ou a formação de grupos. Com isso, na prática, uma REPI-A apresenta uma topologia diferente para cada pacote entregue na rede;

Múltiplas visões da rede

Esta característica é uma consequência da anterior e é causada pela ausência de tabelas de roteamento distribuídas para a formação da rede. Com isso, a rede pode ser totalmente distinta para cada um dos seus participantes. A

Figura 2.6 ilustra um exemplo desta situação. Neste cenário, existem 5 nós na rede: A, B, C, D e E; e existem dois tipos de prefixo: X e Y. Ao enviar uma mensagem, o nó D só possui o nó E ao seu alcance e, como eles possuem prefixo comum, este encaminha a mensagem recebida. Já os nós B e C possuem prefixos diferente do nó E e por isso descartam a mensagem recebida, assim como faz o nó D ao receber a mensagem de volta e detectar ela foi gerada por ele mesmo. Ao final desta troca de mensagens, os nós D e E só terão trocado mensagens entre si e, para eles, a rede é formada apenas por 2 nós. Já os nós B e C terão a visão de uma rede com 3 nós: eles mesmos, o nó E e o nó D. Enquanto que, para o nó A, a rede não foi formada;

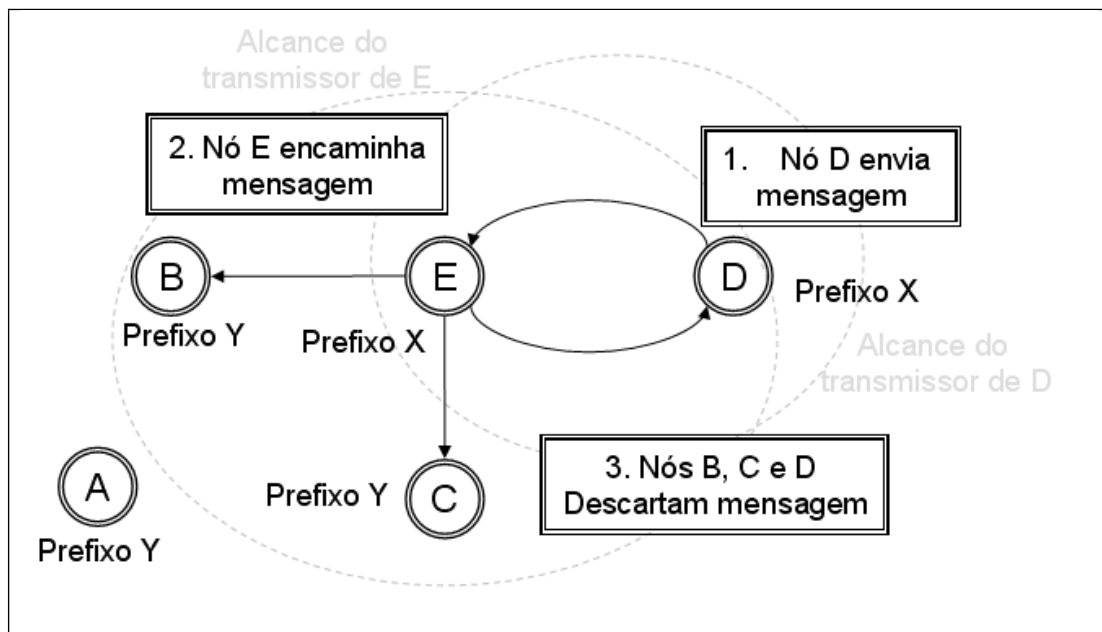


Figura 2.6: Exemplo da formação de uma REPI.

Participação colaborativa

A proposta da REPI permite que usuários participem da rede de forma colaborativa, encaminhando pacotes que possuem interesses diferentes do seu, desde que suas características atendam os critérios de encaminhamento de mensagens da rede. Com isso, o nó colabora com a rede e permite que as mensagens possam alcançar participantes que tenham interesses diferentes de todos os seus vizinhos.

A Colaboração é uma característica presente na maioria das redes P2P. Neste tipo de ambiente, é comum que enquanto um usuário obtém um arquivo de outros participantes, ele também atue como servidor, fornecendo os pedaços que já obteve para outros usuários. Nesta situação, ele colabora com a formação da rede, mas também se beneficia dela. Porém, é comum que mesmo depois de

obter o arquivo completo, ele continue disponibilizando-o para outros usuários, apenas para colaborar com a formação da rede.

2.6 Discussão

A utilização de interesses para a formação de redes par-a-par (P2P - *peer-to-peer*) é proposta em outros trabalhos da literatura.

Em [40] os autores propõe a criação de um protocolo para construir e manter redes sobrepostas baseadas em interesses para facilitar o compartilhamento de arquivos.

Nesta estratégia, os participantes da rede declaram seus interesses e aqueles que possuem interesses em comum são agrupados. O objetivo desta abordagem é diminuir o número de consultas necessárias para a localização de um arquivo, visto que usuários com interesses semelhantes costumam buscar arquivos semelhantes.

Ainda neste trabalho, o protocolo proposto é comparado com outros existentes na literatura e são obtidos resultados satisfatórios, que mostram que o protocolo é capaz de resistir a falhas e mudanças bruscas na topologia da rede. Porém, em função da busca dos interesses dos usuários e formação dos grupos, o tempo de convergência da rede é maior usando esta abordagem que nos outros protocolos analisados.

O que faz com que este tempo seja grande é que os grupos nesta proposta são fortemente acoplados, visto que os nós precisam conhecer todos os interesses na rede para saber em quais grupos eles devem se associar. Como os grupos da REPI-A são fracamente acoplados e a propagação das mensagens é desassociada dos interesses, através do uso das características, este problema não é observado na REPI-A.

Outro trabalho que propõe o uso de interesses em redes P2P é apresentado em [41]. Nele, os autores propõem a criação de um mecanismo para viabilizar a formação de comunidades com interesses em comum para promover o compartilhamento de artigos acadêmicos. O objetivo da proposta é criar comunidades onde os participantes possam sugerir a leitura e debater a respeito de artigos acadêmicos.

Para isso, os autores propõe o uso de agentes para criar um sistema de confiança nos usuários e comunidades da rede, de forma que os artigos sejam compartilhados e classificados de forma confiável na comunidade. No trabalho são feitas simulações que mostram que a formação das comunidades de interesse traz ganhos na busca de arquivo em relação a sistemas que usam agentes sem a formação das comunidades.

2.6.1 Colaboração

O conceito de Colaboração está presente em outros tipos de redes P2P. Porém, em função das diferenças de arquitetura e objetivos das redes, elas são usadas de formas

distintas das vistas na REPI-A.

Uma das formas típicas de Colaboração é a que existe em redes de compartilhamento de arquivos. Ela acontece quando um usuário já possui um determinado arquivo e o disponibiliza, fornecendo seus recursos de computação e de rede para que outros usuários possam obtê-lo. Em geral, o participante faz isso sem ter garantias de retorno, visto que isso não obriga que os outros participantes também compartilhem seus arquivos.

Neste contexto, uma ideia natural é incentivar a Colaboração, para aumentar o desempenho da rede. Pois, como a maioria dos participantes dispõe de poucos recursos de largura de banda e processamento, além de não haver garantia de confiabilidade dos nós, a rede se viabiliza através da capacidade combinada deles. Por isso, um comportamento egoísta dos seus participantes pode prejudicar o desempenho global da rede.

Um exemplo de incentivo oferecido em redes P2P é observado no aplicativo BitComet [42], cliente de redes Bit Torrent. Nele, os participantes da rede são cadastrados e é criado um sistema de recompensa que os pontuam e classificam de acordo com a forma que eles colaboram com a rede.

Os pontos são obtidos de duas formas: quando o participante permanece ativo na rede por períodos maiores que uma hora e quando ele disponibiliza arquivos para compartilhamento. O número de pontos obtidos com o compartilhamento varia de acordo a quantidade de megabytes compartilhada por ele.

De acordo com sua pontuação, os participantes são classificados em quarenta e oito categorias e, de acordo com a sua categoria, eles podem obter mais recursos para aumentar suas taxas de *download* e obter mais desempenho na rede.

Outras aplicações cliente de redes de compartilhamento de arquivos também apresentam sistemas deste tipo. O eMule [15], por exemplo, possui um sistema mais simples, em que a quantidade de largura de banda oferecida pelo participante para *upload* influencia diretamente na largura de banda que ele terá disponível para *download*.

Em REPI-A a Colaboração funciona de forma similar: os nós que não possuem interesse nas mensagens fornecem seu processamento, a energia de suas baterias e sua capacidade de transmissão para ajudar na formação da rede.

Como a REPI-A depende da propagação da mensagem por múltiplos saltos, a Colaboração na rede passa a ser fundamental em alguns contextos, de acordo com o nível de popularidade dos interesses contidos nas mensagens. Pois, quanto mais impopular é um interesse, mais ele depende da Colaboração de outros nós para a sua propagação.

No Capítulo 4 são apresentados alguns resultados que estudam o impacto da Colaboração em uma REPI-A. A criação de mecanismos de incentivo à Colaboração

para a REPI-A está fora do contexto deste trabalho.

Capítulo 3

Uso da REPI-A

Para avaliar a proposta da REPI-A em uma rede Ad Hoc com vinte dispositivos Tmote Sky, desenvolvemos uma aplicação mensageira para interagir com o protocolo REPI-A e o middleware Tmote.

Um dos objetivos desta aplicação é permitir que a rede seja usada de duas formas: uma é o uso tradicional, através de participantes reais, interagindo em uma interface semelhante as de aplicações de trocas de mensagens instantâneas, como o Msn Messenger [37] e o Icq [38]; e outra através de execuções automatizadas, controladas por SAMCRA e *scripts* de execução remota de processos. A primeira delas é mostrada na Seção 3.1 e a outra na Seção 3.2.

3.1 Utilização da REPI-A por usuários reais

A tela principal da aplicação mensageira é apresentada na Figura 3.1 e é através dela que os usuários trocam as mensagens que formam a REPI-A.

Nesta tela, o participante da rede pode realizar as seguintes ações: enviar e receber mensagens na REPI-A; manifestar seus interesses na REPI-A selecionando os canais em que deseja receber mensagens; propagar seus interesses na rede, através da criação de canais customizados; e criar canais privativos de comunicação protegidos por senha, através do canal reservado.

Para enviar uma mensagem, o usuário da aplicação deve selecionar uma aba que representam os canais de interesse disponíveis na rede. Ao selecionar uma aba, ele envia mensagens para aquele canal e pode visualizar as mensagens enviadas por outros participantes da REPI-A. No Figura 3.1, por exemplo, a aba selecionada é a do canal de pesquisa e qualquer mensagem enviada pelo usuário nesta situação será direcionada para este canal.

Quando o usuário envia uma mensagem, o primeiro passo realizado pela aplicação mensageira é montar a estrutura da mensagem. Para isso, ela une o prefixo do

usuário, com suas características e o interesse selecionado, com os dados da aplicação, que são a mensagem que foi inserida na interface e o apelido, que é opcional.

Depois de montada, a mensagem é enviada através do Tmote conectado na aplicação e é transmitida para a REPI-A. Ao chegar em outro Tmote, a mensagem é tratada pelo middleware Tmote, depois pelo protocolo REPI-A e, finalmente, é repassada para ser tratada pela aplicação mensageira.

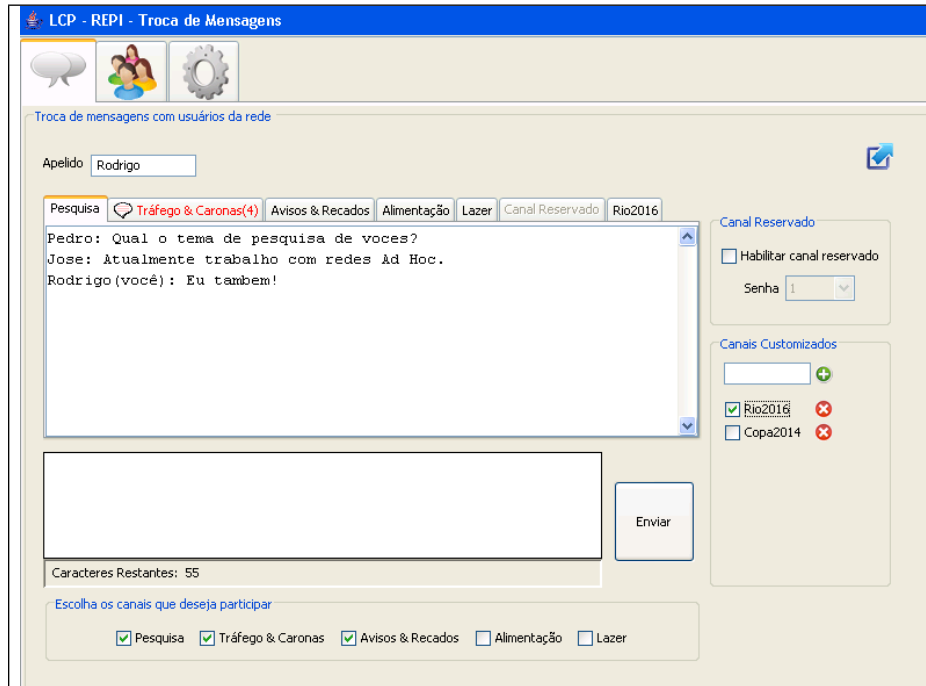


Figura 3.1: Tela de troca de mensagens da aplicação mensageira.

Na implementação atual da REPI-A, são previsto três tipos de canais, que são descritos a seguir:

Canais públicos

Canais de comunicação com interesses pré-definidos, que são oferecidos pela aplicação para todos os usuários da rede. Na implementação atual existem 5 canais públicos que podem ser customizados através de SAMCRA. O objetivo destes canais é criar canais com interesses populares para que os participantes comecem a interagir na rede. No exemplo da Figura 3.1, são apresentados os canais pré-definidos usados em execuções da aplicação mensageira no Laboratório de Computação Paralela da UFRJ, que foram os seguintes: Pesquisa, Tráfego & Caronas, Avisos & Recados, Alimentação e Lazer. Esses interesses foram escolhidos por estarem relacionados a assuntos comuns do dia-a-dia da maioria das pessoas do laboratório;

Canal Reservado

Canal de comunicação privativo, protegido com senha. Neste canal, todas as

mensagens enviadas são protegidas por uma senha e apenas os usuários que estiverem com a mesma senha poderão ler as mensagens enviadas. O objetivo dele é avaliar a aplicação de mecanismos de privacidade na REPI-A;

Canais customizados

Além de utilizar os canais pré-definidos pela aplicação, o participante pode criar os seus próprios canais, com interesses de sua predileção. O objetivo destes canais é flexibilizar o uso da REPI-A e permitir que qualquer interesse possa ser usado na rede.

A tela de troca de mensagens é organizada em 4 painéis distintos, apresentados nas Figuras 3.2, 3.3 e 3.4 e descritos a seguir:



Figura 3.2: Abas de seleção de canais.

Abas de seleção de canais

Neste painel, apresentado na Figura 3.2, o participante tem acesso aos canais de comunicação disponíveis na rede. Para enviar uma mensagem, basta selecionar a aba com o nome do canal desejado, digitar o texto e enviar a mensagem através do botão Enviar.

Em sua configuração básica, o painel de seleção de canais apresenta seis abas: cinco abas para os canais públicos e uma aba para o canal reservado. Contudo, ao longo do uso da aplicação, são adicionadas novas abas para cada novo canal customizado selecionado pelo participante.

Ao criar um novo canal, uma aba é criada automaticamente, e o mesmo acontece ao selecionar um canal customizado criado por outro participante. Os

detalhes destes procedimentos são apresentados a seguir, no item que aborda o painel de Canais Customizados.

Além disso, as abas também apresentam as mensagens enviadas pelos outros participantes, de acordo com os canais. Ao receber uma nova mensagem em um canal que não está selecionado, a aplicação adiciona um ícone especial na respectiva aba e muda a cor da fonte da letra para vermelho, que alerta o participante sobre a chegada da nova mensagem e indica a quantidade de mensagens não lidas. No exemplo da Figura 3.2, o canal Tráfego & Caronas contém quatro mensagens não lidas;



Figura 3.3: Painel de escolha de canal público.

Escolha de Canais públicos Neste painel, apresentado na Figura 3.3, o participante pode escolher em quais canais públicos ele deseja receber mensagens. Para isso, basta marcar o canal desejado e todas as mensagens recebidas nele serão inseridas na respectiva aba.

Como é possível notar, o envio e recebimento de mensagem são desassociados, visto que um é determinado pela aba usada pelo participante para enviar as mensagens, e o outro pelos canais que ele selecionou. Com isso, um participante pode participar de um canal apenas como receptor ou apenas como gerador de mensagens;

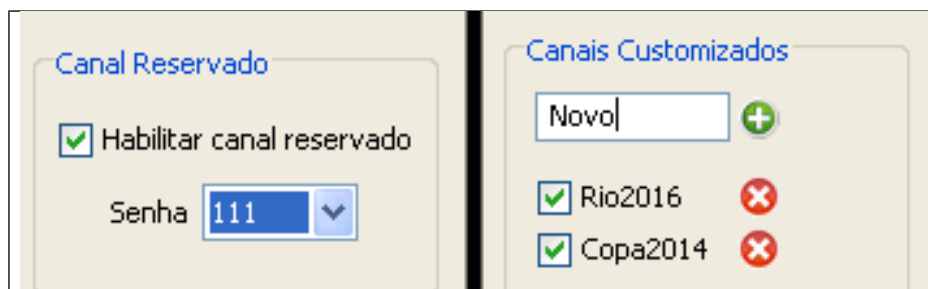


Figura 3.4: Canais customizado e reservado.

Canais Customizados

Este painel, mostrado do lado direito da Figura 3.4, é usado para que o usuário participe de canais customizados, permitindo que qualquer tipo de interesse seja utilizado na rede.

Para criar um novo canal, o usuário deve inserir um nome para ele, respeitando o limite de oito caracteres, e depois clicar no botão verde ao lado da caixa de texto. No exemplo da Figura, um canal chamado de “Novo” estão sendo criado.

A divulgação dos canais customizados é feita em dois momentos: ao criar um novo canal, a aplicação o divulga automaticamente para seus vizinhos imediatos; e o outro é quando uma mensagem é enviada em um canal customizado, pois todos os nós que a recebem são notificados da sua existência, o que permite eles comecem a utilizá-lo.

Na Figura 3.4 é apresentado o painel de Canais Customizados com dois canais selecionados pelo usuário da aplicação: Rio2016 e Copa2014.

Mais detalhes sobre a utilização dos canais podem ser obtidas no manual da aplicação mensageira, disponível no Apêndice B.

3.1.1 Escolha das características do participante da REPI-A

Na implementação atual da aplicação mensageira, o usuário da aplicação pode escolher as suas características de duas formas: através de uma tela onde ele seleciona valores pré-definidos em listas apresentadas pela aplicação ou de forma automática, em que os valores são gerados randomicamente.

Para escolher a primeira opção, um parâmetro, chamado de *chooseProfile*, tem que ser especificado na inicialização da classe Java que implementa a aplicação mensageira.

Quando isso ocorre, o usuário tem acesso a tela mostrada na Figura 3.5. Nela, ele deve ser preencher os vários campos com as suas características. No exemplo da figura estão alguns exemplos de características de uma pessoa, que seguem preferencialmente a distribuição normal e que são usadas em aplicações de redes sociais como o Orkut e o Facebook.

É importante ressaltar que a definição das características do participante está fora do escopo deste trabalho. As características apresentadas na tela são apenas exemplos que poderiam ser usados em uma implementação real. Em todos os experimentos realizados neste trabalho os valores dos campos foram escolhidos de forma randômica seguindo a distribuição normal.

Caso o usuário da aplicação queira que suas características sejam preenchidas de forma randômica, basta não especificar nenhum parâmetro na linha de comando. Além disso, é possível determinar o número de características que são usadas no prefixo, através do parâmetro de linha de comando *gaussian=x*, onde *x* é um valor inteiro entre zero e oito.

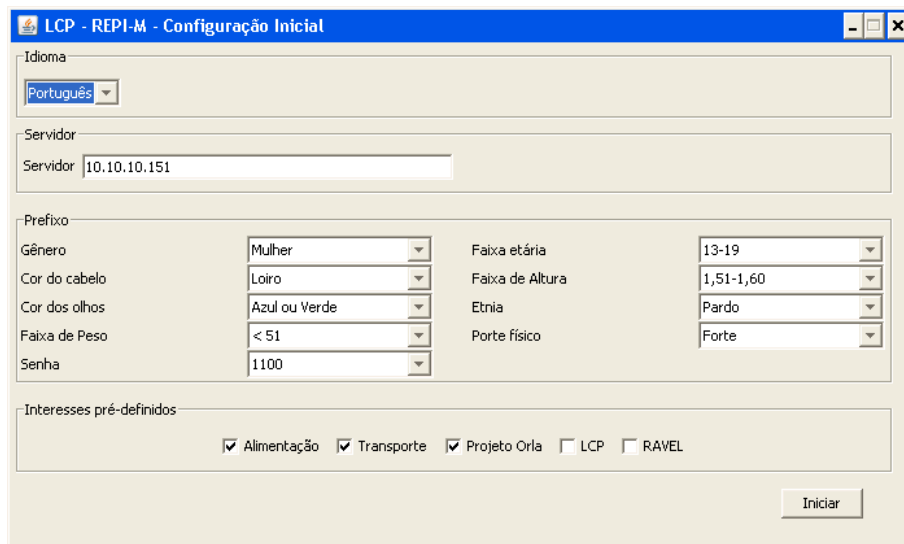


Figura 3.5: Tela de Configuração Inicial da aplicação.

Mais detalhes sobre os parâmetros de inicialização da aplicação mensageira e seu funcionamento são mostrados no Apêndice B.

3.2 Execuções Automatizadas

Neste tela, apresentada na Figura 3.6, é possível configurar a aplicação mensageira para ela enviar uma determinada quantidade de pacotes para a REPI-A sem intervenção do usuário. Este recurso foi usado para a realização dos experimentos apresentados neste trabalho.

Este tipo de execução pode ser feito automaticamente, através do parâmetro de linha de comando *auto*, durante a inicialização mensageira.



Figura 3.6: Tela de ativação da execução automática, que faz a aplicação enviar mensagens na rede.

No painel de execução automatizada, o usuário pode configurar dois parâmetros

da execução: o número de mensagens enviadas, entre um e noventa mil; e o intervalo entre o envio de duas mensagens consecutivas na rede, que pode ser configurada para um intervalo entre 0,5 e 600 segundos.

Com isso, a próxima mensagem é enviada na rede depois de um tempo de espera de um valor randômico dentro do intervalo especificado, para evitar que exista sincronização de envio em experimentos com vários nós enviando mensagens na rede.

Este recurso foi utilizado em conjunto com o Módulo de Execução Automatizada de SAMCRA para realizar os experimentos apresentados neste trabalho, conforme é descrito na próxima subseção.

3.2.1 Módulo de Execução Automatizada de SAMCRA

O Módulo de Execução Automatizada é um dos três módulos do sistema SAMCRA, apresentado em detalhes no Apêndice E. Neste trabalho, ele é usado em conjunto com o recurso de execução automatizada da aplicação mensageira para realizar a avaliação experimental apresentada no Capítulo 4.

Uma das principais funcionalidades deste módulo é permitir que um mesmo experimento seja repetido várias vezes com a mesma configuração da rede, de forma a gerar dados com maior confiabilidade estatística.

Além de repetir os experimentos, SAMCRA salva todos os dados em planilhas eletrônicas e, ao final de todas as execuções, calcula a média e desvio padrão de forma automática.

Para utilizar esta funcionalidade, o usuário deve configurar alguns parâmetros na interface de SAMCRA. Os principais deles são mostrados a seguir:

- Número de Tmotes, que determina quantos dispositivos serão usados no experimento;
- Número de Execuções, que determina quantas vezes o experimento será repetido;
- Número de Mensagens que serão enviadas ;
- Espera Tmotes e Servidor, que determinam quanto tempo cada Tmote e o Servidor irão esperar para iniciar uma nova execução;
- Intervalo entre mensagens, que determina o intervalo mínimo e máximo, em segundos, para o envio de duas mensagens consecutivas na rede. O tempo de envio será um número randômico entre o mínimo e o máximo especificado;
- Nome Script, que determina qual será o *script* de execução remota de processos que iniciará as aplicações mensageiras utilizadas no experimento;

- Tmotes com início automático, que determina quais serão os Tmotes com início automático, ou seja, quais Tmote enviarão mensagens durante a execução. Os Tmotes sem início automático podem atuar na rede apenas como receptores e repetidores de mensagens de outros Tmotes.

Esse parâmetros são configurados na interface de SAMCRA, como mostra a Figura 3.7.

Figura 3.7: Tela com as opções de parametrização de SAMCRA para a realização de uma execução automatizada.

3.3 Desenvolvimento de outras aplicações mensageiras para REPI-A

A REPI-A cria a possibilidade de desenvolvimento de diversos tipos de aplicações, como a aplicação mensageira descrita neste capítulo. Estas aplicações podem ser desenvolvidas para vários contextos, mas existem algumas propriedades comuns que são importantes para o sucesso da adaptação do modelo. As principais são:

Proximidade geográfica dos usuários

Por utilizar dispositivos de comunicação sem fio de baixo alcance (por volta de 100 metros), os usuários da rede devem estar próximos geograficamente. Algumas situações típicas onde há grande aglomeração de pessoas e que tem potencial para a utilização do modelo são: congressos, eventos esportivos e culturais, engarrafamentos, transportes públicos, confraternizações, centros de compras, entre outros;

Anonimato dos usuários

Uma das principais características de uma REPI-A é a ausência de identificação dos seus participantes. Por isso, as aplicações da REPI-A podem ser

modeladas para as situações onde os usuários pretendem trocar informações e a sua identificação prévia não é necessária ou até mesmo não é desejada;

Formação de grupos de acordo com interesses

Outra característica da REPI-A é a formação de grupos de usuários de acordo com os seus interesses. Portanto, as aplicações também devem envolver situações onde os usuários precisam formar grupos de interesse para trocar de informações ou outras ações previstas pela aplicação.

Na próxima subseção apresentamos alguns exemplos de aplicações que podem explorar o modelo proposto.

3.3.1 Formação de grupos de pesquisas

Uma das principais atividades dos pesquisadores é a participação em Congressos, pois é uma ocasião onde eles apresentam os seus trabalhos e podem encontrar outros colegas que tenham interesses em temas de pesquisas similares. Em muitos desses congressos, é comum a participação de centenas ou até milhares de pessoas, muitas vezes em lugares amplos e dispersos, como hotéis e centros de convenções.

Nesta situação, o encontro de pessoas com os mesmos interesses se torna difícil e na maioria das vezes acontece de forma casual, através do intermédio de pessoas conhecidas. Uma aplicação que ofereça aos participantes do congresso a opção de formação de grupos e troca de mensagens com os outros participantes que tenham interesses em comum traria melhor aproveitamento desta atividade.

Esta aplicação pode ser uma extensão da aplicação proposta neste trabalho, visto que ela já oferece a capacidade de troca de mensagens e de formação de grupos. Para melhor se adaptar ao contexto, seria interessante a adição da capacidade dos usuários de criar cartões de visitas virtuais e de salvar e criar catálogos com cartões de visitas de outros participantes da rede.

3.3.2 Comercio eletrônico

Com a popularização e crescimento dos grandes centros de compras (*shoppings centers*) é comum que os consumidores frequentem locais com dezenas e até centenas de lojas dispersas em vários andares. Neste tipo de ambiente, é comum a existência de grande variedade de lojas, com vários tipos de produtos, o que dificulta muito a procura por um item específico.

Neste tipo de local, existe o potencial para uma aplicação onde os consumidores possam formar uma REPI-A com os comerciantes que oferecem os produtos que eles estão interessados e também com outros consumidores que também procurem ou já possuam o produto, para trocar informações.

Por ser um local muito amplo, seria importante a participação de muitos consumidores e comerciantes, o que viabilizaria que a rede se formasse através da Colaboração.

Este tipo de rede também pode ser aproveitada para feiras de ruas ou mercados populares, onde não há condições de estabelecer uma infraestrutura de comunicação fixa.

A aplicação mensageira proposta neste trabalho também poderia ser aproveitada para este contexto, para a utilização da capacidade de troca de mensagens e formação de grupos. Além disso, ela pode ser aprimorada para ter as seguintes funcionalidades:

- categorizar os interesses, de forma a facilitar a localização dos produtos;
- permitir a criação de textos padrão, para os vendedores divulgarem seus preços e condições de pagamentos e para os usuários procurarem os seus produtos;
- permitir a criação de cartões pessoais virtuais, para os vendedores distribuírem aos clientes interessados;
- criação de um canal exclusivo para divulgação de promoções das lojas, de acordo com a categoria.

3.3.3 Apoio em situações de emergência

Em situações de emergência, principalmente em casos de desastres naturais, como enchentes e terremotos, é comum que a infra-estrutura fixa e serviços básicos de comunicação, como a telefonia celular, fiquem prejudicados ou indisponíveis. Além disso, neste tipo de situação as equipes de resgates costumam mobilizar dezenas e até centenas de pessoas que precisam trocar informações com rapidez e baseadas em interesses momentâneos.

Neste contexto, a REPI-A pode ser uma ferramenta de apoio para a organização das equipes de resgates em grupos de interesses para troca de informações para comunicação direcionada, facilitando a comunicação dentro e entre os grupos. Outra opção interessante é a criação de canais especiais para distribuição de recursos importantes como equipes de paramédicos, equipamentos para remoção de pessoas feridas como macas e outros tipos de equipamentos médicos e de socorro.

Outra funcionalidade é a utilização do envio automático de mensagens por pessoas que precisam de socorro. Neste caso, uma mensagem padrão seria configurada e enviada periodicamente pelo sistema de forma automática em um canal de emergência. Esta mensagem pode ser integrada com outros equipamentos, como sistema de GPS presentes em celulares modernos, de forma a facilitar a localização das pessoas que precisam de ajuda. Com este recurso, as pessoas atingidas poderiam ser

localizadas e socorridas mesmo que estejam incapacitadas de passar sua localização para as equipes de resgate.

Para adaptar a aplicação apresentada neste trabalho para este fim, a ferramenta de envio automático de mensagens, apresentado na Seção 3.2, pode ser adaptada para permitir a inserção do texto padrão do pedido de resgate. Além disso, seria interessante a categorização dos canais de comunicação por interesse, de forma a facilitar a localização de serviços e equipes de resgates.

Capítulo 4

Avaliação Experimental

A avaliação experimental é uma etapa importante e desafiante do processo de desenvolvimento de protocolos de comunicação para redes Ad Hoc.

Um dos desafios desta tarefa é a dificuldade de criação e manutenção deste tipo de ambiente, que exige a montagem, monitoração e controle de redes com dispositivos de comunicação sem fio. Para isso, é necessário o desenvolvimento de aplicações para implementar o protocolo e instrumentar a rede, o que em geral significa desenvolvimento de aplicações para plataformas embarcadas, middlewares e aplicações distribuídas.

Outro desafio é a montagem física da rede Ad Hoc, pois ela exige a disponibilidade de espaço físico para a distribuição dos dispositivos e, de acordo com as propriedades deste ambiente, os resultados da avaliação podem ser alterados.

Alguns exemplos de propriedades do ambiente que podem afetar os resultados são: interferências de outras redes, presença de pessoas ou objetos no caminho da comunicação por rádio frequência e até mesmo características climáticas, como temperatura e umidade.

Em face dessas dificuldades, boa parte dos trabalhos de avaliação de redes Ad Hoc da literatura recorrem a simuladores como o NS-3 [43] e o GloMoSim [44].

Porém, alguns trabalhos, como o apresentado em [45], argumentam que esse tipo de simulação não é robusta o suficiente para garantir que seus resultados são fieis aos obtidos em um ambiente real de redes Ad Hoc.

Um dos principais problemas da simulação de redes Ad hoc é a imprevisibilidade da comunicação por rádio frequência, como mostra o trabalho apresentado em [46]. Alguns problemas deste tipo de comunicação são os seguintes:

- A maioria dos simuladores considera que a transmissão por rádio frequência é perfeitamente omnidirecional, representada como um círculo cujo raio varia de acordo com a potência do rádio. Porém, como é mostrado na própria documentação dos dispositivos [2], esta premissa não é verdadeira, pois dentro

do raio de alcance do transceptor de rádio há zonas em que o sinal é muito forte e outras onde não há nenhum sinal;

- Além disso, em uma rede onde muitos nós transmitem ao mesmo tempo, as interferências e colisões passam a ser um fator com potencial de degradar a qualidade da transmissão. Um dos efeitos que é difícil de simular são as interferências que as várias ondas de rádio frequência causam umas nas outras, visto que elas podem ser construtivas, criando zonas com altas taxas de transmissão, ou destrutivas, criando regiões com ausência de sinal.
- Como é mostrado em [46], o tipo de ambiente onde a rede Ad Hoc é montada tem grande impacto nos resultados e até mesmo no raio de transmissão da rádio frequência. Ambientes abertos e com poucos obstáculos, como estacionamento vazios e campos abertos apresentam melhores resultados que ambientes fechados e com muita interferência, como escritórios.

Diante deste contexto, decidimos que a REPI deveria ser avaliada tanto em um ambiente real quanto em simuladores. A avaliação em um ambiente real com vinte nós é apresentada neste trabalho, enquanto que avaliação em ambiente simulados, com até milhares de nós, será realizada em trabalhos futuros.

O objetivo desta estratégia é utilizar os resultados obtidos no ambiente experimental para validar e aprimorar os simuladores, de forma a minimizar os problemas apresentados.

Neste Capítulo descrevemos os ambientes onde as avaliações experimentais apresentadas neste trabalho foram executadas, a metodologia e as métricas utilizadas e os resultados obtidos.

4.1 Ambiente Experimental

Para avaliar o desempenho da REPI-A e compará-la com outros algoritmos propostos na literatura, realizamos uma série de experimentos em uma rede Ad Hoc com vinte dispositivos Tmote Sky. Os detalhes técnicos deste tipo de dispositivos são apresentados no Apêndice G.

Os objetivos destes experimentos foram os seguintes: avaliar o impacto da variação dos parâmetros da REPI-A em diversos tipos de topologias e configurações de rede, compará-la com outros protocolos da literatura e estudar o impacto que a transmissão por rádio frequência tem em uma REPI-A.

4.1.1 Configuração do ambiente

Todos os experimentos descritos neste capítulo foram realizados com o apoio do sistema SAMCRA. Ele foi usado para parametrizar, monitorar, coletar e consolidar os dados das execuções e repetir os experimentos para aumentar a confiabilidade estatística das amostras.

Em algumas execuções, SAMCRA também foi utilizado para rastrear as mensagens trocadas na rede, coletando dados de todas as mensagens trocadas (*trace*). Os detalhes sobre o funcionamento das funcionalidade oferecidas por SAMCRA são apresentados no Apêndice E.

Com o objetivo de avaliar diferentes cenários, a rede foi avaliada em três topologias diferentes. Para melhor organizar o texto, elas são chamadas de Topologia 1, Topologia 2 e Topologia 3, respectivamente.

Todos os experimentos foram realizados com máquinas com o sistema operacional Windows XP com *Service Pack 3* e máquina virtual Java 1.6. Para a execução remota dos processos, utilizamos arquivos de lote (*batch scripts*) do MS-DOS combinados com a ferramenta de execução remota de processos PsExec, do pacote de ferramentas PsTools [47] da Microsoft.

4.2 Métricas

A avaliação de protocolos de roteamento para redes Ad Hoc envolve o estudo do comportamento de uma série de variáveis da rede, tais como a quantidade de mensagens que foi trocada entre os nós, a quantidade de mensagens descartadas, quais foram os grupos formados na rede, a eficiência dos mecanismos de controle de propagação de mensagens, o custo de envio das mensagens, a capacidade de transmissão da rede sem perdas, tempo de entrega de mensagens, entre outras [48].

Além das métricas tradicionais das redes Ad Hoc, criamos novas métricas para avaliar a Colaboração em uma REPI, visto que esse é um conceito introduzido por este protocolo e que não está presente em outros trabalhos da literatura.

Na avaliação da REPI-A, utilizamos sete métricas de avaliação da rede que têm como insumos os dados gerados nos experimentos apresentados nas Seções 4.4, 4.5 e 4.6. A próxima seção apresenta a notação usada para especificá-las.

4.2.1 Notação usada nas métricas

Nesta seção, apresentamos a notação utilizada nas próximas seções para especificar as métricas propostas neste trabalho. A seguir é apresentada a lista de variáveis utilizadas:

N : Total de nós ativos e inativos na rede. Onde os nós ativos são aqueles que encaminham mensagens, enquanto os nós inativos são aqueles que não encaminham mensagens. Exemplos de motivos para um nó não encaminhar mensagens são: só receber mensagens que não atendem aos critérios de encaminhamento do protocolo REPI-A e estar em uma região com problemas de recepção, em função de interferências de rádio frequências;

$C = \{c_1, c_2, \dots, c_8\}$: Conjunto de características de um participante da REPI-A. Este conjunto é composto pelo termos do Prefixo de um participante da REPI-A cuja probabilidade de ocorrência seguem preferencialmente a distribuição probabilística normal. Esse conjunto é utilizado para determinar o encaminhamento de mensagens. Neste trabalho, este conjunto pode ter até oito elementos. Este conjunto também é referenciado como Prefixo C ao longo deste texto;

$Y = \{y_i | i \in \mathbb{N}\}$: Conjunto de interesses de um participante da REPI-A. Este conjunto é composto pelos termos do prefixo cuja probabilidade de ocorrência seguem preferencialmente a distribuição Zipf. Estes campos são utilizados para o endereçamento de mensagens;

$P = C \cup Y$: Prefixo de um participante da REPI-A, é composto pela união de suas características e interesses;

$N^*(y_i)$: Número de nós ativos com um interesse $y_i \in Y$ qualquer, onde $N^*(y_i) \leq N$;

$NF(y_i)$: Número de nós fontes, que são aqueles que geram mensagens com um interesse y_i qualquer, onde $NF(y_i) \geq 1$;

$ND(y_i)$: Número de nós destinatários da mensagens contendo um interesse y_i qualquer, onde $ND(y_i) \geq 1$;

$NFD(y_i)$: Número de nós que além de fonte também são destinatários das mensagens da rede que circulam com um interesse y_i qualquer. $NFD(y_i) = ND(y_i) \cap NF(y_i)$;

$NENC(y_i)$: Número de nós encaminhadores não colaboradores. Nós com um interesse y_i qualquer que encaminham mensagens enviadas por outros nós que também possuem este interesse y_i ;

$NC(y_i)$: Número de nós colaboradores. Nós que não tem um interesse y_i qualquer e que encaminham mensagens contendo este interesse y_i , colaborando para a formação da rede;

$ND^*(y_i)$: Número de nós que receberam mensagens com um interesse y_i qualquer;

D: Conjunto de dados de uma mensagem que são as informações que a aplicação que utiliza a REPI-A transmite na rede (*payload*). Na aplicação mensageira apresentada neste trabalho, *D* é composto pelo texto que o participante inseriu na aplicação e o seu apelido;

$M(y_i) = C \cup \{y_i\} \cup D$: Mensagem com o conjunto de características do participante, e um interesse y_i , representado por um conjunto unitário, e os dados da mensagem;

$MD(y_i)$: Mensagem com um interesse y_i qualquer e que foi entregue a um outro nó também interessado em y_i ;

$MEC(y_i)$: Mensagem encaminhada colaborativa. Mensagem que contém um interesse y_i qualquer e que é encaminhada por outro nó que não tem interesse em y_i ;

$MEI(y_i)$: Mensagem encaminhada por interesse. Mensagem com um interesse y_i que é encaminhada por outro nó que também tem interesse em y_i ;

$MDESC$: Mensagem descartada. Total de mensagens descartadas pelo protocolo REPI-A por qualquer motivo. Exemplos de motivo de descarte são: mensagem excedeu número de saltos permitidos, o nó já encaminhou uma cópia daquela mensagem anteriormente e a mensagem foi inserida originalmente na REPI-A pelo próprio nó que a recebeu;

TMR : Total de mensagens na rede, é igual a $MEC + MEI + MDESC$;

NRE : Número de repetições de execução do experimento. Neste trabalho, $NRE = 20$;

NS : Número de Saltos. Número de nós pelos quais a mensagem foi encaminhada do nó fonte (NF) ao nó destino (ND);

TNS : Total do Número de Saltos. Número de nós pelos quais a todas as mensagens da rede passaram antes de serem entregues ou descartadas;

$TE(y_i)$: Tempo médio de envio das mensagens contendo um interesse y_i qualquer;

Nas próximas Seções são apresentadas as métricas propostas e seus objetivos.

4.2.2 Taxa de Entrega de Mensagens (Métrica 1)

Esta métrica avalia a capacidade da rede de entregar as mensagens enviadas pelos nós. Com ela é possível avaliar o impacto da variação dos parâmetros da rede na

entrega de mensagens e determinar, por exemplo, qual o limite de mensagens que pode circular na rede mantendo um determinado nível de qualidade de serviço.

A Taxa de Entrega de Mensagens (\overline{TEM}) é determinada pela equação 4.1.

$$\overline{TEM} = \frac{\sum_{j=1}^{NRE} \frac{MD_j}{(M*ND)}}{NRE} \quad (4.1)$$

Onde j representa o valor da variável em cada uma das repetições da execução do experimento.

A expressão $(M * ND)$ representa o número de mensagens que seriam entregues ao destinatários em uma rede sem perdas e é igual em todas as repetições da execução do experimento. A multiplicação pelo número de nós destinatários é necessária em função da difusão de mensagens realizada pela REPI-A, de acordo com o número de nós com o mesmo interesse contido na mensagem.

4.2.3 Número de saltos (Métrica 2)

Esta métrica avalia o número de dispositivos pelos quais uma mensagem trafegou antes de chegar ao seus destinatários. O objetivo desta métrica é avaliar quantos dispositivos, em média, são envolvidas na entrega das mensagens ao destinatários.

O número de saltos (\overline{NS}) é determinado pela equação 4.2.

$$\overline{NS} = \frac{\sum_{j=1}^{NRE} NS_j}{NRE} \quad (4.2)$$

4.2.4 Tempo médio de Entrega (Métrica 3)

Esta métrica avalia o tempo médio de entrega das mensagens da rede, indicando qual é o tempo, em milissegundos, que uma mensagem demora para transitar de um dispositivo para outro. Esta métrica é importante para avaliar o tempo de entrega de mensagens em redes com muito saltos e o seu resultado será usado como insumo em trabalhos futuros, onde a REPI será avaliada em simuladores.

O Tempo Médio de entrega (\overline{TM}) é determinada pela equação 4.3.

$$\overline{TM} = \frac{\sum_{j=1}^{NRE} TE_j}{NRE} \quad (4.3)$$

4.2.5 Mensagens Colaborativas (Métrica 4)

Esta métrica avalia qual foi a porcentagem de mensagens encaminhadas de forma colaborativa em relação ao total de mensagens na rede. O seu objetivo é avaliar a participação das mensagens colaborativas no total das mensagens que circularam na rede, qual é sua variação nas diversas configurações usadas e quais impactos que a Colaboração traz.

A Porcentagem de Mensagens Colaborativas \overline{MC} é determinada pela equação 4.4.

$$\overline{MC} = \frac{\sum_{j=1}^{NRE} \frac{MEC_j}{TMR_j}}{NRE} \quad (4.4)$$

4.2.6 Porcentagem de Nós Colaboradores (Métrica 5)

A porcentagem de nós colaboradores determina quantos dos nós que tinham potencial de contribuir com a rede efetivamente o fizeram. O objetivo desta métrica é determinar qual a relação entre os tipos de topologia e configurações estudados e a contribuição na rede.

A porcentagem de nós colaboradores (\overline{NC}) é determinada pela equação 4.5.

$$\overline{NC} = \frac{\sum_{j=1}^{NRE} \frac{NC_j}{ND_j}}{NRE} \quad (4.5)$$

4.2.7 Custo da Colaboração (Métrica 6)

Esta métrica determina qual foi a média de mensagens colaborativas encaminhadas por nó. O objetivo dela é verificar qual é o consumo de recursos nos nós colaboradores e, conseqüentemente, determinar em quais configurações da rede eles são menos onerados.

O Custo da Colaboração (\overline{CCOL}) é determinado pela equação 4.6.

$$\overline{CCol} = \frac{\sum_{j=1}^{NRE} \frac{MEC_j}{NC_j}}{NRE} \quad (4.6)$$

4.2.8 Numero de Nós Colaboradores (Métrica 7)

Esta métrica determina quantos nós da rede foram colaboradores.

4.3 Procedimento experimental

Nesta Seção, apresentamos os experimentos realizados nas três topologias propostas. Como algumas configurações deles variaram de acordo com a topologia onde foram executados, eles serão apresentados em duas partes: primeiro, são apresentados os objetivos e configurações básicas que são comuns a todas Topologias; depois são apresentadas as configurações particulares dos experimentos em cada topologia, em conjunto com a descrição destes.

4.3.1 Experimento 1 - Variação de mensagens na rede

O objetivo deste experimento foi avaliar os impactos causados pela variação do número de mensagens inseridas na rede. Com isso, estudamos as seguintes características da rede: capacidade de entregar mensagens em cenários com congestionamento de mensagens; escalabilidade da rede; e determinar qual é o regime de operação da rede onde a entrega de mensagens é maximizada.

Para isso, todos os Tmotes foram configurados para enviar um mesmo interesse y_1 , ou seja, todos os nós eram destinatários das mensagens, e o número de nós enviando mensagens é alterado ao longo das execuções.

Na primeira execução do experimento apenas um Tmote enviou mensagens, na segunda quatro Tmotes enviaram, e depois oito, doze, dezesseis e vinte nós. O tipo de filtragem foi parcial, ou seja, basta apenas que apenas um campo do Prefixo C seja coincidente para que as mensagens sejam encaminhadas.

4.3.2 Experimento 2 - Impacto da variação de destinatários na rede

O objetivo deste experimento foi analisar os impactos causados pela variação do número de nós destinatários na rede. Para isso, o número de mensagens inseridas na rede foi mantido constante, enquanto o número de nós destinatários foi variado.

Na primeira execução, quatro Tmotes foram destinatários das mensagens, na segunda oito Tmote, depois doze, dezesseis e, finalmente, na última execução, todos os vintes Tmotes.

4.3.3 Experimento 3 - Variação do número de características no prefixo

O objetivo deste experimento foi analisar os impactos causados pela variação do número de campos de características¹ do prefixo. Além disso, este experimento também foi executado com um algoritmo de inundação e um algoritmo probabilístico *Gossip*, com várias configurações de probabilidade de transmissão.

Em todas as execuções cinco Tmotes enviaram mensagens na rede, enquanto outros cinco Tmotes foram configurados como destinatários.

Já a variação ao longo das execuções do experimento foi a seguinte: nas cinco primeiras execuções o número de campos no prefixo C foi variado com os seguintes valores: um, dois, quatro, cinco, seis e oito. A execução com cinco campos foi adicionada após a análise preliminar dos dados, que indicaram que o melhor valor poderia estar entre quatro e seis campos.

Além disso, foi executado um experimento com um algoritmo de inundação e um algoritmo *Gossip* com quatro configurações de probabilidade de transmissão: 25%, 50%, 75% e 85%.

4.3.4 Experimento 4 - Tamanho da rede

O objetivo deste experimento foi analisar os impactos do número de Tmotes no tempo médio de entrega de mensagens na rede. Para isso, número de Tmotes na rede é variado para verificar se em uma rede com mais dispositivos o tempo médio de entrega por salto é constante.

Em todas as execuções apenas um nó inseriu mensagens na rede e o número de nós na rede foi variado em quatro, oito, doze e dezesseis.

4.4 Topologia 1

Nesta seção, apresentamos as configurações utilizadas na primeira topologia de rede proposta neste trabalho e os resultados obtidos nos experimentos.

4.4.1 Definição dos parâmetros de execução

A Topologia 1 foi a primeira a ser montada e nela enfrentamos as maiores dificuldades para encontrar a configuração ideal para a REPI-A, visto que este é o primeiro trabalho prático a implantar uma REPI em uma rede Ad Hoc.

¹Para melhorar a organização do texto, os campos de características do Prefixo C serão referenciados apenas como campos do prefixo C.

Para realizar experimentos propostos, definimos alguns dos parâmetros das execuções, elencados a seguir:

- Número de execuções de cada experimento. Este valor é importante para determinar quantas vezes cada experimento deveria ser repetido para garantir que os dados coletados estavam dentro de um nível de confiança de 95% com menos de 5% de erro, que foram os valores mínimos que consideramos satisfatórios;
- Número de mensagens enviadas, esse valor também foi estipulado de forma a atender aos critérios de confiabilidade estatística do item anterior;
- Taxa de envio de mensagens. Representada pelo intervalo de envio entre duas mensagens consecutivas, este valor é importante pois, como é mostrado em [46] e depois foi comprovado pelo resultados obtidos nos experimentos, este intervalo pode ter grande impacto na Taxa de Entrega de Mensagens da rede.

A primeira decisão que tomamos foi o número de execuções de cada experimento. Para isso, fizemos alguns experimentos preliminares com vinte execuções e, a partir de cálculos com os dados obtidos, chegamos a conclusão que este valor seria suficiente para garantir um nível de confiança de 95% com cerca de 3,04% de erro, que atendia aos nossos requisitos.

O próximo passo foi determinar o intervalo de tempo entre o envio de dois pacotes consecutivos. Em [46] o autor realiza um estudo detalhado do impacto do número de mensagens na taxa de entrega de redes com TinyOS e o padrão 802.15.4/ZigBee. O autor demonstra que com intervalos maiores do que um pacote a cada dois segundos (0,5 pacotes por segundo) a rede começa a ter perdas.

Porém, os experimentos apresentados utilizaram dispositivos de comunicação com largura de banda nominal de 20 Kbps, que é 12,5 vezes menor do que a dos Tmote Sky utilizados em nosso experimento, que possuem largura de banda nominal de 250 Kbps.

Com base nestes valores, estimamos que a transmissão em um intervalo randômico entre meio e um segundo e meio seria suficiente para que a rede só apresentasse perdas relevantes nos experimentos com grande quantidades de mensagens circulantes e, como este era um dos efeitos que buscávamos observar, este valor foi utilizado inicialmente.

Depois de definir o número de execuções de cada experimento e o intervalo de envio de mensagens na rede, só faltava definir o número de mensagens enviadas em cada execução. A questão principal era que o número de mensagens deveria ser maximizado, para termos uma boa amostra, mas ao mesmo tempo, este número

não poderia ser muito grande, para não inviabilizar o tempo de execução de cada experimento.

Como estavam previstos quatro experimentos, que foram executados vinte vezes cada, com uma média de seis configurações diferentes, chegamos a conclusão que mil pacotes seria o número ideal para realizar as execuções em um prazo de cerca de três semanas, tempo que tínhamos para a realização dos experimentos com esta topologia.

4.4.2 Configuração e distribuição dos Tmotes

Os vinte dispositivos Tmote Sky foram distribuídos em cinco computadores, todos eles ligados através da porta USB e com uma instância da aplicação mensageira para controlar cada dispositivo. Os Tmotes foram distribuídos da seguinte forma: três das máquinas possuíam cinco Tmotes ligados; uma das máquina com quatro Tmote; e uma delas com apenas um Tmote. Essa distribuição levou em conta a localização dos Tmotes e a capacidade das máquinas. A máquina que possuía apenas um Tmote era a mesma onde SAMCRA estava foi executado e por isso ela não foi sobrecarregada com vários Tmotes.

Em virtude das limitações físicas de espaço no local dos experimentos, a potência de transmissão do rádio de comunicação dos Tmotes foi configurada para o seu valor mínimo, com alcance de transmissão de aproximadamente cinquenta centímetros e com potência de transmissão de -25 dbm.

Esta configuração foi feita inserindo o parâmetro “*CFLAGS=-DCC2420_DEF-RFPOWER=1*” durante a compilação do middleware Tmote. O objetivo desta configuração era evitar que todos os Tmotes conseguissem transmitir pacotes diretamente para todos os outros dispositivos da rede. Com essa configuração, os Tmotes tiveram uma média de oito vizinhos.

Os Tmotes foram organizados em cima de quatro mesas e distribuídos conforme mostra a Figura 4.1, com distância média de 20 cm entre eles.

Para evitar que o solo causasse interferências no sinal de transmissão, como mostra os experimentos realizados em [49], os Tmotes foram suspensos em cones de papel, produzidos para este experimento, como mostra a Figura 4.2.

4.4.3 Resultados das execuções dos experimentos na Topologia 1

Ao longo das execuções com a Topologia 1, encontramos problemas que fizeram que os resultados ficassem aquém do esperado e fossem influenciados por fatores externos aos nossos objetos de pesquisa.



Figura 4.1: Mapa com a distribuição espacial dos Tmotes na topologia 1.

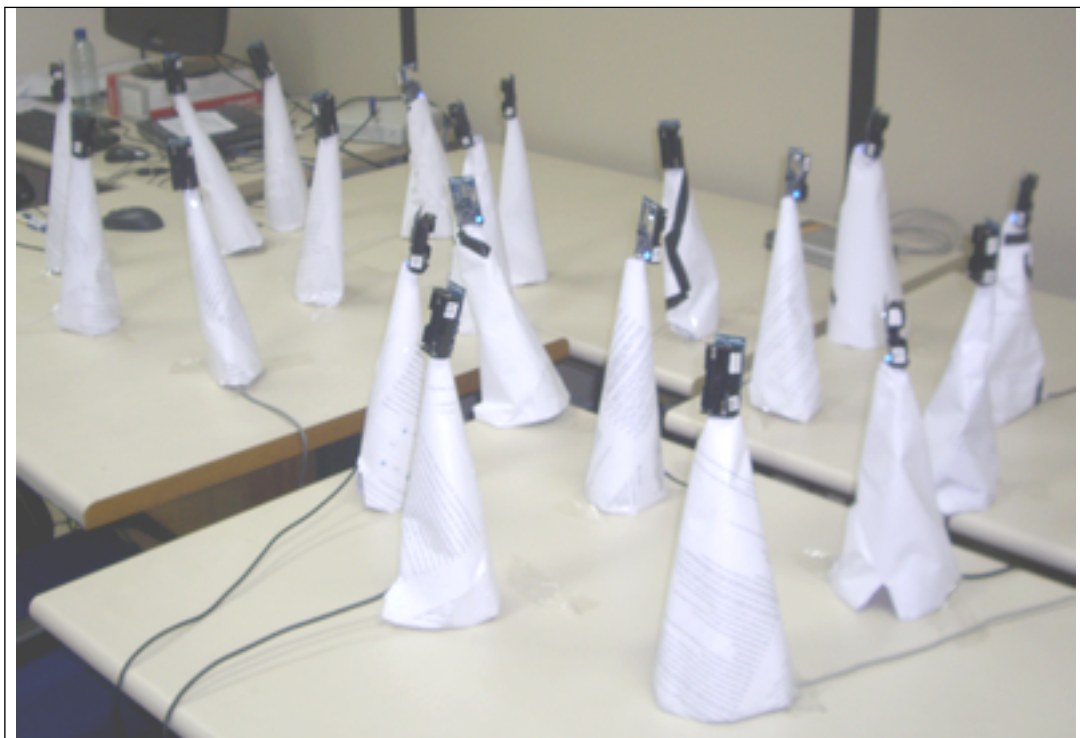


Figura 4.2: Cones utilizados para suspender os Tmotes e evitar interferências do solo na transmissão na topologia 1.

Em função disso, os resultados dos experimentos com esta topologia são apresentados a parte, no Apêndice A. Na próxima seção, os problemas enfrentados na Topologia 1 são apresentados junto com as soluções que adotamos para que eles não fossem repetidos na Topologia 2.

4.5 Topologia 2

Ao longo dos experimentos com a Topologia 1 detectamos alguns problemas que buscamos resolver com a Topologia 2. Esse problemas e as soluções adotadas são discutidas ao longo desta seção.

O primeiro deles foi a baixa Taxa de Entrega de Mensagens, que no melhor caso ficou próxima de 66%. Uma das hipóteses que levantamos como uma das possíveis causas desse problema foi a alta concentração dos Tmotes em uma pequena área. Por isso, na Topologia 2 a potência de transmissão foi mantida e os nós foram redistribuímos em uma área maior, como mostra a Figura 4.3. Nesta configuração, os Tmotes ficaram distante entre si pelo menos cerca de trinta centímetros, contra cerca de vinte centímetros da Topologia 1.

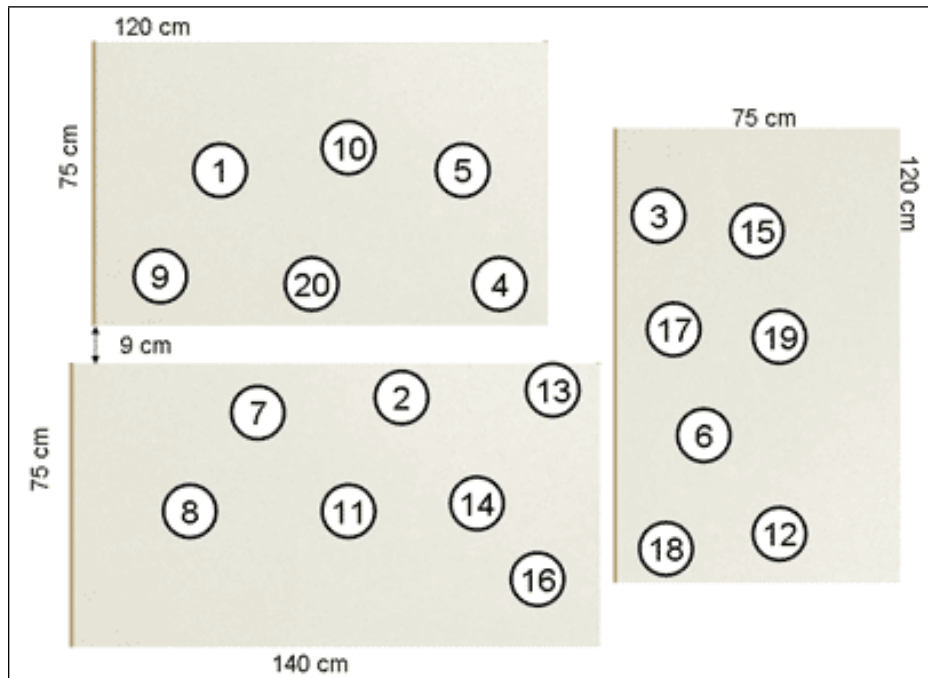


Figura 4.3: Mapa com a distribuição espacial dos Tmotes na topologia 2.

Outro problema encontrado é que os cones de papel de sustentação dos Tmotes, que foram confeccionados manualmente para os experimentos, não se mostraram confiáveis. Em várias ocasiões eles não apresentaram a sustentação necessária e caíram, nos obrigando a repetir experimentos. Para substituí-los, procuramos soluções que sejam de fácil acesso e baixo custo.

Como mostra a Figura 4.4, a solução encontrada foi a utilização de cones de papelão que, devido a baixa altura e grande largura da base apresentaram estabilidade suficiente para que ao longo de todos os experimentos não houvesse nenhuma ocorrência de queda dos Tmotes. Além disso, com eles obtivemos altas taxas de entrega de mensagem, o que indica que o impacto negativo das interferências do

solo foi reduzido.



Figura 4.4: Cones utilizados para suspender os Tmotes e evitar interferências do solo na transmissão na topologia 2.

O último problema que identificamos na Topologia 1 foi que o intervalo de envios de mensagens também poderia ser um dos responsáveis pela baixa Taxa de Entrega de Mensagens. Para avaliar esta hipótese, realizamos experimentos com outros intervalos de tempo de envio na Topologia 2.

4.5.1 Entrega de mensagens em uma execução na REPI-A

Além da realização dos experimentos previstos para a Topologia 2, realizamos uma execução extra para estudar em detalhe o comportamento da rádio frequência nesta Topologia. O objetivo deste estudo era ter um maior embasamento para entender os resultados obtidos nos outros experimentos e entender o comportamento da rádio frequência e sua influência na entrega de mensagens na REPI-A.

Para isso, foi feita uma execução com a primeira configuração do experimento 1, com o Tmote 4 enviando 100 mensagens em um intervalo aleatório de $[5;60]$ segundos entre duas mensagens consecutivas. Todos os nós foram configurados com o mesmo interesse y_1 e as características eram alteradas a cada nova execução de forma aleatória, em um total de vinte execuções.

Para aprofundar o estudo, a execução gravou o rastreamento de mensagens da execução e a execução com taxa de entrega mais próxima da média das vinte execuções foi analisada de forma detalhada. A média de recepção de todas as vinte execuções foi de $91,01\% \pm 2,95\%$ e a execução escolhida para a análise foi a de número 12, que teve média de recepção de $91,21\% \pm 5,37\%$.

Mensagens entregues em um salto

A primeira análise feita foi a verificação do alcance de RF do Tmote 4, única fonte mensagens na rede. A Figura 4.5 apresenta os nós que recebem mensagens diretamente do nó 4, em um salto. Para facilitar a interpretação da figura, foram delineadas três zonas que categorizam os nós: uma zona com grande probabilidade de entrega que engloba oito nós, onde todos eles recebem mais de 90% das mensagens; uma zona de instabilidade, com dois nós com recepção de 36% e 61% das mensagens; e uma zona sem transmissão com nove nós que não recebem nenhuma mensagem.

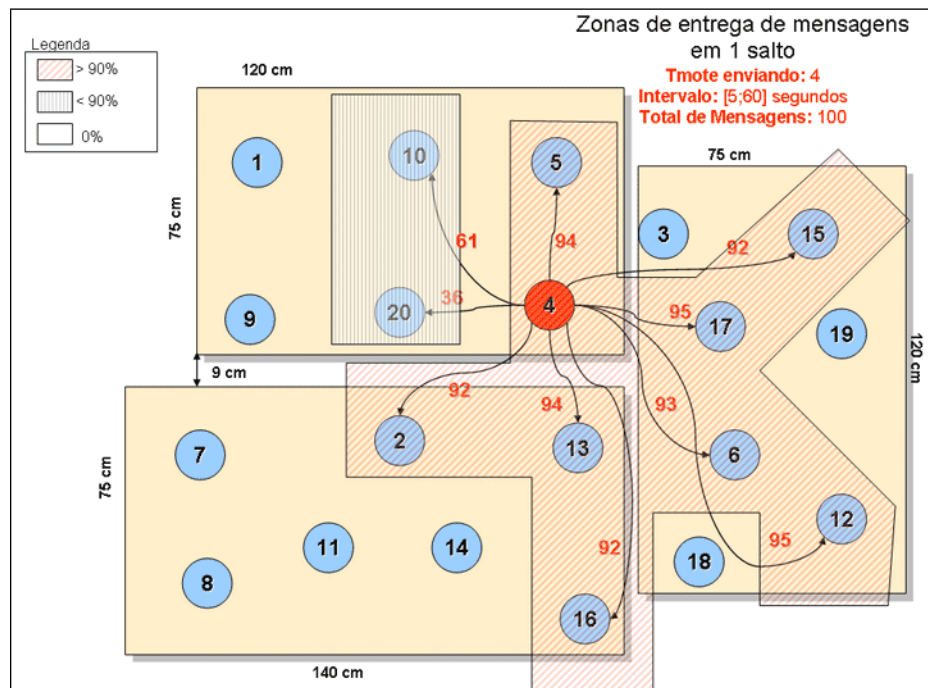


Figura 4.5: Análise da entrega de mensagens em apenas um salto na configuração com apenas um Tmote enviando mensagens.

Analisando a Figura, três casos inesperados são notados: a taxa de entrega dos nós 3, 18 e 19 foi de zero mensagens, enquanto que a dos outros nós que os cercam, inclusive de alguns que estão mais distantes que eles, foi maior que 90%. Lenvantamos duas hipóteses para explicar este fenômeno: a primeira é que, como mostra a Figura 4.6, o padrão de propagação típico da antena do Tmote Sky possui zonas sem transmissão e esses nós podem estar posicionados exatamente nesses pontos; a outra hipótese é que a reflexão dos sinais causou o cancelamento dos sinais dessas regiões, visto que os três nós que apresentaram esse resultado estão em posições parecidas, cercadas por quatro nós com alta taxa de recepção.

Além disso, é possível notar que nenhum nó da rede recebeu as cem mensagens transmitidas, pois os nós com maiores taxas de recepção foram o 12 e o 17, com 95 mensagens recebidas. Ainda assim, havia a possibilidade das cem mensagens

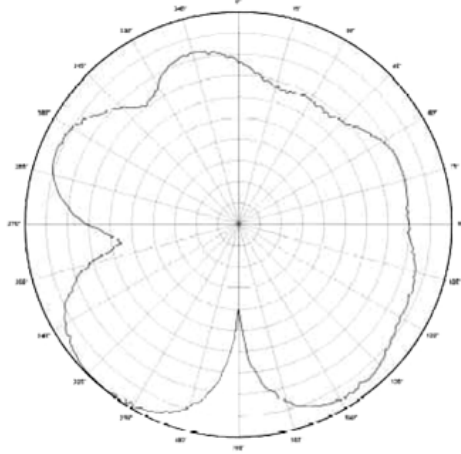


Figura 4.6: Padrão de propagação de rádio-frequência da antena do Tmote Sky, adaptado de [2]

originais terem circulado na rede, visto que não necessariamente as cinco mensagens perdidas foram as mesmas em todos os nós da rede.

Para sanar esta dúvida, analisamos o rastreamento das mensagens para verificar se alguma mensagem não foi recebida por nenhum nó na rede. Foi constatado que as mensagens 39, 57, 79, 83, 96 não foram recebidas por nenhum Tmote. Com isso, confirmamos que apenas noventa e cinco das cem mensagens foram efetivamente injetadas na rede e que as outras cinco se perderam por falhas de comunicação previstas em sistemas com transmissão por rádio frequência.

Mensagens entregues em cada nó da rede

A segunda análise deste estudo verifica o número de mensagens entregues na rede por nó. O seu objetivo é determinar onde houve perda de mensagens e quais foram os motivos dessas perdas, para determinar se elas foram causadas pelos critérios de encaminhamento da REPI-A ou por problemas na transmissão por rádio frequência.

Para isso, foi feito o levantamento do número total de mensagens entregues em todos os nós, assim como o número de mensagens aceitas e o número de mensagens descartadas em função do prefixo C. A Figura 4.7 mostra o resultado deste levantamento. A média de entrega neste cenário foi de $91,21\% \pm 5,37$

É possível notar que a taxa de recepção da maioria dos nós é próxima da média e do desvio padrão, com exceção dos nós 3 e 9, que têm taxas próximas ou abaixo de duas vezes o desvio padrão observado. Além disso, é possível notar que os nós 8, 9, 12, 13, 14 e 17 não encaminharam as mensagens que receberam em função do Prefixo C. A média do total de mensagens recebidas por nó foi de $272,4 \pm 99,7$ mensagens.

A Figura 4.8 mostra o número de vizinhos de cada nó, classificando-os como

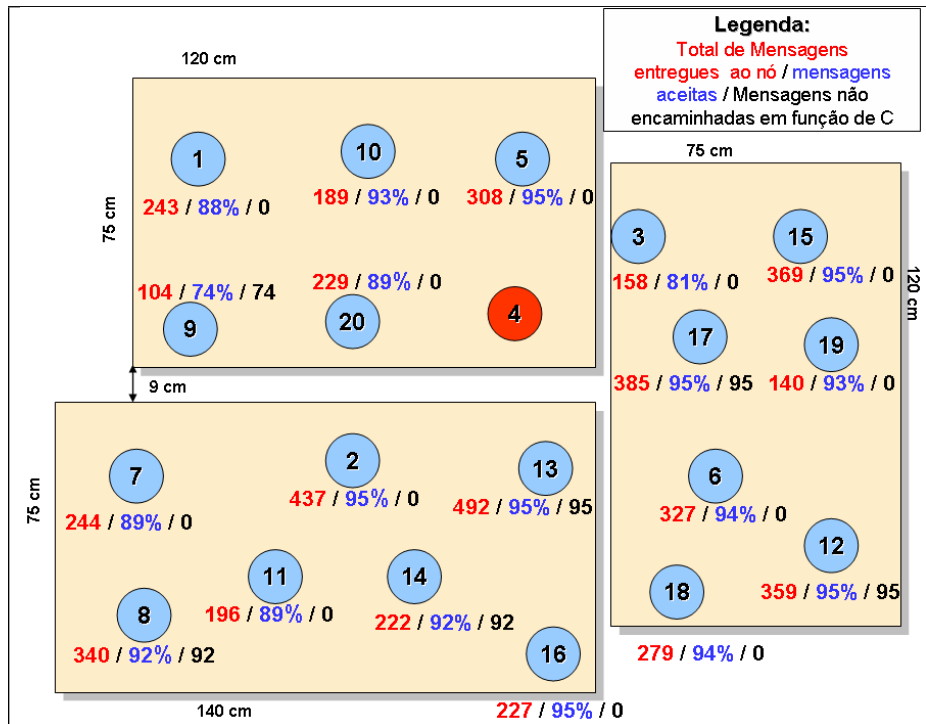


Figura 4.7: Análise do número do total de mensagens, mensagens aceitas pelo destinatário e mensagens descartadas em função do prefixo C.

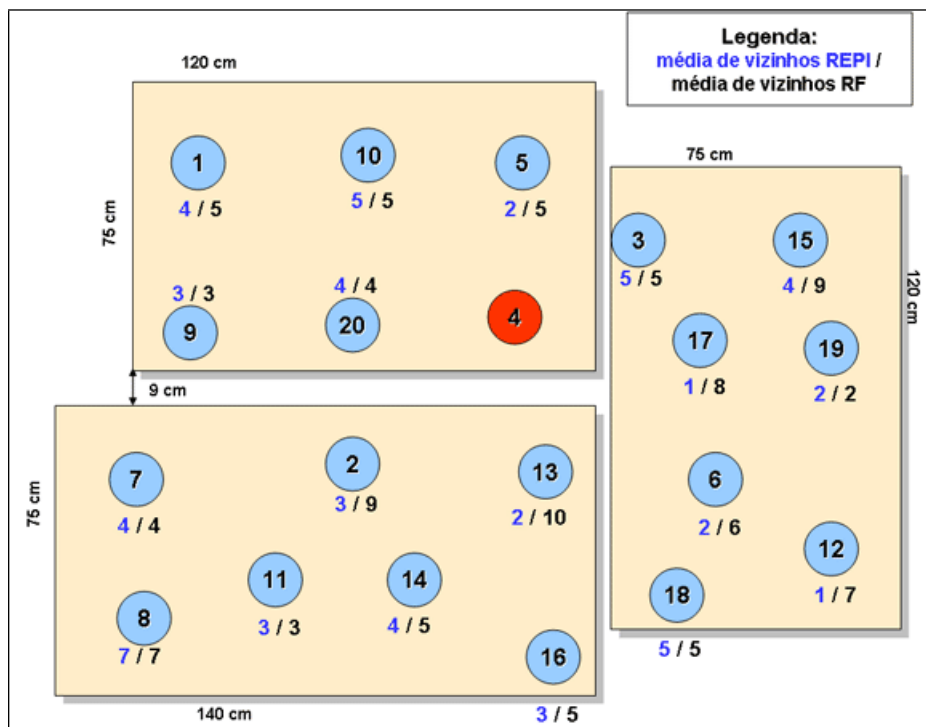


Figura 4.8: Análise do número de vizinhos de Rádio Frequência e da REPI por nó.

vizinhos de rádio frequência (RF) e vizinhos REPI-A. Os vizinhos RF são aqueles dos quais o nó recebe mensagens, mesmo que elas sejam descartadas. Já os vizinhos REPI-A são aqueles dos quais o Tmote recebe mensagens que atendem aos critérios

de encaminhamento da REPI-A.

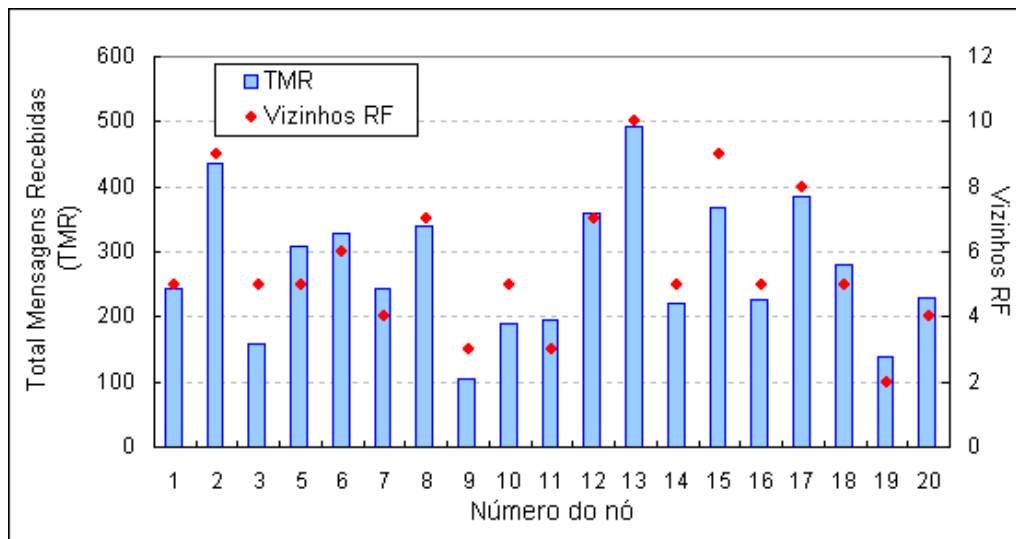


Figura 4.9: Gráfico que apresenta a relação entre o número de vizinhos de rádio frequência e o Total de Mensagens Recebidas por cada nó.

O gráfico da Figura 4.9, que compara o número de vizinhos RF de cada nó com o Total de Mensagens Recebidas (TMR), mostra que o número de vizinhos RF de um nó está intimamente ligado o seu TMR.

Um exemplo desta relação são os nós 9 e 19, que tiveram o menor número de mensagens recebidas, respectivamente 104 e 140, e também são os nós com o menor número de vizinhos RF, três e dois, respectivamente. No outro extremo estão os nós 2 e 13, que receberam o maior número de mensagens e tem a maior quantidade de vizinhos, com 437 e 492 mensagens e 9 e 10 vizinhos, respectivamente.

A seguir, é feita uma análise detalhada do comportamento dos nós 3 e 9, para determinar a causa da baixa taxa de recepção dos dois.

Análise do comportamento do nó 3

A Figura 4.10 apresenta o detalhamento das mensagens recebidas pelo nó 3 e das suas interações com os seus nós vizinhos, mostrando quantas mensagens ele recebeu e aceitou de cada um deles. É possível notar que o nó 3 possui 5 vizinhos, tanto de RF quanto REPI, visto que recebeu e aceitou as mensagens entregues por todos. Porém, é possível perceber que muitas das mensagens recebidas dos seus vizinhos foram repetidas, em especial as vindas do Tmote 18, que entregou trinta e seis mensagens das quais apenas sete foram aceitas. Todas as outras foram descartadas por já serem cópias da uma mensagem que já havia sido entregue através de outro nó.

Para verificar se a rádio frequência foi um dos motivos da perda de mensagens nessa execução, verificamos as interações que o nó 3 teve na média das vinte

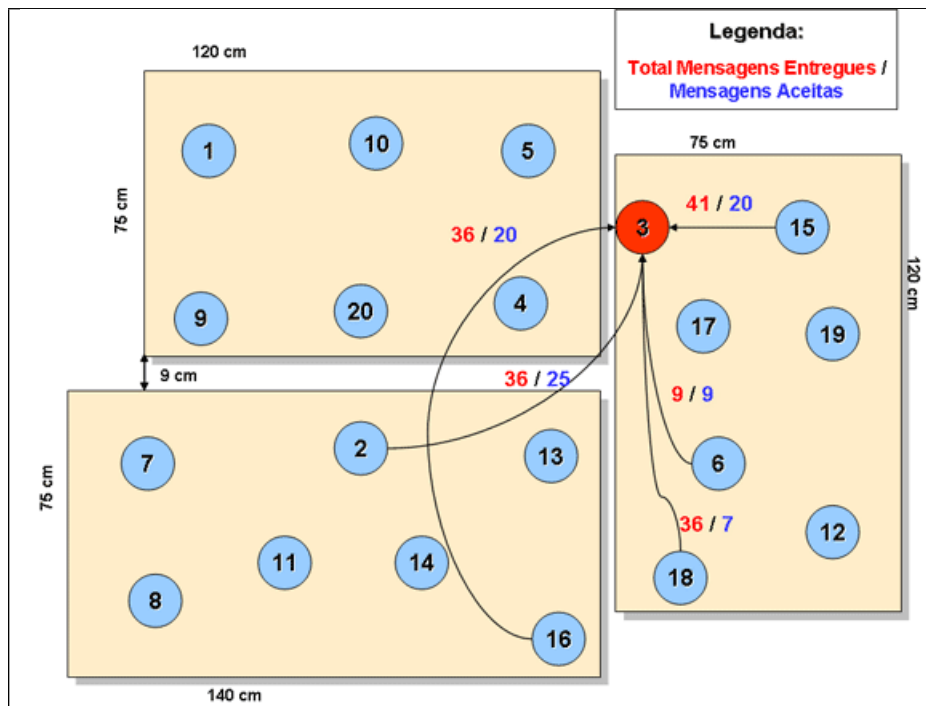


Figura 4.10: Análise das mensagens entregues ao nó 3 na execução estudada .

execuções, onde a influência do prefixo C é minimizada, visto que ele é alterado a cada execução. Com isso, percebemos que na média os maiores fornecedores de mensagens para ele são os nós 2, 15 e 16, com 17, 38 e 11 mensagens em média. Também constatamos que a média de recepção do nó 3 é de $78,5\% \pm 6,9$, compatível com a observada nessa execução, que foi de 81%.

Como os nós 2, 15 e 16 encaminharam todas as mensagens que receberam e os três estiveram entre os nós que receberam as noventa e cinco mensagens que circularam na rede, concluímos que a baixa taxa de recepção do nó 3 é causada por problemas na comunicação por rádio frequência.

Análise do comportamento do nó 9

A Figura 4.11 apresenta a interação do nó 9 com os seus vizinhos imediatos, mostrando quantas mensagens ele recebeu e aceitou de cada um deles. É possível notar que o nó 9 tem três vizinhos RF e REPI: 1, 11 e 15. Também é possível notar que o nó 9 não possui casamento de prefixo C com o nó 4, fonte de todas as mensagens da rede, visto que todas as mensagens que recebeu não foram encaminhadas.

Assim como foi feito no nó 3, avaliamos as interações do nó 9 na média de todas as execuções. Estas interações são representadas na Figura 4.12. Nela é possível perceber que na média das vinte execuções o nó 9 tem sete vizinhos, mais que o dobro que os três da execução estudada. Além disso, a figura representa os nós que não encaminharam mensagens em função do prefixo C, que são marcado com uma

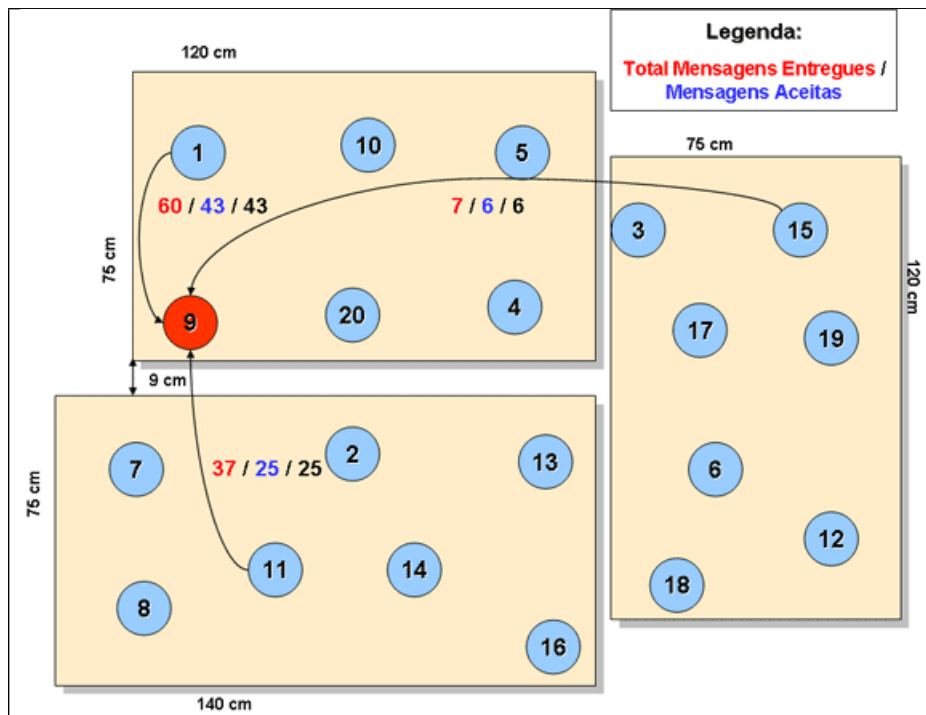


Figura 4.11: Análise das mensagens entregues ao nó 9 na execução estudada.

linha vermelha.

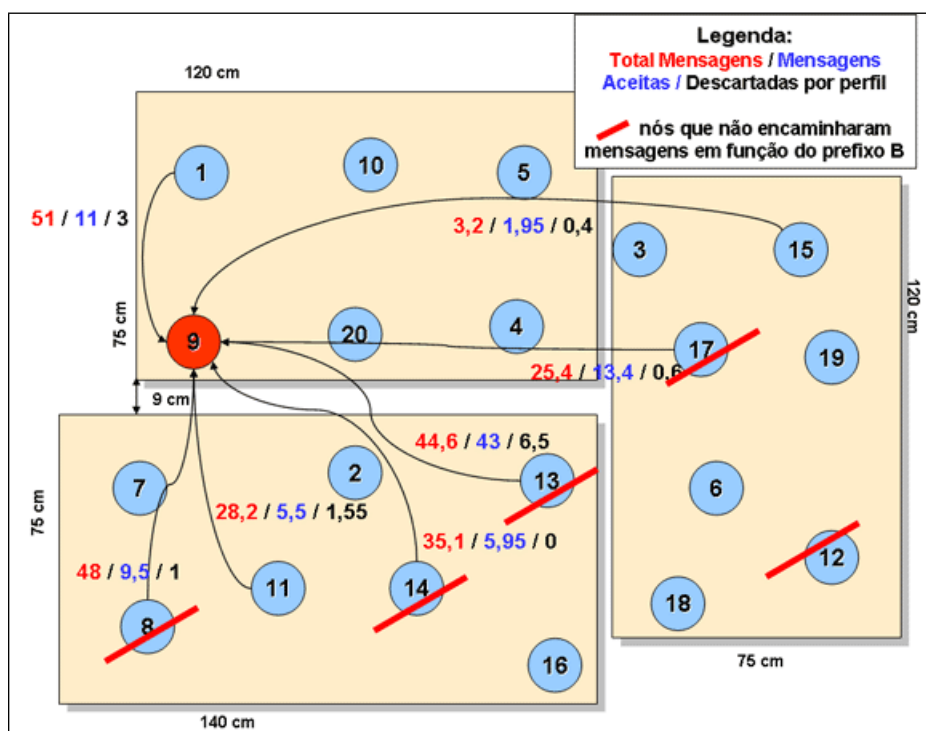


Figura 4.12: Análise das mensagens entregues ao nó 9 em média nas vinte execuções realizadas.

É possível perceber que os nós que mais forneceram mensagens para o nó 9 em média foram os nós 1, 13 e 17, com destaque para o nó 13, com 43 mensagens que

representam 47% das mensagens recebidas em média. Além disso, verificamos que a média de recepção do nó 9 foi em média $90,15\% \pm 4,48$. Ou seja, a taxa de recepção da execução analisada foi menor do que a média subtraída de duas vezes o desvio padrão.

Com isso, concluímos que o critério de encaminhamento de mensagens com base no prefixo C teve impacto negativo nesta execução específica.

Conclusão da análise

O primeiro aspecto importante da análise é que a REPI-A foi capaz de melhorar o desempenho da rede, visto que apenas 10 nós receberam mensagens diretamente do nó 4, num total de 848 mensagens. Enquanto que, através do roteamento da REPI-A, são entregues outras 1095 mensagens para os outros 17 nós da rede.

Além disso, constatamos que o baixo desempenho do nó 3 foi causado por problemas na comunicação por rádio frequência, visto que o seu desempenho é baixo na média das vinte execuções realizadas e que todos os nós que encaminham mensagens para ele atenderam aos critérios de encaminhamento na execução analisada.

Porém, foi constatado que o baixo desempenho do nó 9 na execução doze, que foi avaliada de forma detalhada neste estudo, é causado pela ausência de casamento do prefixo C nos nós 13 e 17, que são os que mais forneceram mensagens para ele na média das vinte execuções, indicando que esta abordagem causou perdas nesta situação.

4.5.2 Experimento 1

O Experimento 1 (variação de mensagens na rede) foi executado na Topologia 2 com a seguinte configuração:

- Primeira execução, apenas o Tmote 4 envia;
- Segunda execução, quatro Tmotes enviam: 1, 4, 14 e 15;
- Terceira execução, oito Tmotes enviam: 1, 2, 4, 7, 10, 12, 15 e 16;
- Quarta execução, doze Tmotes enviam: 1, 2, 3, 4, 7, 9, 10, 11, 12, 15, 16 e 19;
- Quinta execução, dezesseis Tmotes enviam: 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 18, 19;
- Sexta execução, todos Tmotes enviam;

Em todas as execuções a REPI-A é configurada com cinco campos no prefixo C. Os intervalos de envio são variados com as seguintes configurações: [2;4] segundos, [2;20] segundos e [5;60] segundos.

Além disso, um experimento com intervalo de [2;4] segundos foi repetido com cem e mil mensagens para verificar se em ambos os casos temos a mesma confiabilidade estatística. Os resultados desse experimento são mostrados no gráfico da Figura 4.13.

Como é possível notar, os resultados são muito similares, com uma pequena diferença cerca de 2,55% com vinte nós fonte. Como o erro da configuração com cem mensagens é de $\pm 1,54\%$ e na de mil é de $\pm 1,36\%$, consideramos essa diferença desprezível.

Para o nível de confiança de 95% o erro estimado é de 4,12% para a execução com 100 mensagens na rede, contra 3,04% para a execução com 1000 mensagens. Como ambos atendiam ao patamar de menos de 5% que nos estabelecemos, consideramos que os experimentos com cem mensagens eram adequados ao nosso estudo.

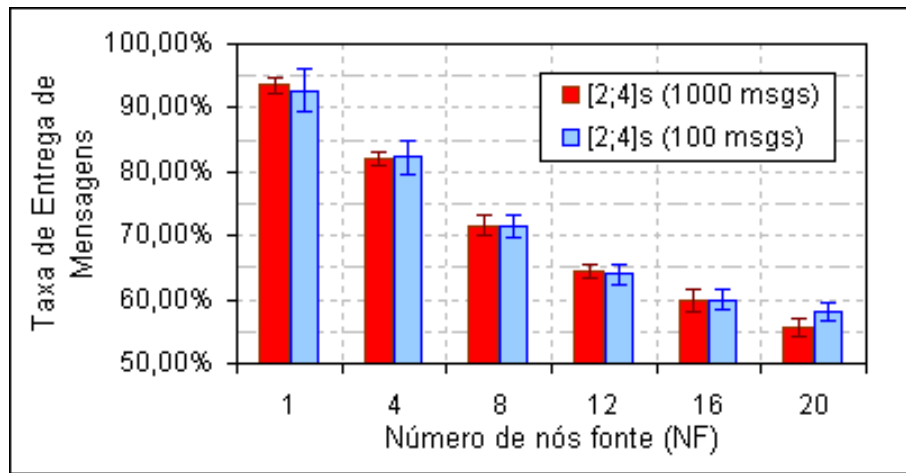


Figura 4.13: Gráfico que mostra o comparativo de dois experimentos realizados com o mesmo intervalo de tempo entre o envio de duas mensagens consecutivas, variando apenas o número de mensagens entregues.

A diminuição do número de mensagens é necessária para que as execuções com intervalo de [5;60] ocorram em tempo hábil. Uma execução com este intervalo de tempo, com envio de mil mensagens e vinte repetições leva entre cerca de 27,77 a 333,33 horas para terminar de executar.

Com exceção do experimento comparativo com cem e mil mensagens, todos os outros experimentos desta topologia foram realizados com cem mensagens.

Taxa de entrega de mensagens em função do número de nós fonte e do intervalo de entrega

Em função dos resultados ruins obtidos nos experimentos preliminares realizados na Topologia 1, onde esta taxa variou entre cerca de 83% até cerca de 45%, resolvemos investigar as causas para este problema. Como dito anteriormente, a primeiro

mudança foi a reorganização dos Tmotes, aumentando em cerca de 50% a distância média entre eles.

A segunda medida foi a busca de um intervalo de envio em que o congestionamento na rede fosse mínimo, de forma a garantir que as nossas medições, cujo foco era na camada de roteamento, não fosse afetada por problemas nas camadas PHY e MAC. Os resultados deste experimento são apresentados na Figura 4.14.

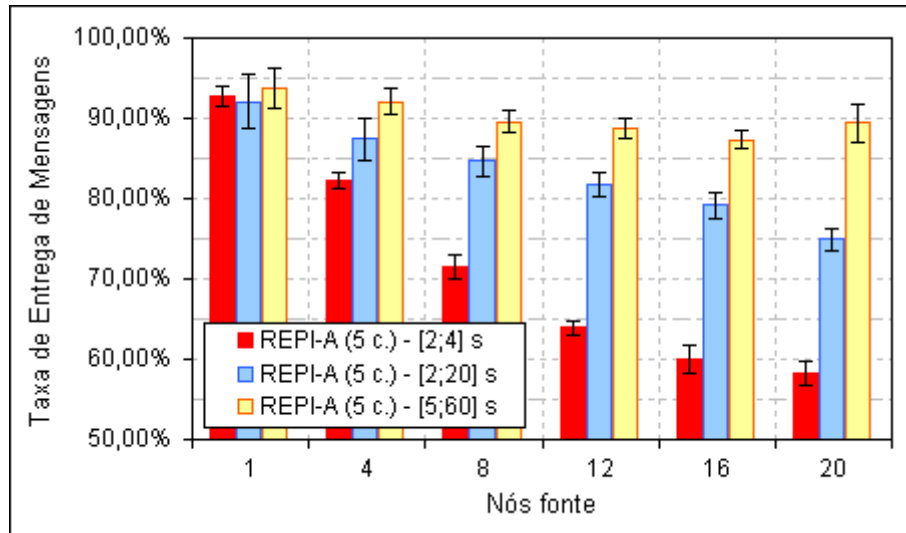


Figura 4.14: Gráfico que mostra a evolução da Taxa Entrega de Mensagens (\overline{TEM}) em função do número de Nós Fonte e do intervalo de tempo entre o envio de duas mensagens consecutivas.

Como é possível perceber, para o intervalo de [2;4]s há uma melhora substancial em relação aos valores obtidos com o intervalo de [0,5;1,5]s com oito campos na Topologia 1. Esta diferença é ilustrada na Tabela 4.1, que mostra que na configuração com oito Tmotes a melhora na taxa de entrega foi de cerca de 29,02%. Porém, fica claro que conforme o número de Tmotes enviando mensagens aumenta, o congestionamento faz com que essa diferença caia progressivamente. Com isso, na configuração com vinte Tmotes a diferença é de apenas 1,61%.

Além disso, nota-se que na configuração com apenas um Tmote enviando a diferença é de apenas 15,92%, o que reforça a tese de que o problema da entrega é causado pelo congestionamento na rede, visto que nesta configuração o congestionamento com intervalo de [0,5;1,5] também é maior.

Voltando a análise do gráfico da Figura 4.14, fica claro que o intervalo entre o envio de duas mensagens consecutivas influencia significativamente a taxa de entrega, principalmente nos cenários com um grande número de mensagens na rede. No pior caso, com vinte Tmotes enviando mensagens, as taxas de entrega foram de cerca 58,20%, 74,88% e 89,38% para as configurações de [2;4]s, [2;20]s e [5;60]s, respectivamente.

Tabela 4.1: Comparação da Taxa de Entrega de Mensagens na Topologias 1 e 2 com dois intervalos de tempo diferentes.

Taxa de Entrega			
Tmotes enviando	Topologia 1 Intervalo [0,5;1,5]	Topologia 2 Intervalo[2;4]	Melhora
1	80,71%	93,56%	15,92%
4	64,83%	82,14%	26,69%
8	55,48%	71,58%	29,02%
12	52,36%	64,46%	23,11%
16	50,84%	59,82%	17,65%
20	54,77%	55,65%	1,61%

Estes resultados comprovam os dados disponíveis na literatura [46] e, considerando a natureza das aplicações mensageiras onde a REPI-A será utilizada, que exigem interação do usuário para ler e escrever novas mensagens, consideramos que o intervalo de [5;60]s é satisfatório e realista.

Além disso, a rede poderá se beneficiar de possíveis futuras melhorias nos padrões de comunicação nas camadas PHY e MAC do padrão 802.15.4 ou de outros a serem utilizados em implementações futuras.

4.5.3 Experimento 2

No experimento 2 (Impacto da variação de destinatários na rede) da Topologia 2, os Tmotes que injetam pacotes na rede são os mesmos em todas as execuções: 5, 11, 15 e 16. Os Tmotes que são destinatários são os seguintes:

- Primeira execução, quatro Tmotes como destinatários: 3, 12, 13 e 20;
- Segunda execução, oito Tmotes como destinatários: 1, 3, 7, 12, 13, 14, 19 e 20;
- Terceira execução, doze Tmotes como destinatários: 1, 3, 4, 6, 7, 9, 12, 13, 14, 18, 19 e 20;
- Quarta execução, dezesseis Tmotes como destinatários: 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14, 17, 18, 19 e 20;
- Quinta execução, todos Tmotes como destinatários;

A seguir são apresentados os dados obtidos com a execução do Experimento 2 na Topologia 2.

Taxa de entrega de mensagens em função do número de destinatários

O gráfico da Figura 4.15 apresenta os impactos do número de destinatários e da variação do intervalo de entrega na Taxa de Entrega de Mensagens (\overline{TEM}) na Topologia 2.

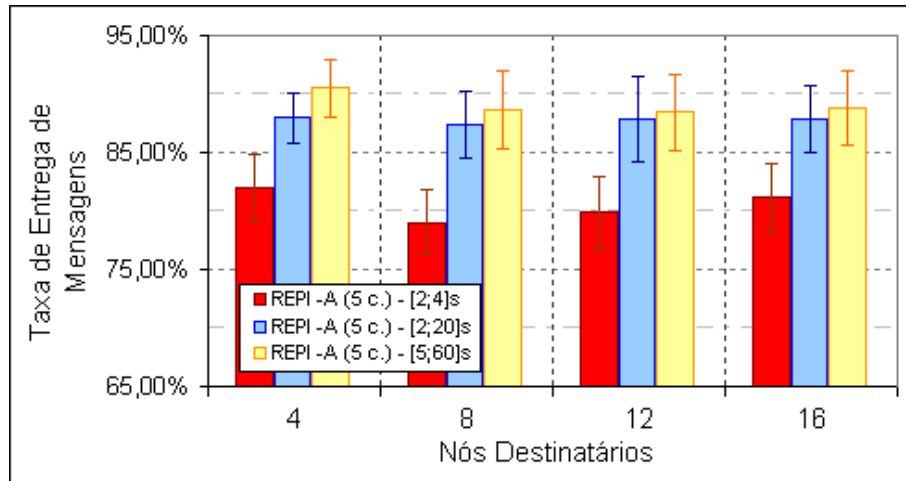


Figura 4.15: Gráfico que mostra a evolução da Taxa de Entrega de Mensagens (\overline{TEM}) em função do número de Nós Destinatários, com três configurações de intervalo de tempo.

Como já era esperado, a \overline{TEM} teve o melhor desempenho na configuração com intervalo de [5;60]s, com 90,50% na configuração com quatro destinatários e 88,79% na configuração com dezesseis destinatários. O desempenho da configuração com intervalo de [2;20]s foi próximo ao da configuração com [5;60]s, com todos os valores entre 87,40% e 87,90%. Da mesma forma que observamos nos outros experimentos, o desempenho da configuração com intervalo de [2;4]s foi inferior, variando entre 79,05% até 81,97%.

Estes resultados comprovam que o congestionamento de mensagens na rede traz impactos negativos na taxa de entrega de mensagens. Nesta situação o impacto foi atenuado pois haviam poucos nós transmitindo na rede, mas ainda assim a diferença entre os resultados foi de cerca de 10%.

Custo da Colaboração

O gráfico da Figura 4.16 apresenta o Custo de Colaboração (\overline{CCOL}) nas três configurações de intervalo de tempo analisadas neste experimento.

Como é possível notar, o custo de Colaboração, que indica o número médio de mensagens que cada nó colaborador encaminhou na rede, cai conforme o número de nós destinatários aumenta. Esse comportamento é esperado, visto que quanto mais nós são destinatários, há menos colaboradores na rede. Porém, ao observar a variação estatística, percebemos que todos os valores estão no mesmo patamar.

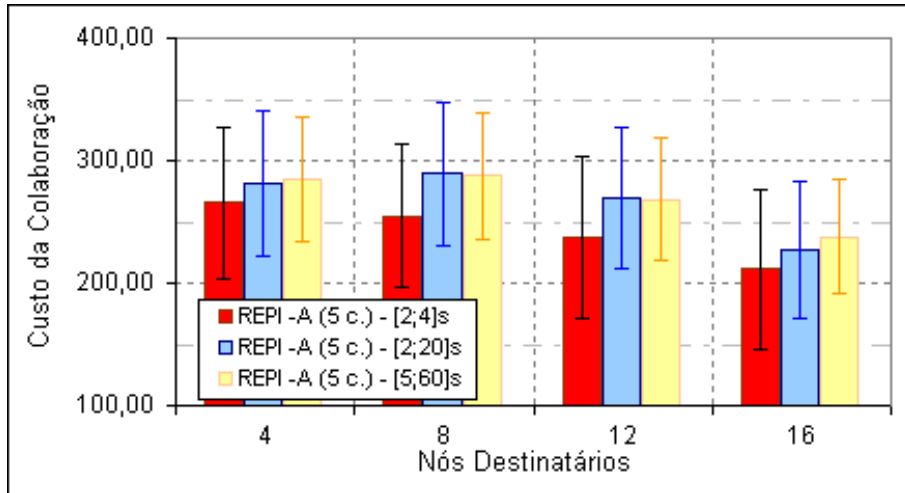


Figura 4.16: Gráfico que mostra a evolução do Custo de Colaboração (\overline{CCOL}) em função do número de Nós Destinatários, com três configurações de intervalo de tempo.

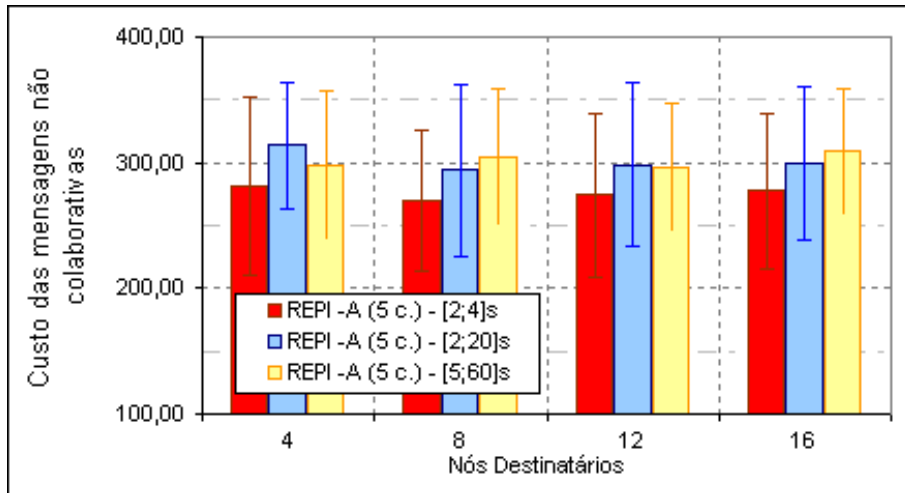


Figura 4.17: Gráfico que mostra a evolução do custo de encaminhamento das mensagens não colaborativas em função do número de Nós Destinatários, com três configurações de intervalo de tempo.

Outra questão fundamental na comparação do custo é a relação do custo das mensagens colaborativas com as não colaborativas, para determinar se os nós colaboradores não foram mais onerados.

Como mostra o gráfico da Figura 4.17 o custo de encaminhamento da mensagens não colaborativas foi maior do que os das colaborativas em todos os casos e se manteve constante nas quatro configurações. Este resultado indica que os nós colaboradores foram menos onerados na rede, o que é bom, visto que não necessariamente serão beneficiados com estas transmissões.

4.5.4 Experimento 3

O experimento 3 (Variação do número de características no prefixo) é executado nesta topologia com a seguinte configuração:

- 5 Tmotes enviando mensagens com interesse y_1 : 1, 7, 8, 9, 11;
- 5 Tmotes com interesse em y_1 : 3, 5, 12, 15, 19;

Taxa de Mensagens Entregues em função do algoritmo de roteamento

O gráfico apresentado na Figura 4.18 mostra a relação entre a Taxa de Entrega de Mensagens e o Custo de Envio para sete configurações diferentes do protocolo REPI-A, além do algoritmo de inundação e quatro configurações do protocolo *Gossip*.

A primeira observação importante é que todas as configurações da REPI-A tiveram custo de envio inferior ao algoritmo de inundação. Esta diferença foi maior em relação a configuração com 5 campos, em que houve uma redução relativa de 13,26% no custo, com uma redução de apenas de 3,03% na \overline{TEM} .

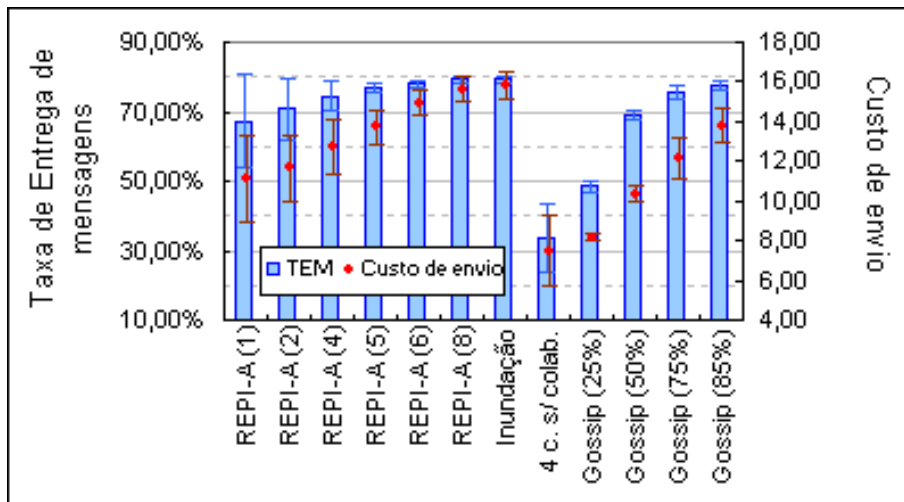


Figura 4.18: Gráfico que mostra a relação entre o Custo de Envio de Mensagens e a Taxa de Entrega de Mensagens (\overline{TEM}) de acordo com o algoritmo utilizado.

Também é possível notar que a Colaboração é fundamental para que o desempenho da REPI-A esteja no mesmo patamar dos outros protocolos estudados, visto que a \overline{TEM} na configuração sem Colaboração é de apenas 33,88%, o pior resultado obtido. Além disso, é possível notar que para uma rede com 20 nós, as configurações com REPI-A com 1 e 2 campos e *Gossip* com 25% e 50% apresentam uma baixa \overline{TEM} .

Outra característica importante que notamos nas configurações de REPI-A com 1 campo, 2 campos e 4 campos sem Colaboração é que estas apresentam alta taxa

de erro estatístico. O desvio padrão nesses três casos foram de, respectivamente, 13,33%, 8,89% e 9,83%.

Isto acontece porque nestas configurações há menos caminhos para a entrega de mensagens, visto que a probabilidade de um nó encaminhar uma mensagem é menor. Como os prefixos são randomizados no início de cada uma das vinte execuções, em algumas delas são formados canais lógicos de comunicação que não existem nas outras, fazendo com que alguns nós recebam poucas mensagens, ou então fiquem isolados de outros grupos da rede.

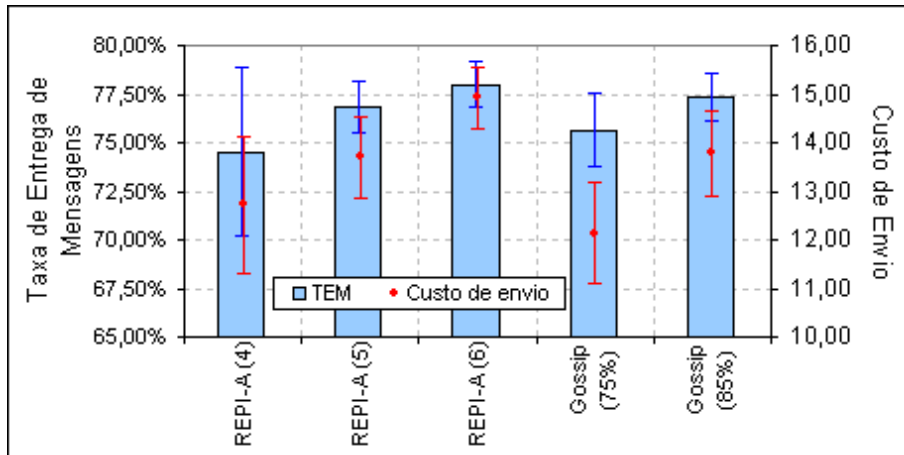


Figura 4.19: Gráfico que mostra a relação entre o Custo de Envio de Mensagens e a Taxa de Entrega de Mensagens (\overline{TEM}) de acordo com o algoritmo utilizado.

Para facilitar a análise das configurações em que tivemos os melhores resultados, o gráfico apresentado na Figura 4.19 mostra em detalhe a comparação das execuções com REPI-A com 4, 5 e 6 campos e *Gossip* com $P_{gossip} = 75\%$ e $P_{gossip} = 85\%$.

Como é possível notar, a configuração com 4 campos é a mais instável das cinco apresentadas, com desvio padrão de 4,37%, o que é uma desvantagem importante, visto que traz menor grau de confiabilidade para a rede.

Em relação à \overline{TEM} , é possível notar que a configuração com 6 campos tem o melhor desempenho, embora, considerando o erro estatístico, esteja no mesmo patamar das configurações com 5 campos e *Gossip* com $P_{gossip} = 75\%$ e $P_{gossip} = 85\%$. Porém, ao considerar o custo de envio, fica claro que as melhores configurações são a REPI-A com 5 campos e *Gossip* com $P_{gossip} = 75\%$. Com vantagem para a segunda, que apresenta uma \overline{TEM} 1,60% menor, mas com um custo 11,44% inferior.

Analisando as variações estatísticas do custo para os dois casos, com desvio padrão de 6,12% para a configuração 5 campos e de 8,55% para a com *Gossip* com $P_{gossip} = 75\%$, percebemos que esta diferença não é tão significativa, colocando as duas configurações em um patamar próximo. Uma das vantagens de REPI em relação a *Gossip* é que ela utiliza informações úteis para formação da rede, em nossa implementação as características e interesses do usuário, que também podem ser

usados para outros fins pelas aplicações.

Número médio de saltos

O número médio de salto determina quantos saltos entre os nós, em média, são necessários para que as mensagens da rede fossem entregues. No gráfico da Figura 4.20 é apresentado o impacto que as diversas configurações da REPI-A têm na entrega de mensagens da rede, tanto em relação as mensagens que são aceitas no destino quanto ao total de mensagens, incluindo as que são descartadas.

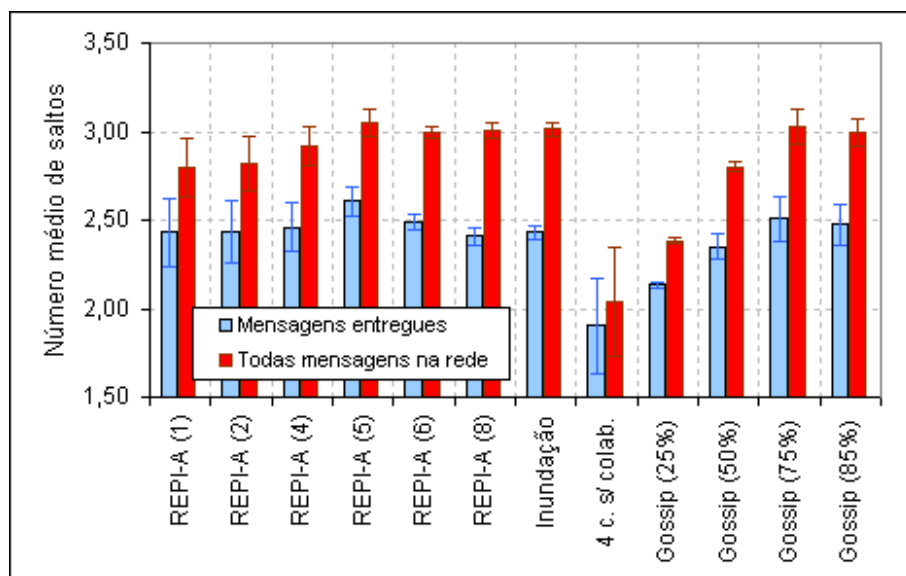


Figura 4.20: Gráfico que mostra o número médio de saltos de todas as mensagens e apenas das mensagens entregues aos destinatários, de acordo com o algoritmo utilizado.

Como é possível observar, em todas as configurações o número médio de saltos das mensagens aceitas é menor do que o do total de mensagens. Esse resultado já era esperado, visto que apenas a primeira cópia de cada mensagem é entregue ao destino, todas as outras são descartadas e, em função do atraso de entrega das mensagens, a probabilidade das mensagens com menor número de saltos chegar antes ao destino é maior do que as com mais saltos.

Além disso, é possível perceber que o valor do número de saltos varia pouco na maioria das configurações, com exceção das configurações de 4 campos sem Colaboração e *Gossip* com 25% e 50%.

Este resultado também era esperado, pois o número de saltos das mensagens está intimamente ligado a probabilidade de um nó encaminhar as mensagens que recebe. Como esta probabilidade é muito baixa nesta configurações, o valor é menor indicando que as mensagens são descartadas com mais frequência. Esta explicação também é válida para justificar o crescimento observado entre as configurações com 1 campo até a de 5 campos.

4.5.5 Experimento 4

O objetivo do Experimento 4 foi determinar o tempo médio de entrega das mensagens na REPI-A. Para isso, foram feitas cinco execuções de experimentos em que um nó envia mensagens em redes com quatro, oito, doze, dezesseis e vinte nós. As quatro primeiras configurações são mostradas na Figura 4.21, a configuração com os vinte nós na rede é semelhante as outras apresentadas na Topologia 2.



Figura 4.21: Representação de quatro configurações do Experimento 4. Na configuração mostrada em 1 há quatro nós na rede, em 2 há oito nós , doze na 3 e dezesseis nós na 4.

As medições são realizadas de duas formas: com e sem o *overhead* da aplicação mensageira. Na medição sem o *overhead*, o cálculo do tempo é feito no próprio Tmote; na medição com *overhead*, o cálculo é feito pela própria aplicação mensageira.

A avaliação sem *overhead* da aplicação mensageira é importante, pois grande parte do *overhead* que ela traz são decorrentes da instrumentação necessária para a avaliação da REPI-A e do tempo da informação ser entregue e retornar através da porta USB da máquina. Com isso, acreditamos que essa medição é mais próxima a que seria observada em uma versão comercial da aplicação executando em um equipamento de comunicação com maior integração, como um aparelho celular.

O primeiro detalhe importante observado no gráfico da Figura 4.22 é que na configuração sem *overhead* o valor do tempo médio de entrega varia muito pouco, entre $19,01 \text{ ms} \pm 5,78$ na configuração com 16 nós, até $19,45 \text{ ms} \pm 5,77$, na configuração com 4 nós. Isto acontece porque há menos congestionamento, pois o processamento

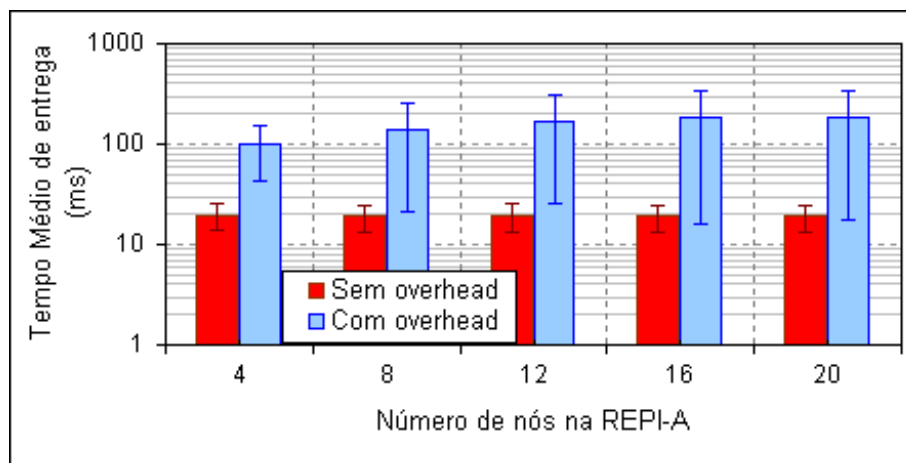


Figura 4.22: Gráfico que mostra a variação do Tempo Médio de Entrega de mensagens na REPI-A em função do número de nós, em medições com e sem o *overhead* da aplicação mensageira.

das mensagens nos Tmotes é rápido e as mensagens passam pouco ou nenhum tempo em *buffers* aguardando o envio.

Por outro lado, na configuração com o *overhead* da aplicação mensageira, também mostrada no gráfico da Figura 4.22, o tempo da mensagem trafegar pela porta USB, ser processada pela aplicação e passar por todos os *buffers* de sistema operacional gera contenções que fazem com que o valor varie de acordo com o número de nós na rede.

Esta variação é acentuada em especial entre a configuração com quatro nós, com atraso de $98,47 \text{ ms} \pm 55,90$, e dezesseis nós, com atraso de $181,46 \text{ ms} \pm 165,23$. Entre a configuração com dezesseis e vinte nós não houve variação significativa, com diferença de $0,46 \text{ ms}$.

Como a média de saltos observado para as mensagens entregues foi de cerca de 2,5 saltos nas melhores configurações estudadas, concluímos que o atraso médio de entrega no pior caso seria de cerca de $453,65 \text{ ms}$, o que consideramos satisfatório para esta classe de aplicações.

4.5.6 Discussão dos resultados da Topologia 2

Através da análise dos resultados obtidos no Experimento 1 constatamos que o congestionamento de mensagens causa perdas significativas na Taxa de Entrega de Mensagens. Para minimizar esse problema, aumentamos o intervalo de envio de mensagens e, dentre os valores avaliados, o de $[5;60]$ segundos apresentou o melhor desempenho.

Em função da natureza das aplicações mensageiras que propomos para a REPI-A, em que as mensagens precisam da interação do usuário para serem enviadas, acre-

ditamos que este tempo é satisfatório. Além disso, esse problema está diretamente ligado a limitações do padrão 802.15.4/Zigbee, como mostram outros trabalhos [46]. Com isso, acreditamos que a REPI-A pode se beneficiar de melhorias de versões futuras dele ou até mesmo de implementações para outros tipos de dispositivos com outros padrões de comunicação.

Já no experimento 2, verificamos que o número de nós destinatários tem pouco impacto na Taxa de Entregas em uma REPI-A. Pela forma como a REPI-A é endereçada, o aumento do número de nós destinatários indica que um determinado interesse se tornou popular e muitos participantes da rede estão enviando mensagens com ele. Por isso, essa avaliação é importante pois mostra que a popularidade de um dado interesse não impacta na transmissão.

Outra constatação importante do Experimento 2 foi que o custo de envio das mensagens colaborativas foi inferior aos das mensagens que foram encaminhadas por interesse. Isso indica que os nós colaboradores foram menos onerados na rede e, como eles encaminham mensagens apenas para colaborar com a rede, sem garantias de recompensação, é importante que os seus recursos não sejam muito consumidos. Porém, ainda assim, acreditamos que é importante que mecanismos de incentivo ou limitação da Colaboração sejam implantados, de forma a garantir compensações para os participantes que colaboram com a rede.

Enquanto isso, no Experimento 3, constatamos que a proposta da REPI-A obteve desempenho comparável a outras propostas da literatura, como o algoritmo de inundação e *Gossip*, com a vantagem de utilizar informações úteis para a formação da rede.

Finalmente, no Experimento 4, constatamos que o tempo médio de envio das mensagens da REPI-A sem considerar o *overhead* da aplicação mensageira é constante em uma rede com até vinte nós. Já ao considerar o *overhead*, este valor cresceu entre as configurações com um e dezesseis nós e ficou estável na comparação da configuração com dezesseis e vinte nós. Isso indica que em uma versão comercial da REPI-A a utilização de sistemas de comunicação integrados, como celulares ou PDAs pode melhorar o tempo de resposta da aplicação, especialmente para a formação de redes grandes, onde a mensagem precisa de múltiplos saltos para chegar ao seu destinatário.

4.6 Topologia 3

Nas Topologias 1 e 2, avaliamos cenários em que havia grande concentração de nós em uma mesmo ambiente. Este cenário se aproxima de situações de aglomerações de pessoas, como eventos esportivos, congressos e transportes públicos.

Na Topologia 3 buscamos avaliar um outro de tipo de cenário, em que existem

grupos de nós que se comunicam entre si mas possuem apenas um nó que é capaz de enviar mensagens para outros grupos. Este tipo de situação é previsto nas topologias básicas de rede Ad Hocs [50] e os nós que se comunicam com os outros grupos são chamados de Pontes (*Bridges*).

Com esta topologia, buscamos observar qual seria o comportamento da REPI-A em um ambiente em que a comunicação entre o nó fonte e os destinatários depende de poucos ou apenas um caminho.

Para isso, os vinte nós foram distribuídos em vários ambientes do Laboratório de Computação Paralela da COPPE/UFRJ, como mostra a Figura 4.23. Nela estão representados a distribuição dos nós em quatro ambientes, com grupos de quatro até seis nós.

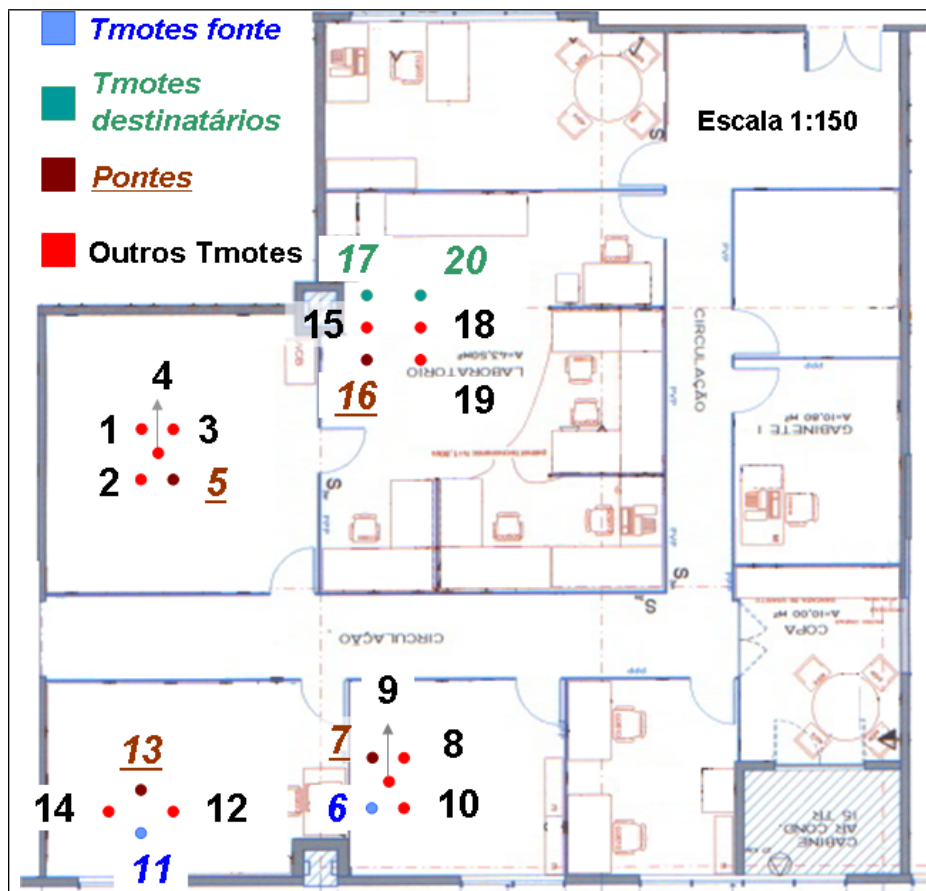


Figura 4.23: Mapa com a distribuição espacial dos Tmotes na topologia 3.

Para avaliar a capacidade da REPI-A de realizar a entrega através de múltiplos saltos passando pelos diversos grupos, dois nós foram configurados como nós fonte e dois nós no extremo oposto da rede foram configurados como destinatários. Os nós de cada grupo foram configurados com o potência 2, dos trinta e dois níveis oferecidos pelo transceptor do Tmote Sky. Nesta configuração a atenuação de transmissão é de -28,7 dbm, o que faz com que seu transceptor fosse capaz de enviar mensagens para outros nós num raio de cerca de um metro.

Já os nós ponte foram configurados com potência 4, com cerca de -21,8 dbm de atenuação. Isso garante um raio de transmissão de cerca de sete metros, para que eles fossem capaz de atingir pelo menos um dos outros grupos, de forma a forçar a transmissão por múltiplos saltos.

Como o comportamento da rede em cenários de grande congestionamento e os efeitos da Colaboração já haviam sido avaliados na Topologia 1 e 2, na Topologia 3 realizamos apenas o Experimento 3 (variação do número de características no prefixo). O objetivo foi comparar o desempenho das melhores configuração da REPI-A obtidas na Topologia 2 com os protocolos *Gossip* e inundação nesta topologia.

4.6.1 Configuração do Experimento

O Experimento 3 foi executado na topologia com dois nós fontes e dois destinatários. Os nós foram escolhidos de forma a forçar o máximo de saltos possíveis entre as fontes e os destinatários. Desta forma, foram selecionados dois nós de dois grupos distintos como os nós fontes: os nós 6 e 11, como mostra a Figura 4.23. Os nós destinatários foram os de números 17 e 20, que eram os dois mais distantes dos nós fontes.

Em todas as vinte execuções foram enviadas cem mensagens com intervalo de envio de [5;60] segundos, a melhor configuração obtida na Topologia 2.

4.6.2 Resultados

A seguir apresentamos a comparação da Taxa de Entrega de Mensagens e do Custo de Envio de Mensagens com as melhores configurações observadas na Topologia 2.

Os resultados desta avaliação são mostrados no gráfico da Figura 4.24.

Erro estatístico

A primeira observação importante é que, conforme aconteceu com as execuções na Topologia 2, percebemos que as configurações com menos campos no prefixo C apresentam maior variação de erro estatístico.

Nesta Topologia esta situação se agrava, pois como há poucos caminhos para que a mensagem trafegue, quando um dos nós ponte não encaminha as mensagens, não há outros caminhos para que ela seja entregue e aumenta a probabilidade de das mensagens não chegarem aos destinatários.

O gráfico da Figura 4.25 mostra os detalhes desta situação. Ele mostra que, por exemplo, na configuração da REPI-A com 4 campos temos uma execução em que apenas 163 mensagens das quatrocentas esperadas (cem mensagens inseridas por dois nós fontes para dois destinatários) foram entregues efetivamente. Enquanto que com a mesma configuração temos execuções em que 334 mensagens são entregues.

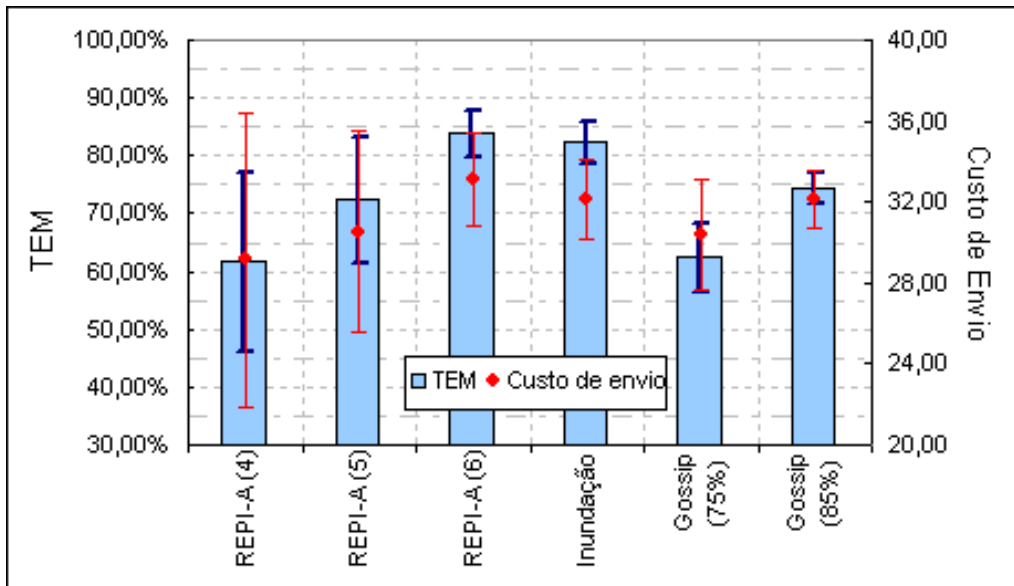


Figura 4.24: Gráfico que mostra a comparação da variação da Taxa de Entrega de Mensagens e do Custo de Entrega de Mensagens com três configurações diferentes da REPI-A, duas configurações do protocolo *Gossip* e o protocolo de inundação.

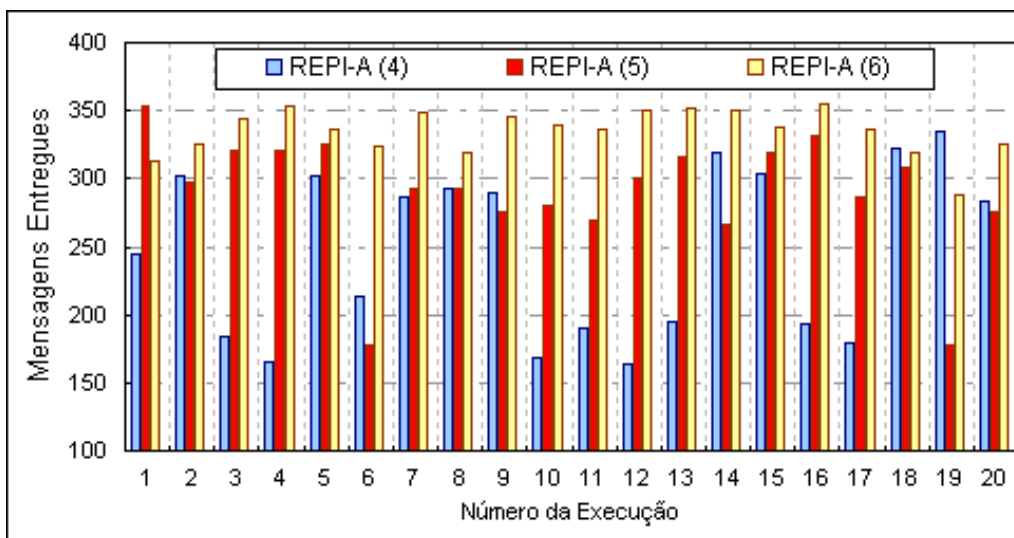


Figura 4.25: Gráfico que mostra a quantidade de mensagens entregues em todas as execuções para as três configurações analisadas.

Essa situação se repete na configuração com 5 campos, que varia entre 178 e 354 mensagens, porém ela é amenizada na configuração com 6 campos, onde a variação passa a ser entre 288 e 355 mensagens.

Esta situação sugere a necessidade de que a REPI-A deve possuir mecanismos adaptativos para reagir a esse tipo de problema, em que, em função do não atendimento dos critérios de encaminhamento, um determinado nó fique isolado na rede. A formulação destes mecanismos está fora do escopo deste trabalho.

Taxa de Entrega de Mensagens e Custo de Envio

Considerando as barras de erro, percebemos que nesta topologia a Taxa de Entrega de Mensagens das configurações REPI-A com 6 campos e flooding tiveram os melhores desempenhos, variando a \overline{TEM} entre cerca de 80% e 85%. Enquanto a configuração Gossip com $P_{gossip} = 85\%$ e REPI-A com 5 campos estavam em um patamar próximo, com média em cerca de 75%, com a configuração com 5 campos com grande variação, com desvio padrão de 10,86%.

Ao considerar o Custo de Envio, a situação se repete com a configuração com 6 campos com o maior custo médio, seguida pelo algoritmo de inundação e *Gossip* 85%, porém com os três dentro da mesma faixa de erro estatístico.

Portanto, ao considerar a estabilidade, a Taxa de Entrega e o Custo de Envio, concluímos que nesta topologia as melhores configurações foram REPI-A com 6 campos, Gossip 85% e o algoritmo de inundação.

Discussão dos resultados

Os resultados desses experimentos comprovaram que a REPI-A também é capaz de ter desempenho similar aos algoritmos de inundação e *Gossip* em uma topologia em que há poucos caminhos para a propagação das mensagens. Porém, como foi já observado, estes resultados reforçam mais uma vez a necessidade da inserção de estratégias adaptativas no protocolo REPI-A.

Os principais objetivos dessas estratégias são aumentar a taxa de entrega de mensagens, reduzir a probabilidade de ocorrência de isolamento de nós e permitir que a rede seja capaz de se configurar de acordo com a Topologia formada.

Uma possível estratégia para a resolução deste problema é a monitoração das mensagens que são descartadas em função do prefixo C. Se a quantidade de mensagens chegar a algum patamar pré-definido, o nó verifica, através de mensagens de sinalização (*beacons*), se os seus vizinhos têm interesse nas mensagens descartadas e se elas estão sendo entregues através outras fontes, de forma a verificar a ocorrência de isolamentos.

Nos casos em que o isolamento for detectado, o algoritmo pode flexibilizar suas configurações ou políticas de encaminhamento. Da mesma forma, um mecanismo pode ser usado para evitar a propagação excessiva de mensagens em uma rede com muitos caminhos.

Porém, vale ressaltar que esta é uma das possíveis soluções do problema e que a formalização destas estratégias e a avaliação do desempenho delas está fora do escopo deste trabalho.

Capítulo 5

Trabalhos Relacionados

Com a popularização dos dispositivos móveis de baixo custo, as redes Ad Hoc se tornaram um campo fértil para estudos acadêmicos. Apesar dos vários avanços obtidos, ainda não existe um padrão dominante para o roteamento de mensagens neste tipo de rede, assim como ainda existem poucas aplicações que explorem todo o potencial que ela oferece.

Neste Capítulo, apresentamos alguns dos principais trabalhos que tratam do roteamento em redes Ad Hoc e que são relacionados a proposta da REPI-A. Além disso, apresentamos algumas aplicações para construção de redes sociais em redes Ad Hoc. É importante frisar que os trabalhos aqui apresentados representam apenas uma pequena fração dos trabalhos existentes.

5.1 Algoritmos de roteamento para redes Ad Hoc

Por ser um tipo de rede volátil, em que os nós podem entrar e sair constantemente e que as mensagens trafegam em um meio não confiável, o roteamento de pacotes em redes Ad Hoc é um tema desafiante e ainda é uma questão em aberto.

Nos últimos anos, vários trabalhos apresentaram propostas para resolver este problema e, embora alguns tenham se destacado, nenhum deles conseguiu se consolidar como uma solução definitiva e, em sua maioria, os algoritmos são adequados para classes específicas de aplicações.

Ao tratar de algoritmos de roteamento para redes Ad Hoc, podemos dividi-los em duas categorias: os pró-ativos, onde existem tabelas de roteamento que são atualizadas e mantidas constantemente; e os reativos, também chamados de sob demanda, em que a rota de entrega do pacote é obtida no momento de sua entrega, de acordo com a demanda da rede.

Os algoritmos pró-ativos [51–54], apresentam um *overhead* maior em redes com pouco tráfego, pois mesmo quando não há comunicação da aplicação, existe troca de dados constante para a manutenção das tabelas de roteamento. Porém, em redes

com muitas trocas de mensagens e com pouca volatilidade, este *overhead* passa a ser desprezível e eles apresentam bom desempenho em comparação com outros tipos de algoritmos.

Já os algoritmos reativos são mais recomendados para redes em que há muita volatilidade, pois neste tipo de ambiente as tabelas de roteamento mudam com frequência e o esforço para mantê-las atualizadas geram *overhead* significativos. A REPI-A é um protocolo reativo e por isso avaliaremos apenas trabalhos que estão nesta categoria.

Nas próximas três Seções são apresentados com detalhes os três trabalhos mais relevantes na literatura de algoritmos de roteamento reativo para redes Ad Hoc: AODV [55], DSR[56] e *Gossip*[57]. Além de mostrar as propriedades de cada um destes algoritmos, também discutimos como seria viável a implementação por interesses em cada um deles, de forma a compará-los com a proposta da REPI-A.

Outros trabalhos podem ser consultados em [58–65]

5.1.1 *Ad hoc On-demand Distance Vector*

O AODV [55] (*Ad hoc On-demand Distance Vector*) é um dos protocolos de roteamento para redes Ad Hoc mais citados na literatura. Ele é projetado para funcionar em redes de pequeno e grande porte e oferece comunicação individual (*unicast*) ou em grupos (*multicast*). O seu principal objetivo é garantir as seguintes características à rede onde é implementado: rápida adaptação as condições dinâmicas dos enlaces, baixo processamento, reduzida taxa de utilização da rede, garantia de ausência de *loops* de rotamento e escalabilidade para redes com até milhares de nós.

O AODV conecta a rede através de rotas pré-definidas, que são criadas e mantidas de acordo com a demanda por comunicação na rede. Essas rotas são construídas em ciclos de envio de mensagens de requisição e rotas (RREQ) e respostas de requisição de rotas (RREP).

Quando um nó deseja enviar uma mensagem na rede e não possui uma rota até o destino, ele envia um pacote RREQ por broadcast. Os nós que recebem esta mensagem atualizam suas tabelas com informações do nó de origem e colocam uma entrada para ele em suas tabelas de roteamento. Nesta tabela estão informações como o endereço IP do nó, o ID do broadcast, número sequencial do nó origem e o número sequencial do destino final mais recente que o nó origem conhece.

Um nó que recebeu um RREQ envia um RREP em duas situações: se ele for o nó destino ou se ele tiver uma rota para o nó destino que tenha número sequencial igual ou maior ao que está na RREQ. Neste caso, o RREQ é devolvido ao nó origem através de unicast, em caso contrário, o RREQ é retransmitido através de broadcast até que a rota seja encontrada.

Para evitar que o algoritmo se torne um flooding, cada nó mantém o registro do IP de origem e os IDs de cada broadcast e, caso recebam um RREQ que já foi processada, ela é descartada imediatamente.

O processo de transmissão por multicast é similar : um nó que deseja se juntar a um grupo envia um RREQ com o IP ajustado para o grupo de multicast e com o *flag* “J” (*join*), que indica sua intenção de se juntar ao grupo. Qualquer nó que receba este pacote e que seja um membro da árvore de multicast pode enviar um RREP. Quando o pacote com a resposta de RREP é enviado, todos os nós no caminho atualizam suas tabelas de roteamento multicast. Depois de receber todos os pacotes RREP, o nó origem escolhe aquele que tiver o maior número sequencial e menor número de saltos até o destino.

Após o prazo estabelecido para a descoberta de rotas, o nó de origem irá enviar um pacote de ativação de multicast (MACT - *Multicast Activation*) para o próximo nó escolhido. Todos os nós que tenham criado entradas em suas tabelas de roteamento multicast e que não tenham recebido o pacote com o MACT irão excluir estas entradas de suas tabelas.

Comparação com a REPI

O AODV é um algoritmo para roteamento de mensagens em redes Ad Hoc onde os nós precisam conhecer qual dos seus vizinhos tem a melhor rota para o destinatário da mensagem. Ao contrário da REPI-A, o AODV é projetado para redes com identificação, onde os nós fontes precisam necessariamente conhecer os nós destinatários, o que dificulta a comunicação por interesses.

Como possui comunicação de grupo, através de *multicast*, a comunicação por interesse poderia ser implementada como uma rede sobreposta ao AODV. Porém, para isso, deveriam ser desenvolvidos mecanismos para mapear os interesses de todos os participantes da rede para criar e manter os grupos de interesses através de grupos de *multicast*.

Um dos problemas desta abordagem é que a entrada e saída de nós dos grupos tende a ser altamente custosa, pois os grupos da rede teriam que ser alterados de forma homogênea. Como o controle dos grupos é distribuído, potencialmente todos os nós da rede poderiam ser afetados.

Neste sentido, a implementação de redes orientadas a interesses em AODV apresentaria uma grande desvantagem em relação a REPI-A, pois os grupos de interesses deveriam ser fortemente acoplados, com manutenção custosa, o que pode ser problemático em uma rede que é volátil por definição.

5.1.2 DSR

O DSR [56] (*Dynamic Source Routing*), assim como o AODV, é um protocolo de roteamento para redes Ad Hoc que atua sob demanda, construindo rotas apenas quando há pacotes circulando na rede. Mas, ao contrário do AODV, no DSR a tabela de roteamento completa é determinada pelo nó remetente, que insere em cada pacote o caminho completo por onde ele deverá trafegar para chegar ao seu destino final.

Quando um nó deseja enviar mensagens para outro em um rede com DSR, sua primeira ação é consultar o seu cache de roteamento, para verificar se já existe alguma rota definida para aquele nó. Caso não exista, o algoritmo inicia o processo de descobrimento de rota. Neste processo, o nó que vai enviar o pacote é conhecido como iniciador (*initiator*) e o nó destino da mensagem é conhecido como o alvo (*target*) da descoberta de rota.

Para iniciar a descoberta de rotas o nó iniciador envia uma requisição de rota através de broadcast para todos os nós no alcance do seu rádio. Cada requisição de rota possui os seguintes campos: identificação do nó iniciador e do nó alvo, um registro de rota, onde todos os endereços da rota são registrados, e um identificador único, determinado pelo iniciador.

Ao receber uma requisição de rota, o nó verifica os dados da requisição: endereço do iniciador e do alvo, além da identificação da requisição; caso já tenha processada essa requisição, o pacote é descartado. Caso contrário, ele verifica se já possui alguma rota para o nó alvo em seu cache.

Caso possua uma rota, ela é enviada de volta para o nó iniciador através de uma resposta de rota; caso contrário, ele adiciona o seu endereço na lista de nós do registro de rota e reencaminha o pacote através de broadcast para seus vizinhos, além de registrar os dados desta requisição para que ela não seja reprocessada no futuro.

Quando a requisição é entregue no nó alvo, este deve emitir uma resposta para o nó iniciador para reportar a existência da rota. Para isso, ele procura uma rota para o nó iniciador em seu cache e, caso não exista, a rota da requisição é invertida e o pacote retorna pelo mesmo caminho por onde veio. Ao receber a resposta da formação da rota com o nó alvo, o nó iniciador já pode começar a enviar outros pacotes.

Durante a transmissão de mensagens em um rota já estabelecida, todos os nós confirmam o recebimento dos pacotes enviados para o seu nó sucessor imediato na rota. Caso exista falha de comunicação em algum dos nós, o nó que detectou o problema envia uma mensagem de erro de rota para o nó iniciador. O nó iniciador remove a rota de seu cache e verifica se conhece outras rotas disponíveis para aquele

nó. Caso não exista, o processo de descoberta de rotas é reiniciado.

Comparação com a REPI-A

Os problemas enfrentados na implementação do modelo de endereçamento por interesses no algoritmo AODV são potencializados no algoritmo DSR. Pois, como este algoritmo não apresenta a capacidade de comunicação em grupo, todos os grupos de interesses teriam que ser mapeados em todos os nós da rede, de forma distribuída.

Além disso, o DSR é um protocolo que precisa ter o conhecimento completo de todos os nós da rede por onde a mensagem deve passar até o seu destino. Em uma rede endereçada por interesse esse tipo de controle se torna complexo pois, no limite, um nó pode ter que ter conhecimento de rotas para todos os nós da rede.

Outro problema complexo seria a manutenção desses grupos, pois a simples alteração do interesse de um nó teria que ser refletida em todos os outros nós do grupo. Como cada nó possui o conhecimento da rota completa até o destinatário, a saída de um nó da rede pode comprometer várias delas, o que também causaria prejuízos para a rede.

5.1.3 Gossip

O *Gossip* [57] é uma técnica de roteamento probabilístico, em que a comunicação de todos os nós é feita através da retransmissão de pacotes através de broadcast. A principal diferença desta estratégia para um algoritmo de *flooding* é que há uma probabilidade $P_{gossip} < 1$ de transmitir cada pacote, com o objetivo é limitar o número de mensagens na rede.

O principal argumento da estratégia do *Gossip* é que, como já foi demonstrado [66], os algoritmos de *flooding* não são eficientes, pois o número de mensagens na rede cresce exponencialmente, gerando uma grande quantidade de pacotes redundantes e contenção na rede. E, ao mesmo tempo, os algoritmos determinísticos, que precisam montar um grafo da rede para transmitir através de rotas pré-definidas, como o AODV e o DSR, possuem muito *overhead* e se tornam ineficientes em redes voláteis.

Além disso, em redes determinísticas há uma tendência de que os nós que participam das rotas mais ativas sofram com maiores gastos de bateria e acabem tendo maior probabilidade de falha. Embora existam técnicas de balanceamento de carga que podem evitar este tipo de situação, elas representam um *overhead* ainda maior para a rede.

O maior desafio em uma rede *Gossip* é que a escolha correta de P_{gossip} é um problema complexo que depende de algumas variáveis dinâmicas como a topologia, número de vizinhos e qualidade do enlace de comunicação. Quando ele é escolhido corretamente, há alta probabilidade de toda a rede receber os pacotes transmitidos,

mesmo que eles tenham sido encaminhados apenas por um subgrupo probabilístico dos nós da rede. Mas se a escolha for errada e a probabilidade de transmissão for alta, a rede pode ser sobrecarregada de mensagens redundantes, como em um *flooding*; e, caso a probabilidade seja baixa, a taxa de entrega dos pacotes aos seus destinos finais será menor que a esperada.

Outro problema é que o valor correto de P_{gossip} é diretamente ligado a topologia da rede e, mesmo quando há uma escolha correta, a entrada e saída de nós pode fazer com que a eficiência da rede seja degradada. Para resolver este problema é possível adotar estratégias adaptativas, onde o valor de P_{gossip} varia de acordo com o estado atual da rede. Trabalhos como [57, 67–71] propõe versões adaptativas baseadas no *Gossip*.

Comparação com a REPI-A

Por ser um algoritmo probabilístico com transmissão por *broadcast*, o algoritmo *Gossip* é o que apresenta maiores similaridades com a proposta da REPI-A.

A principal diferença entre os dois é a forma como eles aplicam a probabilidade de transmissão. Enquanto na REPI-A a probabilidade de transmissão determina se um determinado nó irá encaminhar todas as mensagens que receber, em *Gossip* essa probabilidade é aplicada em cada mensagem enviada. Ou seja, um aplica a transmissão probabilística orientado a mensagem e o outro orientado ao nó.

A principal vantagem da abordagem da REPI-A é que os nós encaminhadores garantem o envio de todas as mensagens dos nós com quem tiveram casamento de prefixo C, e a transmissão é limitada apenas as perdas da transmissão por rádio frequência. Enquanto que, em *Gossip*, mesmo que dois nós não tenham perdas em função da transmissão por rádio frequência, nem todas as mensagens serão entregues e, caso não existam outros nós para compensar essas ausências de transmissão, a comunicação será prejudicada.

Por outro lado, a REPI-A tem a desvantagem de criar nós que, por não atender aos critérios de encaminhamento de mensagens, isolam outros nós na rede, como observado nos experimentos apresentados no Capítulo 4.

Apesar dessa diferença, a implementação das redes endereçadas por interesses em *Gossip* também poderia ser feita com uma rede sobreposta e com grupo de interesse com acoplamento fraco, como acontece em REPI-A.

5.1.4 Resumo comparativo

A tabela 5.1 apresenta um resumo comparativo das propriedades dos protocolos de roteamento apresentados nesta seção.

Tabela 5.1: Tabela de comparação dos protocolos de roteamento para redes Ad Hoc

	REPI	DSR	AODV	Gossip
Conhecimento da rede	Não é necessário	Conhece a rota completa até o destino	Conhece apenas o próximo salto da rota	Não é necessário
Identificação na rede		✓	✓	
Comunicação em grupo	✓		✓	
Grupos de interesse	✓			
Tipo de algoritmo	Probabilístico	Determinístico	Determinístico	Probabilístico
Uso do broadcast	Todas mensagens	Apenas na formação das rotas	Apenas na formação das rotas	Todas mensagens
Algoritmo adaptativo				Existem versões adaptativas

5.2 Aplicações de redes sociais para redes Ad Hoc

Nos últimos anos foram propostos vários tipos de aplicações de redes sociais para dispositivos móveis, que variam muito em seus objetivos e arquiteturas. Para melhor organizar estas aplicações, alguns trabalhos, como [72], apresentam classificações para agrupá-las de acordo com vários tipos de características. Neste trabalho, seguiremos um modelo simples, classificando as aplicações em três classes, apresentadas a seguir:

Aplicações Internet em redes móveis: nesta classe estão as aplicações que, embora sejam projetadas para redes móveis, são apenas adaptações das redes sociais tradicionais da Internet. Ou seja, adaptações do modelo clássico centralizado para dispositivos móveis, para uso em dispositivos com acesso a internet sem fio. Exemplos de aplicações desta classe são as versões para dispositivos móveis de redes sociais como o Facebook, Orkut e Myspace;

Aplicações desenvolvidas diretamente para redes móveis: Estas aplicações ignoram o legado das redes sociais da Internet e criam aplicações baseadas apenas nas capacidades oferecidas pelas redes móveis, explorando suas potencialidades. Como exemplo de aplicações desta classe podemos citar: a aplicação mensageira apresentada no Capítulo 3 e outras aplicações modeladas para REPI-A; o Nokia Sensor, apresentado na Seção 5.2.3; e o Social Serendi-

pity [73];

Aplicações Híbridas: Estas aplicações aproveitam as vantagens oferecidas pelos dois modelos anteriores, permitindo que o usuário se aproveite de sua localidade, interagindo com os usuários que estão fisicamente ao seu redor, mas também possuem conexão com a internet, para sincronizar suas bases de dados ou armazenar os dados coletados durante as experiências móveis. Como exemplo desta classe, podemos citar o MoCoTo, apresentado na Seção 5.2.1; e o mobiclique, apresentado na Seção 5.2.2.

A seguir apresentamos três aplicações de redes sociais para redes Ad Hoc que se enquadram nas classes de aplicações desenvolvidas diretamente para redes Ad Hoc e aplicações híbridas, pois esses dois tipos são mais próximas das aplicações modeladas para a REPI-A. As aplicações que são adaptações de redes sociais da Internet para dispositivos móveis não serão detalhadas neste trabalho, pois estão fora do seu escopo.

5.2.1 MoCoTo

O MoCoTo [3](Mobile Collaborative and Educational Tools) é uma arquitetura de software que oferece serviços de computação móvel para ambientes educacionais. Ele se enquadra em uma classe de aplicações chamadas de m-learning (*mobile learning*), que utiliza redes móveis em ambientes de salas de aula, para melhorar a interatividade entre os alunos e o professor. A aplicação deste tipo de tecnologia ainda é restrita a poucos centros educacionais e ainda é objeto de estudos acadêmicos [3, 25–28].

Funcionamento do MoCoTo

O MoCoTo é um conjunto de aplicações que oferece vários tipos de serviços para ambientes educacionais. Ele é construído em uma arquitetura híbrida, em que os participantes podem interagir através de redes Ad Hoc com comunicação por Bluetooth, ou através de serviços de dados da rede telefônica móvel, utilizando GPRS (*General Packet Radio Service*).

Os serviços oferecidos pelo MoCoTo são:

Chat room

Salas de bate-papo onde os alunos podem trocar mensagens. As salas são criadas pelo professor através de uma interface Web e os alunos as acessam através de celulares e smartphones, utilizando GPRS;

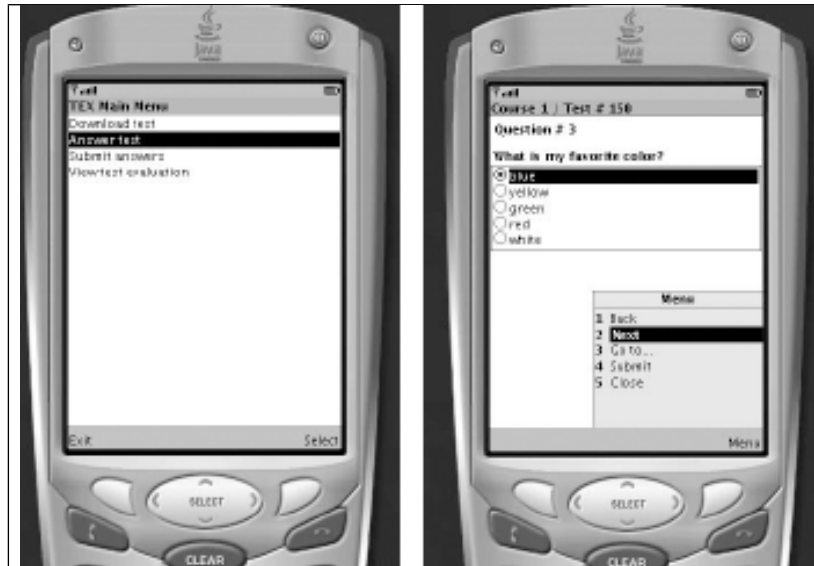


Figura 5.1: Exemplo da aplicação MoCoTo para realizar avaliações de aprendizado. Na figura da esquerda o menu principal da aplicação, no da direita uma das questões do teste, retirada de [3].

Quadro branco virtual compartilhado

Este aplicativo cria um quadro branco virtual compartilhado entre os alunos. O quadro é criado por qualquer aluno e depois ele o compartilha com seus colegas, que se conectam como clientes, através da conexão Bluetooth;

Integração em sala de aula

O objetivo deste serviço é romper barreiras comuns na comunicação entre professores e alunos, como timidez ou distância física, em treinamentos por vídeo conferência. Para isso, o MoCoTo oferece um aplicação no modelo cliente-servidor em que os alunos podem formular perguntas através de um PDA. Essas perguntas são vistas pelo professor e pelos outros alunos e são classificadas em um ranking. Quando um aluno também tem interesse em uma questão, ele vota nela e ela é melhor classificada, podendo ser priorizada pelo professor;

Gerenciamento de testes

Este é um serviço para o professor criar avaliações de aprendizado para os alunos. O professor pode criar vários tipos de questões: múltipla escolha, verdadeiro ou falso, parágrafos, respostas curtas, combinação, múltiplas questões e preencher campos em branco. E os alunos acessam os testes em seus celulares em uma aplicação J2ME nos seus celulares através da internet.

Avaliação do MoCoTo

O MoCoTo oferece uma série de aplicações móveis para aprimorar a comunicação em ambientes educacionais, oferecendo ferramentas para os alunos e professores

superarem barreiras comuns no processo de aprendizagem. Apesar da abrangência do trabalho, através de quatro ferramentas, ele ainda é excessivamente dependente da infraestrutura fixa das operadora de celular.

Além disso, a única ferramenta que pode ser usada através de uma rede Ad Hoc, o quadro branco virtual, utiliza tecnologia Bluetooth, o que permite que apenas oito alunos compartilhem o quadro, o que pode ser uma grande limitação em salas de aulas grandes. Outro problema é a ausência de criação de redes grandes, com roteamento e entrega de pacotes com saltos múltiplos.

5.2.2 MobiClique

O MobiClique [4] é uma aplicação que permite a integração de redes sociais online (OSN - *Online Social Network*) com redes Ad Hoc. Na versão apresentada em [4] o MobiClique usa os dados da conta do Facebook do usuário para realizar interações com outros usuários através de Bluetooth ou WiFi.

Funcionamento do MobiClique

O MobiClique funciona como uma extensão do FaceBook para o mundo físico. Nesta rede, o usuário utiliza as informações cadastradas em sua conta do Facebook como o perfil básico da rede MobiClique, na qual ele irá interagir com pessoas com quem se encontrar fisicamente.

Para isso, os desenvolvedores da MobiClique criaram uma arquitetura com várias aplicações. A primeira delas é uma aplicação simples que deve ser executada em um computador com acesso a internet. Ela se conecta ao Facebook e obtém informações do usuário como: identificação, nome do usuário, lista de grupos, redes e lista de amigos, e as salva em um arquivo de texto. Esse arquivo é copiado para o dispositivo do usuário através de uma sincronização do dispositivo com o computador do usuário.

A aplicação para ingressar na rede MobiClique foi desenvolvida em C++ e C# e roda na plataforma Windows Mobile. Ao carregar esta aplicação, o usuário lê as informações gravadas no arquivo texto criado anteriormente. A partir desse arquivo, a aplicação cria um lista que permite ao usuário escolher quais informações ele deseja disponibilizar na rede móvel. Ao terminar este processo, o usuário está pronto para ingressar na rede MobiClique.

A Figura 5.2 apresenta o funcionamento básico da rede MobiClique. Depois de cumprir os passos apresentados anteriormente, o aparelho entra em um modo de operação que é um loop com três passos:

1. Descoberta de vizinhança: nesta etapa o dispositivo busca por outros aparelhos da rede MobiClique. Na versão apresentada em [4], a conexão Bluetooth foi a

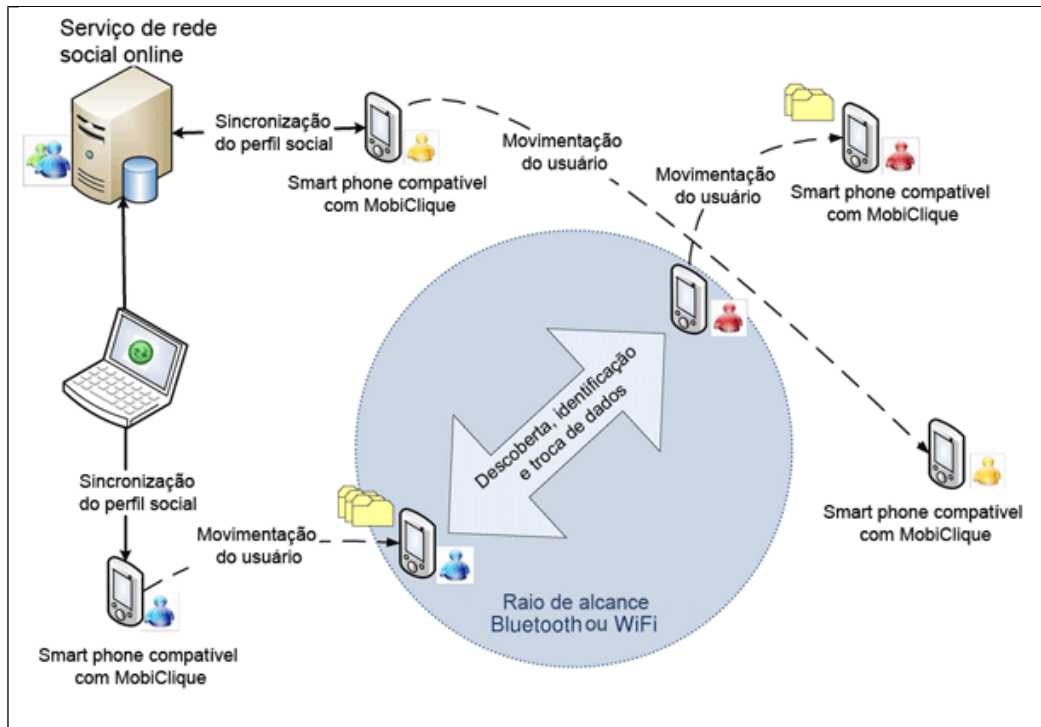


Figura 5.2: Diagrama que exhibe a formação da rede MobiClique, adaptado de [4].

utilizada em virtude do menor consumo de energia em relação a conexões Wifi. A procura por novos dispositivos é feita a cada 2 minutos durante um período de 10,24 segundos, que é a duração mínima prevista pelo padrão BlueTooth. Para reduzir a interferência, as redes podem ter no máximo 3 dispositivos de cada vez;

2. Identificação e autenticação do usuário: Após descobrir um novo usuário, o MobiClique entra na fase de identificação onde os dispositivos se conectam para trocar dados. Se este for o primeiro encontro entre os nós, o perfil completo é trocado. Nos contatos subsequentes os dados são trocados apenas se um dos perfis tiver sido alterado desde o último contato, caso contrário, apenas a identificação do usuário é trocada;
3. Troca de dados: Após a identificação do usuário, a rede MobiClique está pronta para a troca de mensagens da aplicação. As mensagens da rede MobiClique são armazenadas de forma persistente em cada dispositivo e são encaminhadas para outros dispositivos sempre que há uma oportunidade. Com isso, a rede MobiClique permite a comunicação através de múltiplos saltos, de forma assíncrona.

As mensagens da rede MobiClique podem ser direcionadas para usuários ou grupos de usuários. Ao encontrar com um usuário, é possível mandar uma mensagens

para ele diretamente, para um de seus amigos ou para um dos grupos que ele participa. Para encaminhar as mensagens para os amigos do usuário que participou da troca de dados, as mensagens são armazenadas por ele e ele a repassa para todos os amigos em comum que tiver com o destinatário e o processo é repetido até a mensagem ser entregue.

Já as mensagens para grupos são encaminhadas para todos os usuários participantes do grupo até que todos tenham recebido. As mensagens também podem ser eliminadas do sistema ao atingir um TTL (time-to-live) máximo. Esse TTL é composto de duas partes: um número de saltos que a mensagem pode dar até chegar ao destino; e uma marca de tempo (*timestamp*) que determina um horário em que o pacote deve ser descartado. O primeiro que for atingido causa a eliminação da mensagem.

Avaliação do MobiClique

O MobiClique é uma aplicação que estende as redes sociais online para ambientes de redes Ad Hoc, permitindo que seus participantes troquem mensagens individualmente ou em grupos, através de contatos oportunistas, sem estar conectado a Internet. Ele traz capacidades interessantes, como a comunicação assíncrona em grupo e a capacidade de comunicação de longo alcance, com as mensagens transitando por vários dispositivos até chegar ao destinatário final.

Apesar de ser uma proposta interessante, essa aplicação não traz alguns benefícios que consideramos importantes para o usuário de uma rede social em uma rede Ad Hoc, como por exemplo: capacidade de comunicação em tempo real com usuários no seu entorno, comunicação com pessoas desconhecidas, formação de grupos com dezenas e até centenas de usuários e anonimato.

5.2.3 Nokia Sensor

O Nokia Sensor[5] é um aplicativo criado pela Nokia, empresa finlandesa de telecomunicações. Ele foi desenvolvido para uso nos celulares da própria Nokia e permite que seus usuários estabeleçam contatos sociais espontâneos através da comunicação Bluetooth de seus celulares.

Funcionamento do Nokia Sensor

Em uma rede do Nokia Sensor, cada usuário deve criar uma página com informações pessoais, chamada de Folio. O Folio é preenchido antes do ingresso na rede e ele determina como o usuário interage nela. O Folio contém itens como foto digital do usuário, apelido e dados pessoais, como mostrado na Figura 5.3.

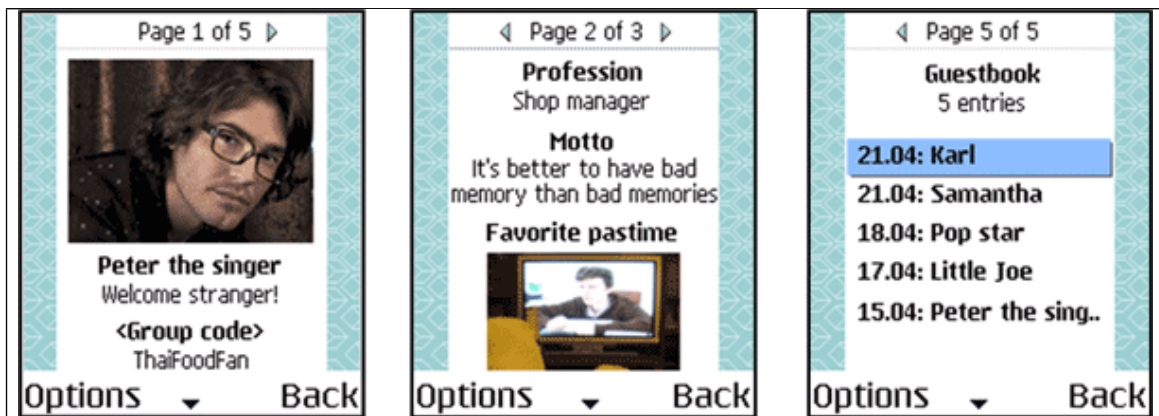


Figura 5.3: Exemplo de das informações que o usuário insere no seu perfil do Nokia Sensor, chamado Folio. Adaptado de [5].

Para ingressar na rede do Nokia Sensor, o usuário realiza um processo de busca de usuário, chamado de *Scan*. Ao concluir o processo, o usuário tem acesso a uma lista com todos os perfis encontrados no alcance do rádio Bluetooth. Ao escolher um outro perfil, o usuário pode interagir com ele de várias formas: troca de arquivos digitais, troca de cartões de visitas e mensagens de texto.

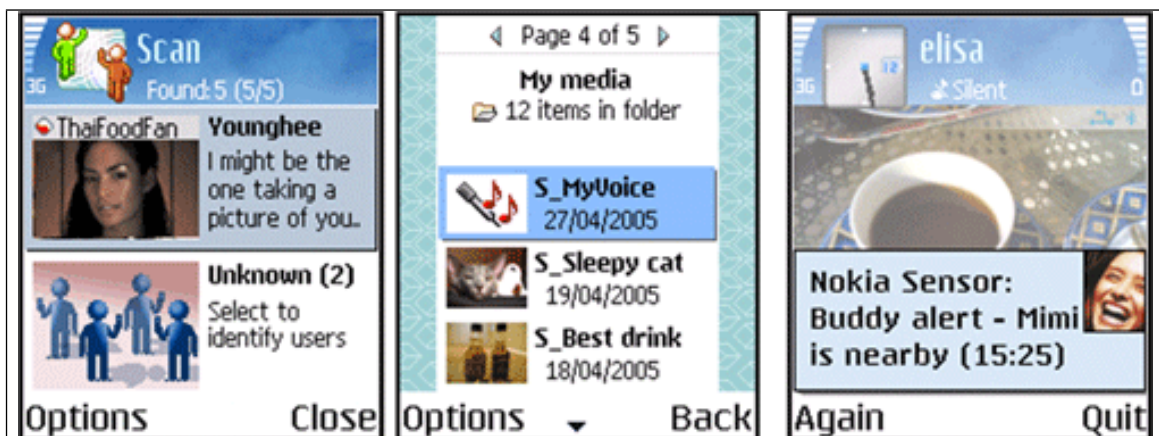


Figura 5.4: Três telas do Nokia Sensor: à esquerda a tela do resultado do processo de Scan, como cinco usuários encontrados; na Figura do meio, a pasta com os arquivos compartilhados na rede Nokia Sensor; e a direita um alerta avisando que um usuário marcado como favorito está ao alcance do rádio Bluetooth.

Além disso, o usuário possui outras opções no uso da aplicação: criar grupos de usuários favoritos; livro de visitas, onde usuários podem deixar mensagens; estatísticas de uso da aplicação, onde o usuário pode verificar quantas vezes seu perfil foi acessado e gráficos com os números de visitas por dia.

Avaliação do Nokia Sensor

O Nokia Sensor é um aplicativo para comunicação e troca de mensagens de usuários de celulares. A aplicação apresenta características interessantes, como a possibili-

dade de troca de arquivos, a criação do perfil do usuário, troca de cartões de visitas e alertas sobre a presença de outros usuários.

Uma limitação importante do Nokia Sensor é a incapacidade de formação de grupos, visto que só há comunicação par a par direta, sem roteamento de pacotes. Além disso, a falta de comunicação de longo alcance, com múltiplos saltos, limita muito a aplicação, visto que o alcance do seu rádio é de apenas 10 metros.

5.2.4 Comparativo das aplicações de redes sociais para redes Ad Hoc

A Tabela 5.2 apresenta o resumo comparativo das principais propriedades da REPI-A e dos três trabalhos de aplicações de redes sociais para redes Ad Hoc apresentados neste Capítulo.

Tabela 5.2: Quadro com o resumo comparativo dos trabalhos relacionados.

Trabalho Característica	REPI-A	MoCoTo	Mobi Cli- que	Nokia Sensor
Comunicação	ZigBee	GRPS e Bluetooth	Bluetooth	Bluetooth
Alcance do rádio	125 metros	Cobertura da opera- dora de celular	10 metros	10 metros
Troca de Mensagens	✓	✓	✓	✓
Troca de arquivos	-	-	-	✓
Formação de grupos	✓	✓	✓	-
Participantes na rede	Até 65535	Até 8	Até 3	-
Mensagens Individuais	✓	-	✓	✓
Depende de infraestrutura fixa	-	✓	✓	-
Grupos de longo alcance	✓	-	✓	-

Capítulo 6

Conclusão

O objetivo deste trabalho foi modelar, implementar, implantar e avaliar uma Rede Endereçada por Interesse em uma rede Ad Hoc, através da utilização de uma rede de sensores com vinte dispositivos. Acreditamos que estes objetivos foram atingidos pelo seguintes motivos:

- Através do desenvolvimento de uma arquitetura computacional composta pelo middleware Tmote, uma Aplicação Mensageira, o sistemas SAMCRA, todos desenvolvidos neste trabalho, e elementos de hardware desenvolvidos por terceiros, criamos a REPI-A em um ambiente real;
- A REPI-A foi avaliada através de sete métricas com mais de cinquenta configurações diferentes, além de ter seu desempenho comparado com algoritmos de inundação e *Gossip*. A partir dos resultados obtidos, comprovamos que a proposta da REPI-A é promissora e verificamos quais são as melhores configurações para a implantação de uma REPI-A em uma rede com vinte dispositivos em duas topologias típicas de redes Ad Hoc;
- A REPI-A, através da aplicação mensageira proposta neste trabalho, cria um arcabouço para o desenvolvimento de uma série de novas aplicações, que podem ser usadas para a formação de grupos de comunicação por interesses e interação social em situações do dia-a-dia. Essas aplicações tem potencial para facilitar a realização de tarefas em áreas como comércio, situações de emergência e formação de grupos de pesquisa;
- Também a partir da análise dos resultados dos experimentos realizados, identificamos a importância do desenvolvimento de mecanismos adaptativos para permitir que a REPI-A conheça o contexto da rede e possa ajustar propriedades como o tamanho do conjunto de características usado para o encaminhamento de mensagens e a política de encaminhamento;

- Através dos resultados dos experimentos foi comprovada a importância da Colaboração em uma rede endereçada por interesses, conceito criado pelo modelo associado a REPI. Além disso, o seu impacto foi estudado e identificamos que a sua importância está diretamente ligada com a popularidade dos interesses, o que também indica a importância da adoção de mecanismos adaptativos para controlar a Colaboração;
- Com os resultados obtidos na avaliação experimental, comprovamos a tese defendida por alguns autores da importância dos experimentos reais para a avaliação de protocolo de roteamento para redes Ad hoc. Foi possível comprovar o quão imprevisível é o comportamento da comunicação por rádio frequência, o que pode trazer disparidades para execuções simuladas.

Somando-se a isso, como fruto secundário deste trabalho, foi desenvolvido o Sistema SAMCRA, que é capaz de configurar, monitorar e coletar dados de execuções em redes Ad Hoc reais, implantadas com qualquer tipo de dispositivo de comunicação.

SAMCRA foi usado na execução de todos os experimentos apresentados neste trabalho e é flexível o suficiente para avaliar o desenvolvimento de outros protocolos para redes Ad Hoc. Em [29] é mostrado como SAMCRA pode ser usado por outros grupos de pesquisa.

6.1 Trabalho Futuro

Além das realizações apresentadas neste trabalho, ainda existem oportunidades de desenvolvimento de trabalhos envolvendo a REPI, REPI-A e SAMCRA. Algumas delas são:

Aprimorar o sistema SAMCRA

Através das seguintes adições:

- Criar mecanismos para a adição de novos protocolos de forma dinâmica;
- Criar uma ferramenta para editar as variáveis analisadas, de forma a facilitar o uso da ferramenta por outros grupos de pesquisa;
- Permitir a especificação do número máximo de nós em tempo de execução, através de parâmetros de inicialização da aplicação. Atualmente este valor só pode ser modificado em tempo de compilação;
- Aprimorar a configuração dos experimentos, permitindo que a configuração do intervalo e quantidade de mensagens enviadas seja definida para cada nó;

- Criar um módulo para gerar os grafos da rede automaticamente. Atualmente, são coletados dados que podem ser usados com ferramentas de geração automática de grafos, mas seria interessante que essa funcionalidade fosse agregada a SAMCRA;
- Desenvolver um módulo para execução de experimentos em lote, para que diversas configurações possam ser testadas várias vezes com apenas um comando.

Avaliar outras configurações da REPI

Através dos experimentos realizados, foi possível notar que algumas características da REPI, como, por exemplo, a Colaboração, são benéficas em algumas configurações e em outras trazem apenas *overheads*, diminuindo o desempenho e trazendo maior congestionamento na rede. Com isso, concluímos que estratégias adaptativas poderiam ser usadas para aprimorar o desempenho do protocolo. De forma que ele possa reagir aos eventos da rede e adotar as configurações que se beneficiem do contexto atual;

Avaliar a execução da REPI-A em outros cenários

A partir dos estudos realizados neste trabalho, realizar outros experimentos com a REPI-A, tanto com a adição de mais nós com a mesma arquitetura computacional, quanto em outros tipos de arquitetura. Uma das opções é a utilização de outros protocolos para o controle das camadas PHY e MAC como, por exemplo, o padrão 802.11 configurado para formação de redes Ad Hoc [74].

Simular a REPI em redes com até milhares de nós

Neste trabalho avaliamos a REPI em uma rede Ad Hoc com vinte dispositivos, porém, é importante avaliar se os resultados obtidos com esta escala também são válidos em redes com centenas e até milhares de nós. Os resultados deste trabalho podem ser usados para aferir e validar os simuladores usados nessas simulações.

Referências Bibliográficas

- [1] DE CASTRO DUTRA, R., GRANJA, R., DE MORAES, H. F., et al. “REPI: Rede de comunicação Endereçada Por Interesses”. In: *WP2P 2010: Anais do VI Workshop de Redes Dinâmicas e Sistemas P2P . Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC2010)*, may 2010.
- [2] MOTEIV. *Tmote Sky Datasheet* <http://sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf>, 2006. Acessado em junho de 2010.
- [3] LOPES, R., CORTES, O. “An Ubiquitous Testing System for m-Learning Environments”. In: *Systems and Networks Communications, 2007. ICSNC 2007. Second International Conference on*, pp. 32–32, Aug. 2007. doi: 10.1109/ICSNC.2007.20.
- [4] PIETILÄINEN, A.-K., OLIVER, E., LEBRUN, J., et al. “MobiClique: middleware for mobile social networking”. In: *WOSN '09: Proceedings of the 2nd ACM workshop on Online social networks*, pp. 49–54, New York, NY, USA, 2009. ACM. ISBN: 978-1-60558-445-4. doi: <http://doi.acm.org/10.1145/1592665.1592678>.
- [5] PERSSON, P., JUNG, Y. “Nokia sensor: from research to product”. In: *DUX '05: Proceedings of the 2005 conference on Designing for User eXperience*, p. 53, New York, NY, USA, 2005. AIGA: American Institute of Graphic Arts. ISBN: 1-59593-250-X.
- [6] ERGEN, S. C. *ZigBee/IEEE 802.15.4 Summary*. Disponível em <http://www.sinemergen.com/zigbee.pdf>. Relatório técnico, Wireless Sensor Networks Berkeley Lab, 2004. Acessado em junho de 2010.
- [7] ANATEL. *ANATEL, Relatório Anual 2008*. Disponível em <http://www.anatel.gov.br>. Relatório técnico, ANATEL, 2008. Acessado em junho de 2010.
- [8] ORKUT. <http://www.orkut.com/>, 2005. Acessado em junho de 2010.

- [9] FACEBOOK. <http://www.facebook.com/>, 2004. Acessado em junho de 2010.
- [10] TWITTER. <http://www.twitter.com>, 2006. Acessado em junho de 2010.
- [11] MYSPACE. <http://www.myspace.com>, 2003. Acessado em junho de 2010.
- [12] ALEXA. <http://www.alexa.com/>, 1996. Acessado em junho de 2010.
- [13] LIMEWIRE. <http://www.limewire.com>, 2000. Acessado em junho de 2010.
- [14] KAZAA. <http://www.kazaa.com>, 2006. Acessado em junho de 2010.
- [15] EMULE. <http://www.emule.com/>, 2002. Acessado em junho de 2010.
- [16] COHEN, B. <http://www.BitTorrent.com>, 2001. Acessado em junho de 2010.
- [17] GRISWOLD, W. G. “Five Enablers for Mobile 2.0”, *Computer*, v. 40, n. 10, pp. 96–98, 2007. ISSN: 0018-9162. doi: <http://dx.doi.org/10.1109/MC.2007.346>.
- [18] EAGLE, N., PENTLAND, A. “Social serendipity: mobilizing social software”, *Pervasive Computing, IEEE*, v. 4, n. 2, pp. 28–34, Jan.-March 2005. ISSN: 1536-1268. doi: [10.1109/MPRV.2005.37](http://dx.doi.org/10.1109/MPRV.2005.37).
- [19] COUNTS, S., TER HOFTE, H., SMITH, I. “Mobile social software: realizing potential, managing risks”. In: *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pp. 1703–1706, New York, NY, USA, 2006. ACM. ISBN: 1-59593-298-4. doi: <http://doi.acm.org/10.1145/1125451.1125767>.
- [20] LUGANO, G. “Mobile Social Software: Definition, Scope and Applications”. In: *eChallenges conference*, pp. 1434–1441, The Hague, Holland.
- [21] MUSOLESI, M., HAILES, S., MASCOLO, C. “An ad hoc mobility model founded on social network theory”. In: *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 20–24, New York, NY, USA, 2004. ACM. ISBN: 1-58113-953-5. doi: <http://doi.acm.org/10.1145/1023663.1023669>.
- [22] MOTANI, M., SRINIVASAN, V., NUGGEHALLI, P. S. “PeopleNet: engineering a wireless virtual social network”. In: *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pp. 243–257, New York, NY, USA, 2005. ACM. ISBN: 1-59593-020-5. doi: <http://doi.acm.org/10.1145/1080829.1080855>.

- [23] QUERCIA, D., CAPRA, L. “FriendSensing: recommending friends using mobile phones”. In: *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pp. 273–276. ACM, 2009. ISBN: 978-1-60558-435-5. doi: <http://doi.acm.org/10.1145/1639714.1639766>.
- [24] KOSTAKOS, V. “Social networking 2.0”. In: *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pp. 3381–3386, New York, NY, USA, 2008. ACM. ISBN: 978-1-60558-012-X. doi: <http://doi.acm.org/10.1145/1358628.1358861>.
- [25] BERGER, S., MOHR, R., NÖSEKABEL, H., et al. “Mobile Collaboration Tool for University Education”. In: *WETICE '03: Proceedings of the Twelfth International Workshop on Enabling Technologies*, p. 77, Washington, DC, USA, 2003. IEEE Computer Society. ISBN: 0-7695-1963-6.
- [26] LIU, C.-H., CHEN, W.-Y. “The effectiveness of the handheld devices based on mobiles learning”. In: *ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences*, pp. 400–405, New York, NY, USA, 2009. ACM. ISBN: 978-1-60558-710-3. doi: <http://doi.acm.org/10.1145/1655925.1655999>.
- [27] HENRY, L., SANKARANARAYANAN, S. “Application of intelligent agents for mobile tutoring”. In: *ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences*, pp. 963–969, New York, NY, USA, 2009. ACM. ISBN: 978-1-60558-710-3. doi: <http://doi.acm.org/10.1145/1655925.1656100>.
- [28] LARA, E. D., KUMAR, R., WALLACH, D. S., et al. “Collaboration and multimedia authoring on mobile devices”. In: *in Proc. of the First Intl. Conf. on MobiSys*, pp. 287–301, 2003.
- [29] GRANJA, R., DE CASTRO DUTRA, R., DE MORAES, H. F., et al. “SAM-CRA: Um sistema para avaliação experimental de Redes Ad Hoc”. In: *Salão de Ferramentas, Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC2010)*, may 2010.
- [30] OCTOPOP. <http://br.octopop.com/>, 2008. Acessado em junho de 2010.
- [31] DE CASTRO DUTRA, R., AMORIM, C. *Modelo de comunicação endereçada por interesses*. Relatório técnico, ES 733 / PESC - COPPE - UFRJ, 2010.
- [32] NUMBER, M., GOULDING, A., SPACEY, R., et al. “Java Remote Method Invocation”. In: *In 68th IFLA COuncil and General Conference*, p. 10, 2002.

- [33] ARNOLD, K., GOSLING, J., GOSLING, J., et al. “The Java Language Specification”. , 1996.
- [34] JCONFIG. <http://www.jconfig.org/>, 2000. Acessado em junho de 2010.
- [35] WHITEHEAD, E. J., MURATA, M. “XML Media Types”. . Internet RFC 2376, July 1998.
- [36] POLASTRE, J., HUI, J., LEVIS, P., et al. “A unifying link abstraction for wireless sensor networks”. In: *In Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys*, pp. 76–89. ACM Press, 2005.
- [37] MSNMESSENGER. <http://download.live.com/messenger>, 1999. Acessado em junho de 2010.
- [38] ICQ. <http://www.icq.com/>, 1996. Acessado em junho de 2010.
- [39] YAHOOMESSENGER. <http://messenger.yahoo.com/>, 1998. Acessado em junho de 2010.
- [40] KHAN, S. K. A., TOKARCHUK, L. N. “Interest-based self organization in group-structured P2P networks”. In: *CCNC’09: Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, pp. 1237–1241, Piscataway, NJ, USA, 2009. IEEE Press. ISBN: 978-1-4244-2308-8.
- [41] WANG, Y., VASSILEVA, J. “Trust-Based Community Formation in Peer-to-Peer File Sharing Networks”. In: *WI ’04: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 341–348, Washington, DC, USA, 2004. IEEE Computer Society. ISBN: 0-7695-2100-2. doi: <http://dx.doi.org/10.1109/WI.2004.144>.
- [42] BITCOMET. <http://www.bitcomet.com>, 2006. Acessado em junho de 2010.
- [43] HENDERSON, T. R., ROY, S., FLOYD, S., et al. “ns-3 project goals”. In: *WNS2 ’06: Proceeding from the 2006 workshop on ns-2: the IP network simulator*, p. 13, New York, NY, USA, 2006. ACM. ISBN: 1-59593-508-8. doi: <http://doi.acm.org/10.1145/1190455.1190468>.
- [44] ZENG, X., BAGRODIA, R., GERLA, M. “GloMoSim: a library for parallel simulation of large-scale wireless networks”, *SIGSIM Simul. Dig.*, v. 28, n. 1, pp. 154–161, 1998. ISSN: 0163-6103. doi: <http://doi.acm.org/10.1145/278009.278027>.

- [45] CAVILLA, A., BARON, G., HART, T., et al. “Simplified simulation models for indoor MANET evaluation are not robust”. In: *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 610 – 620, oct. 2004. doi: 10.1109/SAHCN.2004.1381964.
- [46] ZHAO, J., GOVINDAN, R. “Understanding packet delivery performance in dense wireless sensor networks”. In: *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 1–13, New York, NY, USA, 2003. ACM. ISBN: 1-58113-707-9. doi: <http://doi.acm.org/10.1145/958491.958493>.
- [47] RUSSINOVICH, M. “PsTools v2.43”. . <http://www.microsoft.com/brazil/technet/sysinternals/SystemInformation/PsTools.msp>, 2007. Acessado em junho de 2010.
- [48] NOVATNACK, J., GREENWALD, L., ARORA, H. “Evaluating ad hoc routing protocols with respect to quality of service”. In: *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, v. 3, pp. 205 – 212 Vol. 3, aug. 2005. doi: 10.1109/WIMOB.2005.1512905.
- [49] HARTUNG, C., HAN, R., SEIELSTAD, C., et al. “FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments”. In: *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, pp. 28–41, New York, NY, USA, 2006. ACM. ISBN: 1-59593-195-3. doi: <http://doi.acm.org/10.1145/1134680.1134685>.
- [50] JUNG, S., CHANG, A., GERLA, M. “Comparisons of ZigBee Personal Area Network (PAN) Interconnection Methods”. In: *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on*, pp. 337–341, 17-19 2007. doi: 10.1109/ISWCS.2007.4392357.
- [51] PERKINS, C. E., BHAGWAT, P. “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers”, *SIGCOMM Comput. Commun. Rev.*, v. 24, n. 4, pp. 234–244, 1994. ISSN: 0146-4833. doi: <http://doi.acm.org/10.1145/190809.190336>.
- [52] PARKER, T., LANGENDOEN, K. “Guesswork: robust routing in an uncertain world”, *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, v. 0, pp. 699, 2005. doi: <http://doi.ieeecomputersociety.org/10.1109/MAHSS.2005.1542860>.

- [53] PEI, G., GERLA, M., HONG, X., et al. “Wireless Hierarchical Routing Protocol with Group Mobility (WHIRL)”. In: *In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1538–1542, 1999.
- [54] MURTHY, S., GARCIA-LUNA-ACEVES, J. J. “An efficient routing protocol for wireless networks”, *Mob. Netw. Appl.*, v. 1, n. 2, pp. 183–197, 1996. ISSN: 1383-469X. doi: <http://dx.doi.org/10.1007/BF01193336>.
- [55] PERKINS, C., BELDING-ROYER, E., DAS, S. “Ad hoc On-Demand Distance Vector (AODV) Routing”. . RFC 3561, 2003. Disponível em: <http://www.ietf.org/rfc/rfc3561.txt>.
- [56] JOHNSON, D. B., MALTZ, D. A., BROCH, J. “DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks”. In: *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pp. 139–172. Addison-Wesley, 2001.
- [57] HAAS, Z., HALPERN, J., LI, L. “Gossip-based ad hoc routing”. In: *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, v. 3, pp. 1707–1716 vol.3, 2002. doi: 10.1109/INFCOM.2002.1019424.
- [58] HESTER, L., HUANG, Y., ANDRIC, O., et al. “NeuRFon netform: a self-organizing wireless sensor network”. In: *Proceedings of the Eleventh International Conference on Computer Communications and Networks*, pp. 364–369, 2002. ISBN: 0-7803-7553-X.
- [59] CHIN, K.-W., LOWE, D., LAU, W. H. O. “Routing in MANETs with Address Conflicts”. In: *MOBIQUITOUS '05: Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 225–236, Washington, DC, USA, 2005. IEEE Computer Society. ISBN: 0-7695-2375-7. doi: <http://dx.doi.org/10.1109/MOBIQUITOUS.2005.51>.
- [60] SABEUR, M., ALSUKKAR, G., JOUABER, B., et al. “A new Routing & Mobility Management Solution for Wireless Mesh Network”. In: *MOBIQUITOUS '07: Proceedings of the 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*, pp. 1–3, Washington, DC, USA, 2007. IEEE Computer Society. ISBN: 978-1-4244-1024-8. doi: <http://dx.doi.org/10.1109/MOBIQ.2007.4450990>.

- [61] HUANG, C.-F., TSENG, Y.-C., WU, H.-L. “Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network”, *ACM Trans. Sen. Netw.*, v. 3, n. 1, pp. 5, 2007. ISSN: 1550-4859. doi: <http://doi.acm.org/10.1145/1210669.1210674>.
- [62] JUNG, S., CHANG, A., GERLA, M. “Temporary Interconnection of ZigBee Personal Area Network (PAN)”. In: *MOBIQUITOUS '07: Proceedings of the 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*, pp. 1–5, Washington, DC, USA, 2007. IEEE Computer Society. ISBN: 978-1-4244-1024-8. doi: <http://dx.doi.org/10.1109/MOBIQ.2007.4451010>.
- [63] WHITEHOUSE, K., SHARP, C., BREWER, E., et al. “Hood: a neighborhood abstraction for sensor networks”. In: *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pp. 99–110, New York, NY, USA, 2004. ACM. ISBN: 1-58113-793-1. doi: <http://doi.acm.org/10.1145/990064.990079>.
- [64] MELAMED, R., KEIDAR, I., BAREL, Y. “Octopus: a fault-tolerant and efficient ad-hoc routing protocol”, *Wirel. Netw.*, v. 14, n. 6, pp. 777–793, 2008. ISSN: 1022-0038. doi: <http://dx.doi.org/10.1007/s11276-006-0013-6>.
- [65] SAMPATH, D., GARCIA-LUNA-ACEVES, J. J. “PROSE: scalable routing in MANETs using prefix labels and distributed hashing”. In: *SECON'09: Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 601–609. Institute of Electrical and Electronics Engineers Inc., The, 2009. ISBN: 978-1-4244-2907-3.
- [66] NI, S.-Y., TSENG, Y.-C., CHEN, Y.-S., et al. “The broadcast storm problem in a mobile ad hoc network”. In: *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 151–162, New York, NY, USA, 1999. ACM. ISBN: 1-58113-142-9. doi: <http://doi.acm.org/10.1145/313451.313525>.
- [67] LEVIS, P., PATEL, N., CULLER, D., et al. “Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks”. In: *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, pp. 2–2, Berkeley, CA, USA, 2004. USENIX Association.
- [68] LI, X.-Y., MOAVENINEJAD, K., FRIEDER, O. “Regional gossip routing for wireless ad hoc networks”, *Mob. Netw. Appl.*, v. 10, n. 1-2, pp. 61–

77, 2005. ISSN: 1383-469X. doi: <http://doi.acm.org/10.1145/1046430.1046436>.

- [69] GARBINATO, B., ROCHAT, D., TOMASSINI, M. “Power-efficient gossiping in multi-hop ad hoc networks”. In: *Autonomics '08: Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, pp. 1–8, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN: 978-963-9799-34-9.
- [70] KINI, A. V., VEERARAGHAVAN, V., SINGHAL, N., et al. “SmartGossip: an improved randomized broadcast protocol for sensor networks”. In: *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, pp. 210–217, New York, NY, USA, 2006. ACM. ISBN: 1-59593-334-4. doi: <http://doi.acm.org/10.1145/1127777.1127812>.
- [71] VAN RENESSE, R., BIRMAN, K. P., VOGELS, W. “Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining”, *ACM Trans. Comput. Syst.*, v. 21, n. 2, pp. 164–206, 2003. ISSN: 0734-2071. doi: <http://doi.acm.org/10.1145/762483.762485>.
- [72] MAKINEN, O., LUUKKAINEN, S. “Mobile Social Media Service Scenarios”. In: *ICWMC '09: Proceedings of the 2009 Fifth International Conference on Wireless and Mobile Communications*, pp. 373–378, Washington, DC, USA, 2009. IEEE Computer Society. ISBN: 978-0-7695-3750-4. doi: <http://dx.doi.org/10.1109/ICWMC.2009.69>.
- [73] EAGLE, N., PENTLAND, A. “Social Serendipity: Mobilizing Social Software”, *IEEE Pervasive Computing*, v. 4, n. 2, pp. 28–34, 2005. ISSN: 1536-1268. doi: <http://dx.doi.org/10.1109/MPRV.2005.37>.
- [74] ANASTASI, G., BORGIA, E., CONTI, M., et al. “IEEE 802.11b Ad Hoc Networks: Performance Measurements”, *Cluster Computing*, v. 8, n. 2-3, pp. 135–145, 2005. ISSN: 1386-7857. doi: <http://dx.doi.org/10.1007/s10586-005-6179-3>.
- [75] KNUTH, D. E. *Seminumerical Algorithms*, v. 2, *The Art of Computer Programming*. Second ed. Reading, Massachusetts, Addison-Wesley, 10 jan. 1981. Seção 3.4.1, subseção C, algoritmo P.
- [76] JEREW, O., JONES, H., BLACKMORE, K. “On the minimum number of neighbours for good routing performance in MANETs”. In: *Mobile Adhoc*

and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on, pp. 573–582, Oct. 2009. doi: 10.1109/MOBHOC.2009.5336952.

- [77] JAFER, E., O'FLYNN, B., O'MATHUNA, C., et al. “A Study of the RF Characteristics for Wireless Sensor Deployment in Building Environment”. In: *SENSORCOMM '09: Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications*, pp. 206–211, Washington, DC, USA, 2009. IEEE Computer Society. ISBN: 978-0-7695-3669-9. doi: <http://dx.doi.org/10.1109/SENSORCOMM.2009.41>.
- [78] IEEE. “802.15.4 Standard, 2003”. . <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>, 2003. Acessado em junho de 2010.
- [79] “ZigBee Alliance”. . <http://www.zigbee.org/>, 2009. Acessado em junho de 2010.
- [80] ZIMMERMANN, H. “OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection”, *Communications, IEEE Transactions on*, v. 28, n. 4, pp. 425–432, 1980. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1094702.
- [81] IEEE. “802.15.4 Standard, 2006”. . <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>, 2006. Acessado em junho de 2010.
- [82] “Bluetooth Products”. . <http://www.bluetooth.com/Bluetooth/Products/Products/>, 2009. Acessado em junho de 2010.
- [83] HILL, J., SZEWCZYK, R., WOO, A., et al. “System architecture directions for networked sensors”, *SIGPLAN Not.*, v. 35, n. 11, pp. 93–104, 2000. ISSN: 0362-1340. doi: <http://doi.acm.org/10.1145/356989.356998>.
- [84] GAY, D., LEVIS, P., VON BEHREN, R., et al. “The nesC language: A holistic approach to networked embedded systems”. In: *PLDI '03: Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*, pp. 1–11, New York, NY, USA, 2003. ACM. ISBN: 1-58113-662-5. doi: <http://doi.acm.org/10.1145/781131.781133>.
- [85] CULLER, D., BONNET, P., ESTRIN, D., et al. “TinyOS Open Technology Alliance”. . <http://sing.stanford.edu/doc/tinyos-alliance.pdf>, 2004. Acessado em junho de 2010.
- [86] *MSP430x1xx family user's guide*. Texas Instruments, 2006.
- [87] *CC2420 Data Sheet - 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Texas Instruments, 2006.

- [88] BLUMENTHAL, J., GROSSMANN, R., GOLATOWSKI, F., et al. “Weighted Centroid Localization in Zigbee-based Sensor Networks”. In: *in IEEE International Symposium on Intelligent Signal Processing, WISP 2007*, 2007.
- [89] SRINIVASAN, K., LEVIS, P. “RSSI is Under Appreciated”. In: *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*, 2006.

Apêndice A

Resultados dos Experimentos da Topologia 1

Neste capítulo apresentamos os resultados obtidos com os experimentos realizados na Topologia 1.

A.1 Experimento 1

O experimento 1 foi executado nesta topologia com a seguinte configuração:

- Primeira execução, apenas o Tmote 1 enviando;
- Segunda execução, quatro Tmotes enviando: 1, 2, 14 e 19;
- Terceira execução, oito Tmotes enviando: 2, 3, 7, 10, 13, 15, 16 e 19;
- Quarta execução, doze Tmotes enviando: 1, 2, 3, 7, 9, 10, 13, 14, 15, 16 e 19;
- Quinta execução, dezesseis Tmotes enviando: 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 19;
- Sexta execução: Todos Tmotes enviando;

A REPI-A foi configurada de duas formas: com apenas dois campos no prefixo B e com oitos campos no prefixo B . Os resultados deste experimento na topologia 1 são apresentados a seguir.

A.1.1 Taxa de Entrega de Mensagens em função do número de mensagens na rede

A Figura A.1 apresenta a evolução da Taxa de Entrega de Mensagens (\overline{TEM}) em função do número de mensagens (M) inseridas na rede. Como já era esperado, a

\overline{TEM} diminuiu em função do maior número de mensagens, variando de 80,71% para oito campos e 75,26% para dois campos na execução com apenas um Tmote enviando mil mensagens; até 50,84% de entrega com dezesseis Tmotes enviando com oito campos e 46,12% para dezesseis Tmotes enviando com dois campos no prefixo.

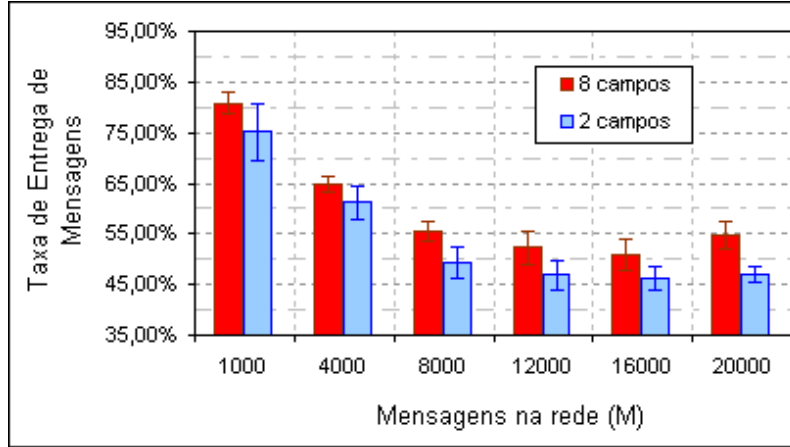


Figura A.1: Gráfico que mostra a evolução da Taxa de entrega de Mensagens (\overline{TEM}) em função do número de mensagens enviadas, com duas configurações da REPI-A.

O primeiro aspecto interessante dos dados obtidos foi perceber que em ambos os casos o desempenho da rede foi muito ruim, visto que, mesmo no melhor caso, a \overline{TEM} foi de cerca de 80%.

Elencamos as seguintes hipóteses como causas deste problema:

- A proximidade dos Tmotes, que foram colocados nestas posições para garantir que, mesmo com baixa potência, haveria espaço suficiente para a transmissão por rádio frequência (RF). Porém, a proximidade trouxe um outro problema: mais Tmotes ficaram dentro do mesmo raio de alcance de RF, o que gerou interferência na rede, trazendo degradação do sinal;
- O intervalo de envio entre mensagens foi pequeno, o que também causou interferência na rede, em especial quando todos os Tmotes estavam transmitindo. Nesta situação, a contenção na rede era tanta que após a última mensagem ser inserida, os Tmotes ficavam por quase uma hora trocando mensagens que ficaram armazenadas nos diversos *buffers* da rede.
- No caso com dois campos, o algoritmo se tornou muito restritivo e as mensagens não chegaram aos seus destinatários em função do excesso de filtragem na rede, enquanto que com oito campos o excesso de mensagens causou muita interferência.

Para comprovar as hipóteses apresentadas, este experimento foi executado novamente com outras configurações na Topologia 2, apresentada na Seção 4.5, onde esta

discussão é retomada. Na sequência apresentaremos os outros resultados obtidos nos experimentos da Topologia 1.

Outro aspecto interessante do gráfico é que na configuração com vinte Tmotes enviando mensagens, na qual foram inseridas vinte mil mensagens na rede, houve um aumento na \overline{TEM} , revertendo a tendência de crescimento observada até então.

A primeira hipótese levantada foi a de que os quatro Tmotes inseridos nesta configuração poderiam ter tido uma taxa de entrega superior, por estarem em pontos mais externos da rede, onde há menos interferência.

Para avaliar esta hipótese, a \overline{TEM} da configuração com vinte tmotes foi calculada considerando apenas os dados dos dezesseis Tmotes que estavam na configuração anterior. Com este cálculo, obtivemos uma \overline{TEM} de 50,19%, similar a taxa de 50,84% observada na configuração com dezesseis Tmotes. Com isso, concluímos que a queda na taxa de perda de mensagens foi causada pelo melhor desempenho do quatro Tmotes inseridos, que sofreram menos interferência por estarem em uma zona de menor congestionamento.

A.1.2 Número de saltos

Na versão da aplicação mensageira utilizada neste experimento, ela contabilizava o número de saltos de todas as mensagens da rede, sem fazer distinção entre as que foram aceitas pelos Tmotes e as que foram descartadas.

Portanto, a análise detalhada do número de saltos das mensagens entregues será feita apenas nos experimentos da Topologia 2, apresentados na Seção 4.5, pois eles foram realizados em uma versão da aplicação que já contava com esta funcionalidade.

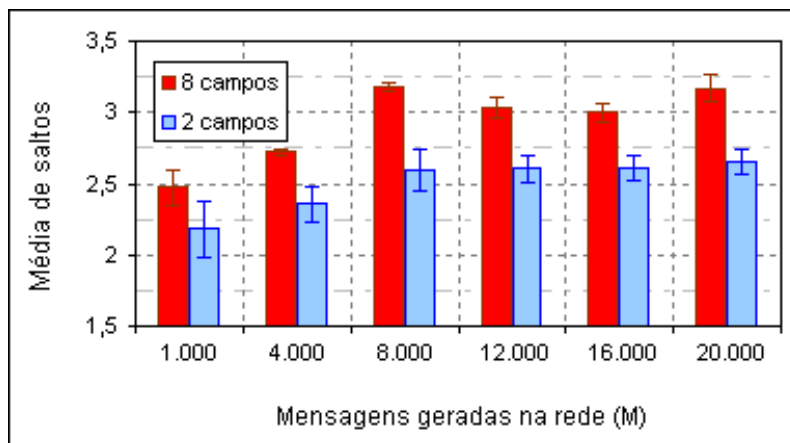


Figura A.2: Média do número de saltos em função do número de mensagens geradas e com duas configurações da REPI-A.

Na Figura A.2 são apresentados os valores do Número Médio de Saltos das mensagens com dois e oito campos no prefixo B . É possível notar que na rede com dois campos, onde a probabilidade de transmissão é menor, o número de salto é sempre

menor que na configuração com oito campos, variando de uma diferença de 11,78% quando foram inseridas mil mensagens na rede, até uma diferença de 18,27%, quando foram inseridos oito mil pacotes na rede.

Este resultado já era esperado, uma vez que em uma rede com maior probabilidade de transmissão as mensagens são encaminhadas por mais nós.

A.2 Experimento 2

O experimento 2 foi executado nesta topologia com a seguinte configuração:

Os Tmotes que injetam pacotes na rede são os mesmos em todas as execuções: 2, 7, 12 e 19. Os Tmotes que são destinatários são os seguintes:

- Primeira execução, quatro Tmotes como destinatários: 3, 6, 10 e 13;
- Segunda execução, oito Tmotes como destinatários: 1, 3, 6, 10, 11, 13, 18 e 20;
- Terceira execução, doze Tmotes como destinatários: 1, 3, 4, 5, 6, 8, 10, 11, 13, 15, 18 e 20;
- Quarta execução, dezesseis Tmotes como destinatários: 1, 3, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18 e 20;
- Quinta execução, todos Tmotes como destinatários;

A seguir são apresentados os dados obtidos com a execução do Experimento 2 na Topologia 1.

A.2.1 Taxa de Entrega de Mensagens em função do número de destinatários na rede

A Figura A.3 apresenta a evolução da Taxa de Entrega de Mensagens (\overline{TEM}) em função do número de nós destinatários e com três configurações: com dois campos no prefixo B com Colaboração, com dois campos sem Colaboração e com oito campos com Colaboração.

Como é possível notar, a configuração com oito campos foi a melhor das três, mantendo uma taxa de entrega constante, na faixa de 66%. Isto acontece porque nesta configuração o descarte de pacotes em razão do prefixo é muito menor do que na configuração com dois campos, que é muito mais restritiva, conforme mostra o gráfico da Figura A.4.

Neste gráfico é possível perceber que na configuração com dois campos há um descarte de mensagens por perfil na faixa de dezoito mil mensagens, enquanto que

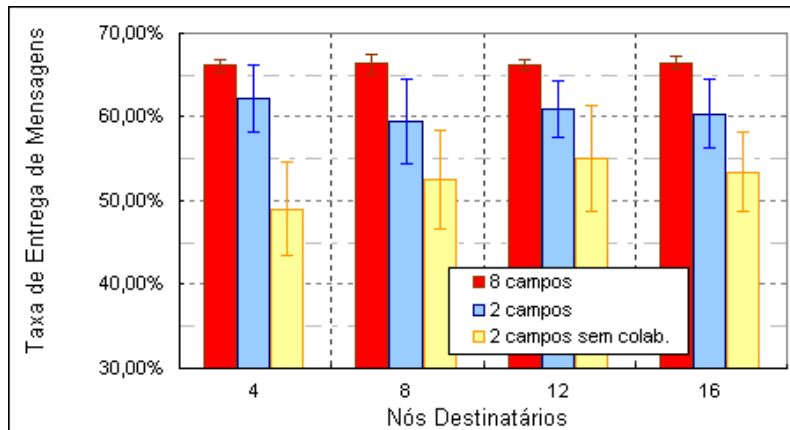


Figura A.3: Gráfico que mostra a evolução da Taxa de Entrega de Mensagens (TEM) em função do número de destinatários com três configurações da REPI-A.

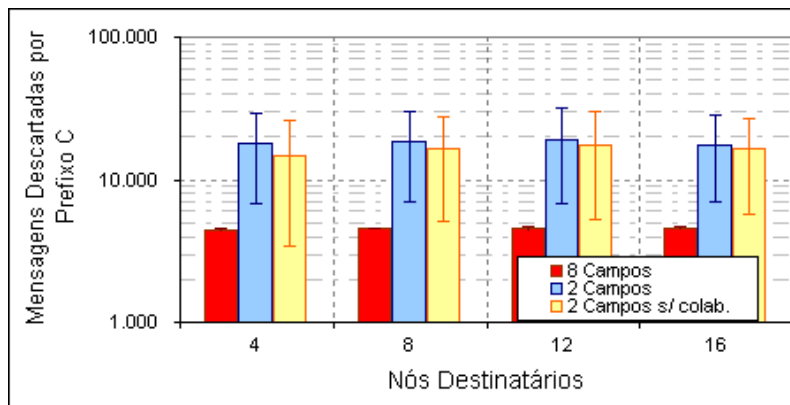


Figura A.4: Gráfico que mostra a quantidade de mensagens descartadas por ausência de casamento de prefixo C com três configurações da REPI-A.

na configuração sem Colaboração esse número varia entre quatorze e dezoito mil. Em ambos os casos, a perda é muito maior que na configuração com oito campos, que só descarta cerca de quatro mil mensagens.

Além disso, no gráfico da Figura A.3 é possível notar o impacto que a Colaboração tem na entrega de mensagens. Como era esperado, esse impacto é amenizado conforme o número de destinatários na rede aumenta, uma vez que isso aumenta o número de nós que encaminham a mensagem por possuir o mesmo interesse que ela. Ainda assim, na configuração com dezesseis destinatários, onde apenas quatro nós não colaboram com o envio, há uma diferença de cerca de 5% na taxa de entrega.

Outra conclusão é que quanto mais impopular é um interesse, mais dependente de Colaboração ele é, o que sugere que mecanismos adaptativos para regular o uso da Colaboração poderia trazer ganhos de desempenho para REPI.

A.2.2 Porcentagem de Mensagens Encaminhadas de forma colaborativa

A Porcentagem de Mensagens Encaminhadas de Forma Colaborativa (\overline{MC}), apresentada na Figura A.5, indica como a Colaboração na rede variou de acordo com a quantidade de nós com interesse Y igual ao dos nós fonte. Como já era esperado, em ambos os casos o aumento do número de nós interessados diminui o número de mensagens do forma colaborativa, uma vez que há menos nós colaboradores na rede.

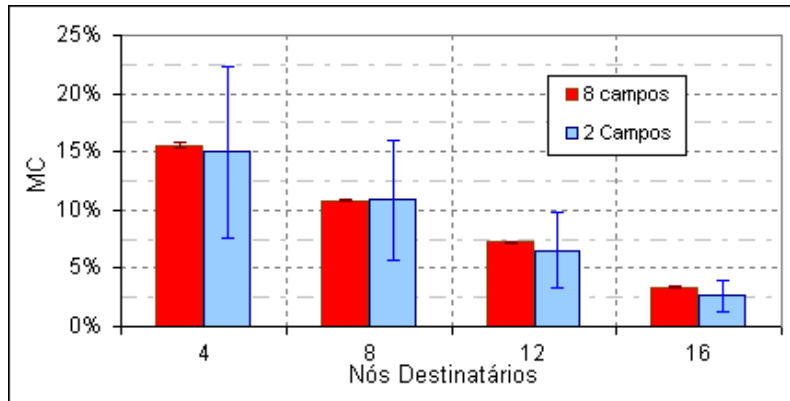


Figura A.5: Gráfico que mostra a Porcentagem de Mensagens Encaminhadas de forma colaborativa (\overline{MC}) em função do número de Nós Destinatários, com duas configuração da REPI-A.

A.2.3 Porcentagem de nós colaboradores

A Porcentagem de Nós Colaboradores determina qual foi o percentual de nós da rede que tinham potencial para colaborar com ela e que efetivamente o fizeram. Como é possível observar na Figura A.6, em ambos os casos os percentuais de nós colaboradores é alto, com valor mínimo de mais de 90%.

Além disso, é possível notar que o valor com dois campos se mantém entre cerca de 90% e 94% para todas as configurações, enquanto a porcentagem para oito campos é de 100% para as configurações com 12 e 16 nós. Este também é um resultado esperado, visto que na rede com oito campos as mensagens possuem mais caminhos de propagação e é mais provável que elas passem por todos os nós, facilitando que esses colaborem. Já em uma rede com dois campos, onde há menos caminhos para as mensagens trafegarem, é mais provável a ocorrência de nós isolados e incapazes de colaborar.

É interessante observar que os resultados obtidos foram iguais em todas as execuções com oito campos. Enquanto isso, na configuração com dois campos houve grande variação estatística. Isto ocorre também ocorre em função da maior probabilidade da existência de nós que não tenham casamento de prefixo com nenhum

dos nós fonte, o que ocorre apenas em algumas execuções.

Isso indica que as configurações com mais campos além de melhor desempenho são mais estáveis e confiáveis.

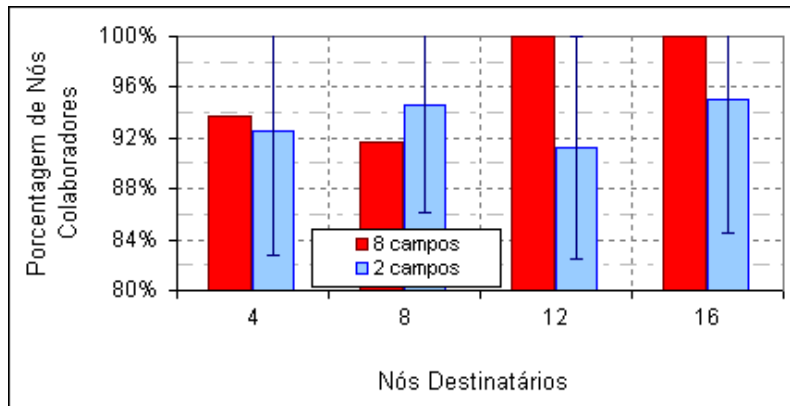


Figura A.6: Gráfico que mostra a Porcentagem de nós colaboradores (\overline{NC}) em função do número de Nós Destinatários com duas configurações da REPI-A.

A.2.4 Custo Colaboração

O Custo de Colaboração (\overline{CCOL}) determina qual é a Colaboração média individual que cada nó colaborador tem no total de mensagens colaborativas entregues na rede. Um alto custo de Colaboração pode ser problemático, pois indica que os nós colaboradores estão sendo onerados, o que faz com que seus recursos sejam consumidos sem que ele tenha uma compensação garantida, visto que não há garantias de que os outros nós colaborarão para que ele receba suas mensagens.

No gráfico da Figura A.7 são comparados quatro valores: o custo da entrega de mensagens com dois e oito campos tanto dos nós que colaboraram com a rede quanto com os que não colaboraram. Como é possível notar, o custo dos dois tipos de nós é similar, com uma leve vantagem para o custo dos nós que colaboraram com a rede.

Este é um bom indicador para a rede, pois demonstra que não há sobrecarga nos nós colaboradores.

Além disso, é possível perceber que as configurações com oito campos tem um custo superior aos das configurações com dois campos. Este é um resultado esperado, visto que nesta configuração o número de mensagens na rede é superior.

A.3 Experimento 3

O experimento 3 é executado nesta topologia com a seguinte configuração:

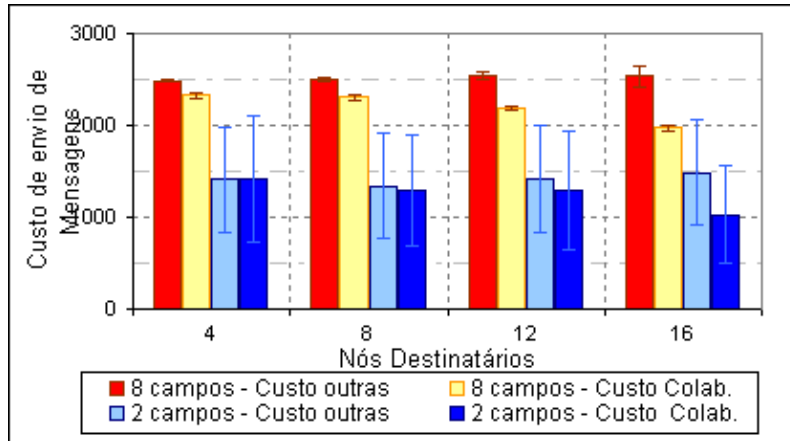


Figura A.7: Gráfico que mostra o custo de Colaboração (\overline{CCOL}) em função do número de Nós Destinatários com quatro Configurações da REPI-A.

Os seguintes Tmotes injetam mensagens na rede: 7, 11, 12, 18 e 20; enquanto os seguintes Tmotes são destinatários: 1, 3, 4, 5 e 6. Esta configuração é mantida constante, enquanto que o número de campos no prefixo é variado nos valores de 1, 2, 3, 4, 6 e 8, além de uma execução com um algoritmo de inundação.

Os resultados do experimento 3 na topologia 1 são apresentados a seguir.

A.3.1 Número de nós colaboradores

Esta métrica avalia a variação do Número de Nós Colaboradores da rede em função da variação do número de campos do Prefixo C. Neste cenário, quinze dos vinte nós possuem perfil Y diferente do das mensagens que circularam na rede e todos eles são potenciais nós colaboradores.

Como mostra o gráfico apresentado na Figura A.8, a Colaboração é máxima nas configurações com quatro, seis e oito campos. Enquanto que nas outras três os valores também são altos, com 13,9 para apenas um campo, 14,9 para dois campos e 14,85 para três.

Esse comportamento já era esperado, visto que nesta topologia os nós possuíam muitos vizinhos, o que fez com que as mensagens tenham muitos caminhos para circular pela rede. Como temos cinco nós fonte, a probabilidade de que os nós da rede colaborem com pelo menos um deles é ainda maior.

A.3.2 Taxa de Entrega de Mensagens em função do algoritmo de roteamento

O gráfico da Figura A.9 apresenta a Taxa de Entrega de Mensagens em função do algoritmo de roteamento utilizado na rede, mostrando o comparativo de seis configurações do protocolo REPI e uma com um algoritmo de inundação (*Flooding*). O

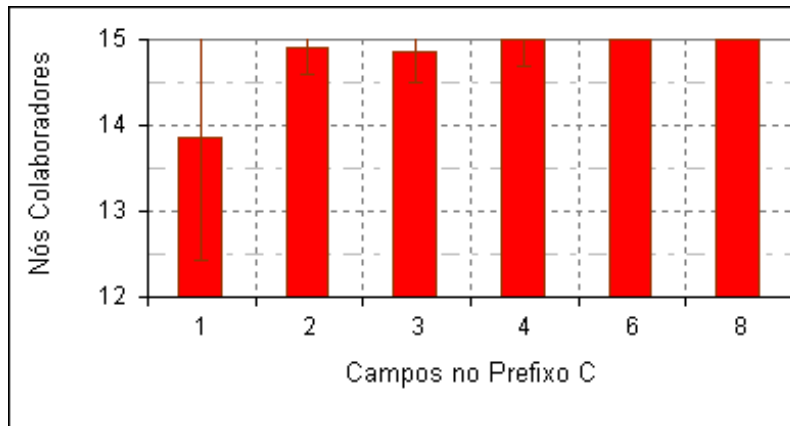


Figura A.8: Gráfico que mostra o número de nós colaboradores (\overline{NC}) de acordo com a variação do número de campos no Prefixo C.

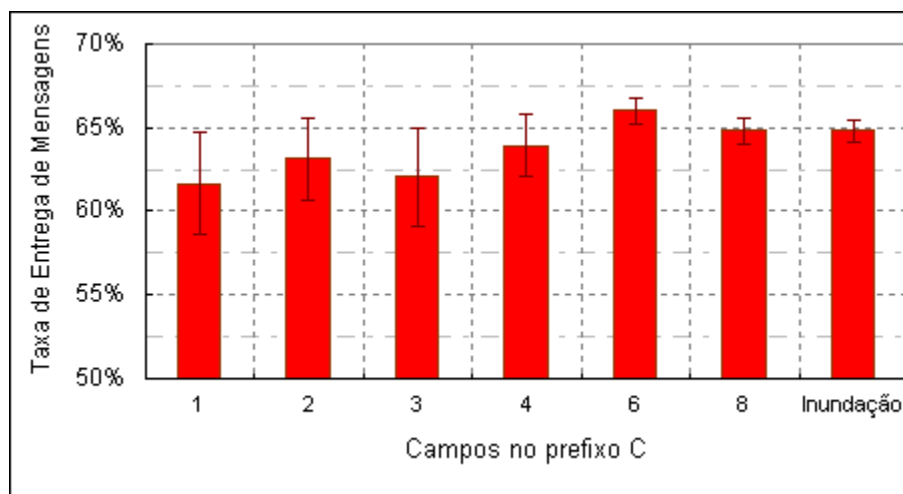


Figura A.9: Gráfico que mostra a Taxa de Entrega de Mensagens (\overline{TEM}) de acordo com a variação do número de campos no prefixo C e comparação com algoritmo de inundação.

resultado apresentado contraria as nossas expectativas, visto que o esperado era que, ao aumentar o número de campos no prefixo, o que aumenta o número de caminhos na rede, esperávamos que a taxa de entrega também fosse maior. Porém, é possível perceber que, dentro das variações estatísticas, a taxa de entrega permaneceu por volta de 64% para todas as configurações.

Além disso, como é possível notar pela análise da Figura A.10, o número do Total de Mensagens na Rede, aumentou cerca de 85% comparando a configuração com um campo com a de oito campos, sem que o número de mensagens entregues sofresse variações relevantes.

Acreditamos que este fato ocorreu em razão da grande contenção na rede, em função do pequeno intervalo de transmissão, que nesse experimento é de $[0,5;1,5]$ segundos. Como mostram os resultados dos experimentos da topologia dois, com intervalos maiores, a diferença entre as configurações é muito mais nítida, além da

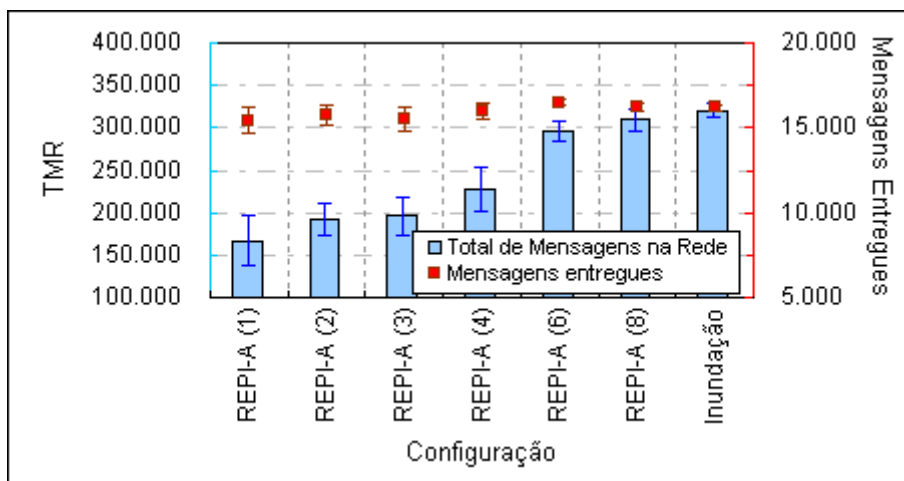


Figura A.10: Gráfico que mostra a evolução da relação entre o Total de Mensagens Recebidas (TMR) e de mensagens entregues ao destinatário final em função do número de campos do prefixo C e em comparação com algoritmo de inundação.

taxa de entrega ser superior.

O principal problema deste intervalo reduzido é o congestionamento, visto que quando novas mensagens são inseridas na rede, as anteriores ainda estão circulando. Porém, como já foi mostrado em outros trabalhos [46], mesmo com os mecanismos de detecção de colisão a contenção gera perdas de pacotes substanciais em redes de sensores com o padrão 802.15.4. Este problema ocorre tanto por colisão de pacotes quanto por interferências de rádio frequência, principalmente neste cenários, com até vinte dispositivos transmitindo simultaneamente.

Apêndice B

Manual da Aplicação Mensageira

A aplicação mensageira é interface do usuário para participar da rede, através da aplicação de troca de mensagens. As suas principais funcionalidades são:

- Possibilitar a troca de mensagens através da REPI-A;
- Monitoramento de informações dos usuários com os quais a aplicação trocou mensagens, conforme descrito na Seção B.8;
- Monitoramento do uso da rede pela aplicação, através da coleta de informações sobre troca, descarte, encaminhamento e aproveitamento de mensagens;
- Monitoramento dos possíveis erros ocorridos no Tmote controlado pela aplicação;
- Monitoramento da configurações globais da rede, fornecidas por SAMCRA, apresentado na Seção E.2;
- Execuções automatizadas para a realização de experimentos, conforme detalhado na Seção E.2
- Monitoramento do tempo de troca de mensagens na rede, com e sem o *overhead* gerado pela própria aplicação mensageira, conforme descrito na Seção B.12.

Nas próximas Seções são apresentados os detalhes da aplicação, sua interface gráfica e seu uso.

B.1 Parâmetros de inicialização

Ao iniciar a execução da aplicação mensageira, o usuário pode especificar uma série de parâmetros que determinarão o comportamento desta. Estes parâmetros são opcionais e devem ser especificados na linha de comando que executa a classe principal

da aplicação. Na versão 1.39 , usada neste trabalho, os parâmetros de inicialização são os seguintes:

auto X

Faz com que a aplicação seja inicializada sem interferência do usuário. O parâmetro X é opcional e deve ser usado quando há mais de um Tmote ligado na mesma máquina, o que faz com que o Tmote de número X seja selecionado pela aplicação. Esse parâmetro é necessário para a execução automatizada da aplicação, apresentada na Seção E.2;

autoClose X

Similar ao parâmetro auto, mas além do início automático, a aplicação também é fechada automaticamente após o término da execução automatizada;

chooseProfile

Permite que o usuário escolha as suas características par participar da REPI-A. Se esse parâmetro não for especificado, esses campos terão preenchimento randômico, seguindo a distribuição probabiística normal, ou seja, o efeito será o mesmo que a especificação do parâmetro *gaussian=8*, mostrado a seguir;

gaussian=x(0-8)

Indica quantas das oito características previstas no prefixo serão preenchidos com valores randômicos seguindo a distribuição gaussiana. O valor 0 (zero) faz com que os valores dos campos sejam iguais ao da última execução da aplicação. Os valores da distribuição Gaussiana são obtidos através do método *nextGaussian* da classe Random, que faz parte das classes básicas do Java. Este método retorna um número pseudo-randômico que segue a distribuição Gaussiana (normal) em uma variável do tipo *double* com média 0.0 e desvio padrão 1.0. A implementação do método é feito com base no método polar descrito em [75];

ints=x(0-5)

Seleciona os x primeiros canais públicos;

adminMode

Ativa o modo administrador, descrito na Seção B.5;

SaveVars

Faz com que a aplicação salve os dados sobre o uso da rede em um arquivo em disco. Com isso, a aplicação passa a ter persistência dos dados gerados caso sua execução seja interrompida;

Version

Mostra a versão atual da aplicação;

NoIntro

Não exibe as telas iniciais da aplicação, mostradas na Seção B.4 ;

Server=ENDEREÇO Especifica o endereço do servidor remoto na rede TCP/IP, deve ser usado caso a opção *noIntro* seja especificada;

Verbose

Exibe as mensagens sobre eventos relevantes (*log*) geradas ao longo da execução do programa na linha de comando onde a aplicação foi iniciada. Caso contrário, as mensagens vão para o arquivo de texto, chamado “log.txt”, salvo no mesmo diretório onde a aplicação se encontra.

B.2 A interface gráfica da aplicação

Para oferecer as funcionalidades descritas anteriormente, foi desenvolvida uma interface gráfica para a aplicação messageira utilizando o *ToolKit* gráfico Swing, que faz parte das classes básicas do Java.

O principal objetivo desta interface é facilitar o uso da aplicação messageira, com uma interface simples e, ao mesmo tempo, torná-la poderosa o suficiente para a avaliação do funcionamento da rede.

Esta flexibilidade é atingida usando os parâmetros apresentados na Seção anterior e cujos impactos na interface são mostrados nas Seções B.3, B.4 e B.5.

B.3 Tela de escolha de motes

Durante o uso típico da aplicação messageira, a rede possui um Tmote conectado por máquina. Mas é possível existir situações em que vários Tmotes sejam conectados em um mesmo PC, especialmente durante execuções automatizadas. Neste tipo de situação, a aplicação exibe uma tela de escolha de Tmote, mostrada na Figura B.1. Como mostra a Figura, a tela é simples e apresenta uma lista numerada com os Tmotes conectados e os números das portas seriais aos quais eles estão conectados. No exemplo da Figura, existem três Tmotes conectados no mesmo computador.

Nesse caso, deve existir uma instância da aplicação para cada Tmote. É importante ressaltar que ao especificar o parâmetro ***auto X*** ou ***autoClose X***, a escolha do Tmote é feita pela própria aplicação, de acordo com o valor de ***X***.

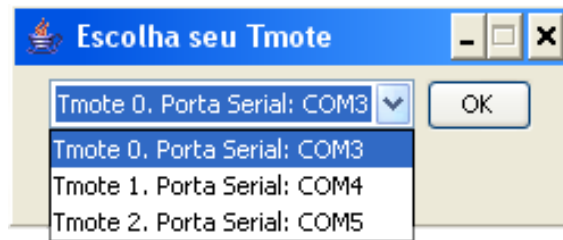


Figura B.1: Tela de escolha do Tmote em uma máquina com 3 Tmotes conectados.

B.4 Tela de Configuração Inicial

A Tela de Configuração Inicial, apresentada na Figura B.2, é usada para que o usuário possa configurar os seguintes itens da aplicação, de acordo com a ordem que são mostrados na Tela, de cima para baixo:

Idioma

Idioma usado pela aplicação mensageira, na versão 1.39 estão disponíveis os idiomas Português e Inglês;

Servidor

Endereço IP de SAMCRA, apresentado na Seção E;

Prefixo

Valores dos campos do prefixo do usuário (características e interesses), que são usados para construir a REPI-A. Esta parte da tela só é exibida se o usuário especificar o parâmetro *chooseProfile*, como mostra a Figura B.2. Em caso contrário, esta parte da tela é omitida e ela é apresentada como mostra a Figura B.3. Se o parâmetro *gaussian=x(0-8)* for especificado, os valores mostrados nesta tela já estarão randomizados de acordo com o valor especificado para X, que determina que os X primeiros campos serão randomizados. Os campos são ordenados da esquerda para a direita, de cima para baixo. Por exemplo, em um caso em que X=3, teremos os três características preenchidas com valores randômicos e as outras não serão utilizadas;

Canais públicos

Campos onde o usuário pode especificar de quais canais públicos ele deseja receber mensagens. O nome desses canais é definido em SAMCRA e podem ser customizados, como mostra a Seção E.2.

B.5 Modos padrão e administrador

Para flexibilizar o uso da Aplicação mensageira, ela possui dois modos de operação: o modo padrão, em que apenas as telas necessárias para utilização da rede e as

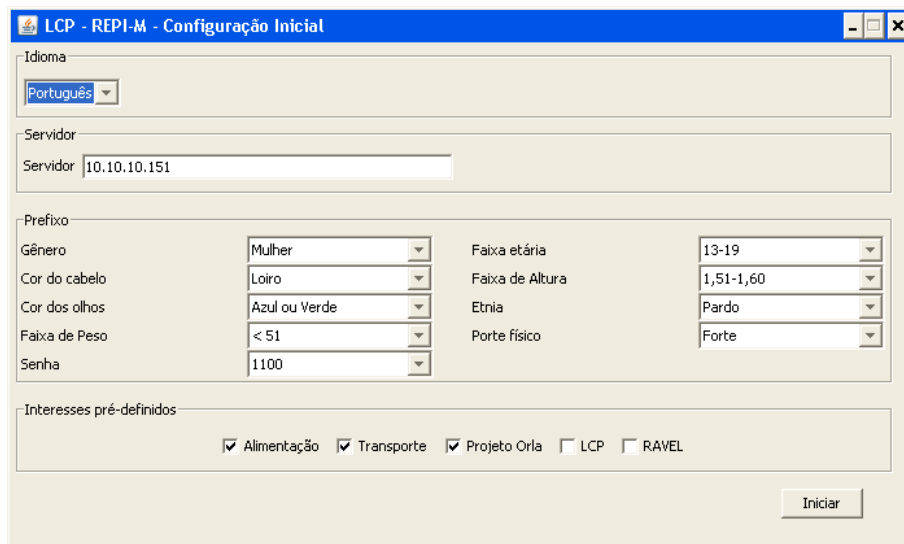


Figura B.2: Tela de Configuração Inicial da aplicação mensageira.

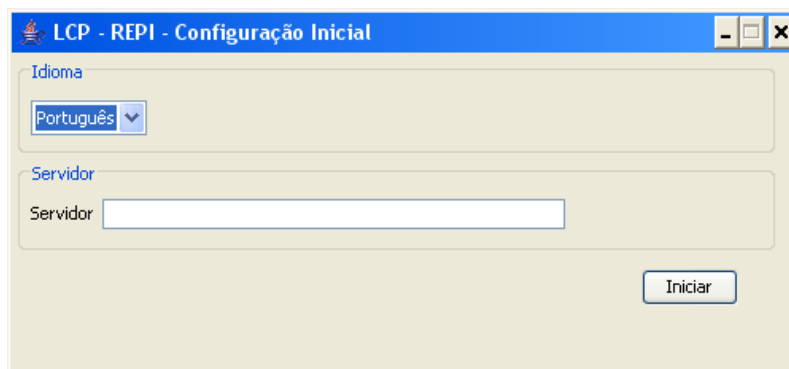


Figura B.3: Tela de Configuração Inicial da aplicação mensageira sem o painel de escolha do prefixo.

configurações básicas da aplicação são exibidas; e o modo administrador, onde além das telas do modo usuário, são exibidas configurações avançadas da aplicação e telas para monitorar e configurar o funcionamento da Rede.

Em ambos os modos de operação, a organização dos elementos da aplicação é feita através de abas, identificadas com ícones mnemônicos, mostrados na Figura B.4. Neste Figura, os ícones são numerados de 1 (um) a 8 (oito) para facilitar a organização do texto.



Figura B.4: Tela de Configuração Inicial da aplicação.

No modo Padrão, o usuário acessa apenas as três primeiras abas, indexadas de 1

a 3, no modo administrador ele tem acesso a todas as oito abas. As funcionalidades básicas das abas, de acordo com a numeração, são descritas a seguir:

1. Troca de mensagens

Tela onde o usuário pode trocar mensagens através da REPI, é descrita com detalhes na Seção 3;

2. Mapa de usuários da rede

Tela que exhibe um mapa onde o usuário pode ver informações sobre os interesses dos usuários da rede que lhe enviaram mensagens e a data do último contato com esses usuários. Ela é descrita com detalhes na Seção B.6;

3. Configurações

Tela onde o usuário pode configurar o comportamento da aplicação, é descrita com detalhes na Seção B.7. Todas as próximas abas só estão disponíveis no modo administrador;

4. Árvore de informações

Tela que exhibe uma árvore com informações detalhadas dos usuários da rede, mostrando os prefixos e indicando se houve ou não casamento de interesses com o prefixo local. Esta tela é descrita com detalhes na Seção B.8;

5. Tabela de contabilidade de pacotes recebidos

Tela que apresenta uma tabela com a contabilidade de todos os pacotes recebidos de cada um dos outros Tmote da rede, especificando se estes foram descartados e se tiveram casamento de interesse com o prefixo local. Esta tela é descrita em detalhes na Seção B.9;

6. Configurações do Servidor

Tela onde o usuário pode visualizar as configurações obtidas no Servidor de Parametrização da rede, descrito na Seção E.2. Esta tela é descrita com detalhes na Seção B.10;

7. Execução de testes automatizados

Tela onde o usuário pode realizar execuções automatizadas na rede, enviando pacotes sem interferência humana. Ela é descrita com detalhes na Seção B.11

8. Contagem de tempo de envio de pacotes

Tela onde o usuário pode medir o tempo necessário para envio de pacotes na rede. Tela descrita com detalhes na Seção B.12;

Nas Seções seguintes, as telas de cada uma das abas são descritas com detalhes.

B.6 Mapa de usuários da rede

A tela de Mapa de usuários da rede, apresentada na Figura B.5, exibe informações sobre os interesses dos outros participantes da rede. Nela é possível ver informações de todos os usuários com os quais a aplicação trocou mensagens.

No mapa de usuários existem dois tipos de ícones: o ícone do usuário local, representado por um computador e que é identificado pelo expressão formada pelo apelido atual mais o sufixo “(Você)”. No exemplo da Figura B.5, a expressão “Tmote1(Você)” indica que o apelido do usuário local é Tmote1; e os ícones dos outros usuários da rede, que são representados por círculos coloridos, que podem ter as cores verde, vermelho ou amarelo. As cores dos ícones indicam se houve

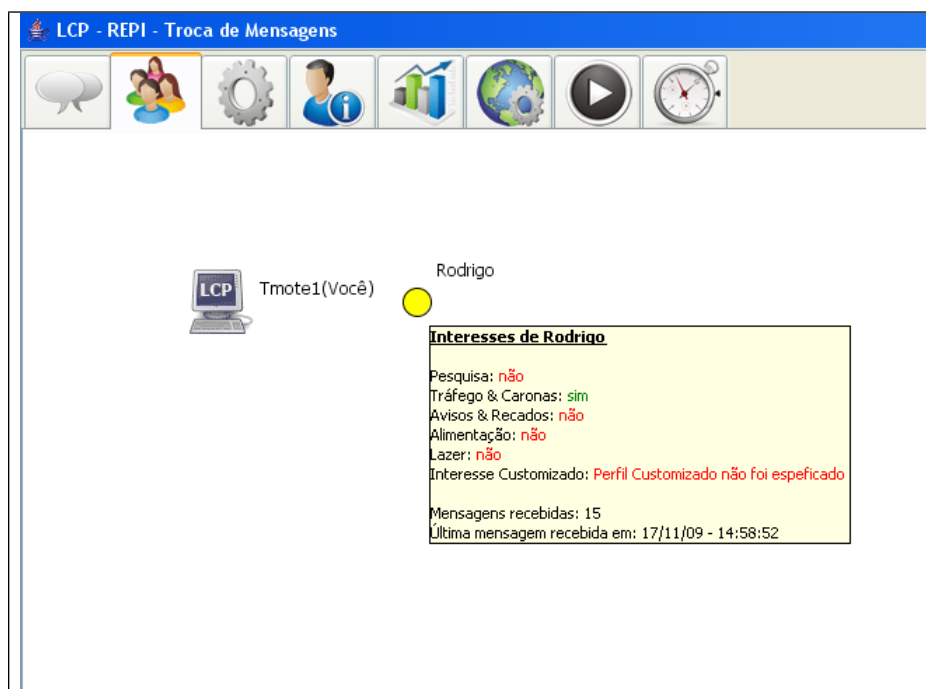


Figura B.5: Mapa de usuários da rede.

casamento no prefixo C, com o seguinte código de cores: verde, indica que houve casamento de interesses em todos os campos; amarelo, indica que houve casamento de interesse em pelo menos um campo; vermelho, indica que não houve casamento de interesse em nenhum dos campos.

Ao descansar o ponteiro do mouse sobre um dos elementos do mapa, algumas informações adicionais sobre o nó são apresentadas em um painel de dados. A Figura B.5 apresenta este painel. Nele é possível obter as seguintes informações a respeito de cada nó da rede com quem o usuário local se comunicou: seus interesses; total de mensagens recebidas; e horário da última mensagem recebida. Ao descansar o mouse sobre o ícone do Tmote local, é possível visualizar os canais públicos que estão selecionados e a quantidade de mensagens que foram enviadas e recebidas pela

aplicação mensageira.

B.7 Administração pelo usuário

Na aba de configurações, mostrada na Figura B.6, o usuário pode configurar alguns aspectos da aplicação que impactam em seu funcionamento e sua interface gráfica.



Figura B.6: Configurações da aplicação.

Nesta aba existem cinco itens passíveis de configuração, além do modo de funcionamento da aplicação, que é configurável apenas através do parâmetro de inicialização *adminMode*. O primeiro destes itens é o idioma e os restantes são apresentados a seguir:

Mostrar opções avançadas de envio de mensagens

Ao habilitar esta opção, o usuário tem acesso aos painéis de envio de mensagens no canal reservado e nos canais customizados, mostrados na Figura 3.4;

Manter a janela de conversa sempre visível

Esta opção só têm efeito sobre a aplicação se o painel de troca de mensagens for destacado da interface. Quando ela está habilitada, a janela onde ele é exibido permanece sempre visível no ambiente gráfico do sistema operacional;

Exibir alerta visual ao receber novas mensagens

Ao marcar esta opção, o usuário recebe um alerta visual, mostrado na Figura B.7, para lhe avisar que novas mensagens foram entregues. Este alerta só é mostrado se a janela de troca de mensagens não estiver sendo exibida;

Salvar automaticamente as estatísticas da aplicação no disco

Esta é uma opção avançada e só é exibida no modo administrador. Ao marcar esta opção, a aplicação salva todos os dados coletados sobre o uso da rede em

um arquivo no disco, sempre que qualquer dado for modificado. Dessa forma, a aplicação mantém a persistência de todos os dados coletados.

Essa opção deve ser evitada ao se realizar testes automatizados com muitas trocas de mensagens e execuções prolongadas, visto que pode ocorrer sobrecarga no disco. Porém, ela deve ser marcada na realização de testes com usuários reais. O parâmetro *saveVars* faz com que esta opção seja marcada automaticamente.

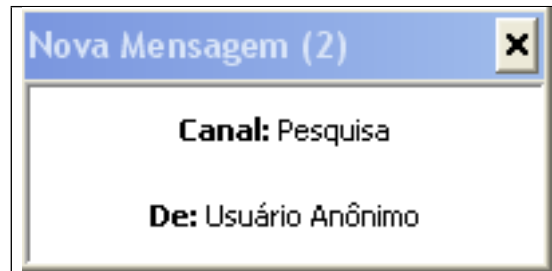


Figura B.7: Janela de alerta visual que notifica a chegada de novas mensagens.

B.8 Árvore de informações dos participantes

Na aba Árvore de informações do usuário, mostrada na Figura B.8, a aplicação exibe informações básica sobre a participação e os prefixos de todos os usuários da rede. Para melhor organizar a apresentação dos dados, os prefixos foram organizados em uma árvore, cujas folhas são numeradas de acordo com o endereço TinyOS do Tmote correspondente. O Tmote controlado pela aplicação local é identificado pelo sufixo “(você)”, que no exemplo da Figura B.8 é o número 13.

O painel com as informações, apresentado ao selecionar um dos Tmotes, é organizado em três Seções:

Informações Gerais

Apresenta o último apelido escolhido pelo usuário, o horário de entrega (envio se for o Tmote local) da última mensagem e, no prefixo local, o prefixo em bits;

Campos Biométricos (Distribuição Gaussiana)

Apresenta os valores do prefixo C daquele participante da rede. Ao exibir esta informação dos outros participantes da rede, como mostra a Figura B.9, os campos são mostrados em duas cores: verde, que indica ele tem o mesmo valor que o do prefixo local; ou vermelha, que indica que o campo tem valor diferente que o do prefixo local;



Figura B.8: Painel da árvore de informações dos participantes da rede, mostrando dados do prefixo local.

Interesses Distribuição Zipf

Apresenta os interesses dos participantes nos canais públicos da rede. Caso o prefixo seja de outro Tmote, os campos serão mostrados em duas cores: verde, se ambos os prefixos possuem este interesse; vermelho, em todos os outros casos. Os nomes desses interesses são determinados por SAMCRA, mostrado na Seção B.10.

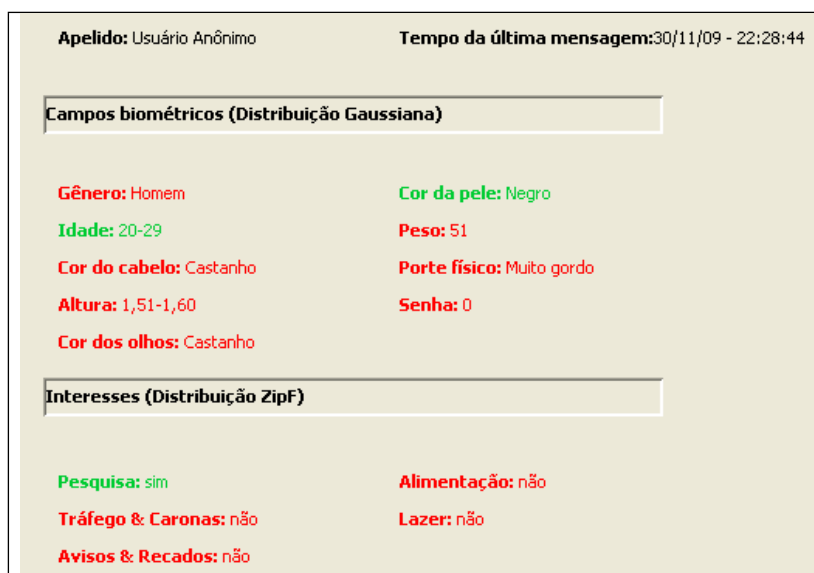


Figura B.9: Painel da Árvore de Informações dos Participantes da Rede, mostrando dados do prefixo de outro participante da rede.

B.9 Controle de Tráfego

Na aba de Controle de Tráfego, mostrada na Figura B.10, a aplicação exibe o detalhamento de todas as mensagens recebidas na rede, indicando quais foram descartadas e quais foram aproveitadas pela aplicação. Nela, cada tipo de mensagem é apresentada em duas linhas: uma para os dados organizados de acordo com o último nó por onde a mensagem passou e outra de acordo com o nó que originou a mensagem;

Esta tabela apresenta os dados organizados em seis linhas com as seguintes informações:

Geral

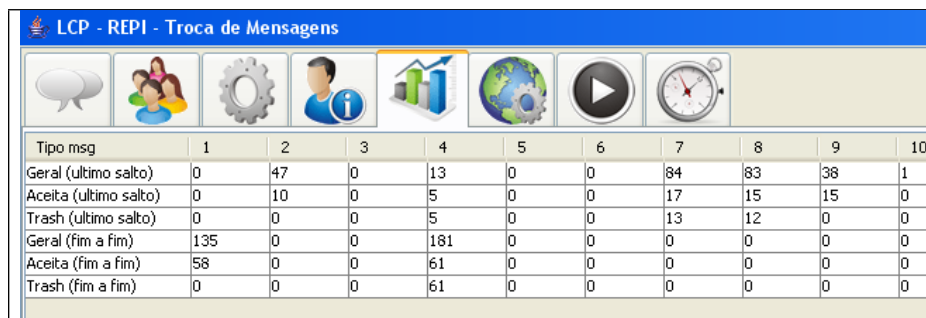
Total de pacotes recebidos pelo Tmote em relação ao último salto e em relação ao nó fonte ;

Matching

Total de pacotes que tiveram casamento de interesses com o Tmote local e foram aproveitados pela aplicação em relação ao último salto e em relação ao nó fonte;

Trash

Total de pacotes que foram descartados pela aplicação em relação ao último salto e em relação ao nó fonte;



Tipo msg	1	2	3	4	5	6	7	8	9	10
Geral (ultimo salto)	0	47	0	13	0	0	84	83	38	1
Aceita (ultimo salto)	0	10	0	5	0	0	17	15	15	0
Trash (ultimo salto)	0	0	0	5	0	0	13	12	0	0
Geral (fim a fim)	135	0	0	181	0	0	0	0	0	0
Aceita (fim a fim)	58	0	0	61	0	0	0	0	0	0
Trash (fim a fim)	0	0	0	61	0	0	0	0	0	0

Figura B.10: Tabela de informações das mensagens recebidas na rede, de acordo com o tipo.

As colunas numeradas de 1 a 20 representam os endereços TinyOS dos últimos Tmotes por onde as mensagens passaram antes de chegar no Tmote local (último salto) ou do Tmote que originou a mensagem (fim-a-fim). Essa tabela depois é consolidada em SAMCRA e com ela é possível criar um grafo da rede, mostrando todos os nós que tiveram comunicação direta ou fim-a-fim e como os pacotes trocados foram aproveitados.

B.10 Configurações da Rede

Esta aba, apresentada na Figura B.11, apresenta as configurações da rede, obtidas em SAMCRA. Além disso, o usuário também pode mudar o endereço de SAMCRA, mas essa mudança só terá efeito depois da aplicação ser reiniciada.



Figura B.11: Descrição das configurações obtidas no servidor de configuração da rede.

O estado atual das seguintes configurações são exibidas neste painel: a política de controle de pacotes; política de encaminhamento de mensagens; probabilidade de transmissão do algoritmo probabilístico; indicativo se há encaminhamento colaborativo na rede; HTL e indicativo se há envio de trace. A explicação detalhada de cada uma desses itens é apresentada na Seção E.2.

B.11 Execuções Automatizadas

Esta aba, apresentada na Figura B.12, permite a execução automatizada de experimentos, que faz com que a aplicação envie uma determinada quantidade de pacotes para a REPI-A sem intervenção do usuário. Este recurso foi usado para a realização dos experimentos apresentados neste trabalho. A execução automatizada pode ser iniciada automaticamente, através do parâmetro de linha de comando *auto* ou *autoClose* e a devida configuração do Módulo de Execução Automática de SAMCRA, apresentado na Seção E.2.

Neste painel, o usuário pode configurar dois parâmetros da execução: o número de mensagens enviadas, entre um e noventa mil; o intervalo entre o envio de duas mensagens consecutivas na rede, que pode ser configurada para um intervalo entre 0,5 e 600 segundos.



Figura B.12: Tela de ativação da execução automática, que faz a aplicação enviar mensagens na rede.

Com isso, a próxima mensagem é enviada na rede depois de um tempo de espera de um valor aleatório dentro do intervalo especificado, para evitar que exista sincronização de envio em experimentos com vários nós enviando mensagens na rede.

B.12 Módulo de contagem de tempo de envio de mensagens

Esta aba, apresentada na Figura B.13, é usada para realizar experimentos para contabilizar o tempo de entrega de pacotes na rede. Para isso, ela faz dois tipos de contabilidade: com e sem o *overhead* da aplicação messageira.

A contabilidade com *overhead* é feito calculando o tempo que uma mensagem leva para ir até os outros Tmotes da rede e retornar durante uma execução normal da aplicação. Ao longo do caminho é feita a contagem de quantos saltos a mensagem realizou. De posse do tempo de viagem da mensagem e do número de saltos, é calculado o tempo médio por salto.

Para calcular o tempo de entrega de mensagens sem o *overhead* da aplicação, a comunicação é feita com uma mensagem REPI modificada, que possui dois campos extras: um que marca o tempo que o pacote é enviado; outro que registra o tempo que ele retorna. Este cálculo é feito pelo middleware Tmote, sem interferência da aplicação messageira. Para isso, ela usa a função *getNow* do componente *Timer-MilliC* do TinyOS, que retorna o tempo atual em milissegundos, em uma variável de 32 bits. Quando o pacote retorna ao Tmote de origem depois de passar pela rede, o tempo de retorno é inserido no pacote e este é encaminhado para a aplicação messageira, que calcula o tempo de circulação do pacote e o tempo médio por salto. Esses dados são apresentados ao usuário na tabela mostrada na Figura B.13.

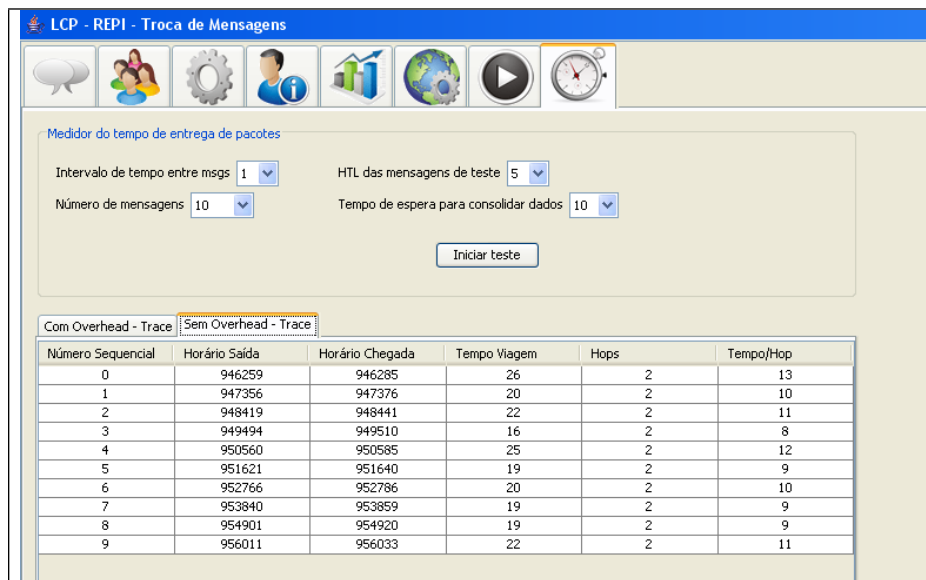


Figura B.13: Painel de contagem de tempo de envio de pacotes, com e sem *overhead* da aplicação.

Apêndice C

Redes Ad Hoc Sem Fio

As Redes Ad hoc sem fio são aquelas formadas por dispositivos que se comunicam através de conexões sem fio e não possuem infraestrutura fixa. Uma rede Ad Hoc possui várias características que a tornam uma área de pesquisa desafiante, a saber:

Topologias dinâmicas:

Como os dispositivos de uma rede Ad Hoc podem ser móveis e toda a comunicação é sem fio, com transceptores com alcance limitado, ela está sujeita a saída e entrada de nós. Por isso, as redes Ad hoc devem possuir capacidades de adaptação e de se auto-organizar quando há mudanças bruscas em sua topologia. Esta movimentação dos nós pode ter impactos no desempenho da rede [76];

Canais de comunicação assimétricos:

A comunicação sem fio através de rádio frequência (RF) é assimétrica[46]. Isto significa que dados dois nós de uma rede, chamados de A e B, é possível que o nó A consiga entregar pacotes para o nó B, mas a recíproca não seja verdadeira. Com isso, não há garantia de que, ao receber um pacote de um outro nó da rede, será possível estabelecer comunicação bidirecional com ele;

Ausência de infraestrutura fixa:

As redes Ad hoc, por definição, devem ser formadas sem a necessidade de uma infraestrutura fixa. Ou, seja, elas devem ser capazes de se auto-organizar e fazer a entrega dos pacotes da rede através dos nós móveis distribuídos;

Suscetibilidade a interferências:

Como qualquer outro dispositivo de comunicação sem fio, as tecnologias usadas para a criação de redes Ad hoc também estão sujeitas a ação de interferências na transmissão, que podem degradar a qualidade do sinal [77]. Esta interferência pode ser causada por vários fatores, como sinal de outras redes, reflexão de sinal e obstáculos físicos entre os dispositivos de comunicação;

Limitação de consumo de energia:

Em geral, dispositivos de comunicação sem fio são alimentados por baterias com tempo de vida limitado, o que faz com que a economia de energia seja muito importante neste tipo de rede;

Limitações de largura de banda:

Apesar dos grande avanços das tecnologias de comunicação sem fio, estas ainda são inferiores as tecnologias cabeadas em relação a confiabilidade e a largura de banda. Atualmente, os equipamentos para comunicação cabeada alcançam algumas centenas de gigabites por segundo de largura de banda, enquanto as tecnologias sem fio para dispositivos móveis de baixo consumo ainda operam entre menos de 1 até pouco mais de 100 megabits por segundo.

Apêndice D

Padrão 802.15.4 (*Zigbee*)

O padrão 802.15.4 [78] faz parte da família de padrões 802 da IEEE (*Institute of Electrical and Electronics Engineers*), que engloba padrões de redes locais e metropolitanas. O seu desenvolvimento é responsabilidade do décimo quinto grupo de trabalho da família 802 (802.15), encarregado dos padrões de Redes Pessoais sem Fio (WPAN - *Wireless Personal Area Networking*).

As WPANs, como o nome sugere, são redes criadas através da comunicação sem fio de dispositivos de uso pessoal, em geral com alcance de transmissão entre 10 e 100 metros. As WPANs são montadas com dispositivos de baixo custo, baixo consumo de energia, curto alcance de comunicação e tamanho reduzido.

O Grupo 802.15 classifica as WPANs em três classes, separadas pela taxa de transmissão, consumo de energia e qualidade de serviço (QoS), a saber:

WPAN de alta taxa de transmissão - IEEE 802.15.3: Apresenta altas taxas de transmissão de dados (entre 11 e 55 Mbps) e é adequada para aplicações de multimídia que exigem muita qualidade de serviço;

WPAN de taxa de transmissão média - IEEE 802.15.1/Bluetooth:

Apresenta taxa de comunicação média (entre 1 e 3 Mbps) e é adequada para comunicação entre celulares e PDAs, para troca de arquivos multimídia. Têm qualidade de serviço adequada para a transmissão de voz;

WPAN de taxa de transmissão baixa - IEEE 802.15.4: Apresenta baixa taxa de transmissão (entre 20 e 250 kbps) e é adequada para série de aplicações industriais, caseiras e médicas com consumo muito baixo e com necessidade de qualidade de serviço relaxada.

Neste trabalho o foco principal será o padrão IEEE 802.15.4, que é utilizado na implementação aqui apresentada. Porém, o padrão IEEE 802.15.1/Bluetooth é utilizado em vários dos trabalhos relacionados citados no Capítulo 5.

O padrão 802.15.4 é usado como base para o padrão comercial ZigBee, e é comum que na literatura ele seja tratados por ambos os nomes, na forma de 802.15.4/ZigBee. O ZigBee é mantido pela ZigBee Alliance [79], um consórcio de empresas que inclui gigantes da área de eletrônica, telecomunicação e computação como AT&T, Samsung, Sony e Telecom Italia.

Atualmente, tanto a ZigBee Alliance quanto a IEEE trabalham na especificação do padrão, cabendo a IEEE o foco na camadas mais próximas do hardware, física e enlace, e a ZigBee Alliance as demais, até a camada de aplicação, conforme a pilha de protocolos do modelo OSI [80] da ISO. Além da especificação do padrão, a ZigBee Alliance também é responsável por testes em dispositivos para garantir a compatibilidade e serviços como divulgação do padrão e criação de especificações e estudos do melhor uso dele em várias indústrias.

Dispositivos compatíveis com o padrão 802.15.4/ZigBee operam na faixa de frequência ISM (*Industrial, Scientific and Medical*), que varia de acordo com a região do mundo: 868 MHz na Europa, 915 MHz nos EUA e Austrália e 2,4 GHz no resto do mundo. Com taxas de transmissão de dados de 250kbps em 2,4GHz, 40kbps em 915MHz e 20kbps em 868MHz.

D.1 Dispositivos de uma rede 802.15.4/ZigBee

Em uma rede 802.15.4/ZigBee os dispositivos podem ser de dois tipos: Dispositivos com Funcionalidade Total (FFD - *Full-Function Device*); e Dispositivos com Funcionalidade Reduzida (RFD - *Reduced-Function Device*). Os dispositivos FFD podem se comunicar com qualquer outro dispositivo da rede, enquanto os RFD só podem se comunicar com os FFD.

Os dispositivos podem ter três tipos de funções na rede: coordenador, roteador ou dispositivo final. O coordenador (ZC - *ZigBee Coordinator*) é o FFD que inicia a rede e é único. Ele pode ser usado para se conectar a outras redes e é a raiz da árvore da rede. O Roteador (ZR- *ZigBee Router*) é o FFD responsável pelo encaminhamento de mensagens na rede, enviando mensagens próprias e encaminhando a de outros dispositivos. O dispositivo final (ZED - *ZigBee End Device*) é um RFD e por isso só se comunica com o seu nó pai, que deve ser um roteador ou o coordenador da rede.

D.2 Topologias de uma rede 802.15.4/ZigBee

As redes 802.15.4/ZigBee podem ter três tipos de topologias: Estrela (*Star*), Malha (*Mesh*) ou Múltiplas Árvores (*Cluster Tree*). Os três tipos são descritos a seguir.

D.2.1 Topologia em Estrela

Nesta topologia, um dispositivo FFD exerce um controle único central e é chamado de Coordenador PAN. Todas os nós da rede se comunicam com ele, e uma mensagem só pode ser passada entre os nós através dele. Além disso, todos os dispositivos são sincronizados periodicamente com o coordenador através de mensagens de sinalização (*beacons*). Se o coordenador falhar, toda a rede falha.

Na topologia estrela é comum que coordenador seja alimentado através de alimentação de uma fonte fixa, visto que tem alto gasto de energia, enquanto os outros dispositivos funcionam com pilhas.

Esta topologia é muito usada para automação doméstica, controle periféricos de computadores pessoais e jogos. Um exemplo de formação típica de uma rede com topologia estrela é apresentado na Figura D.1.

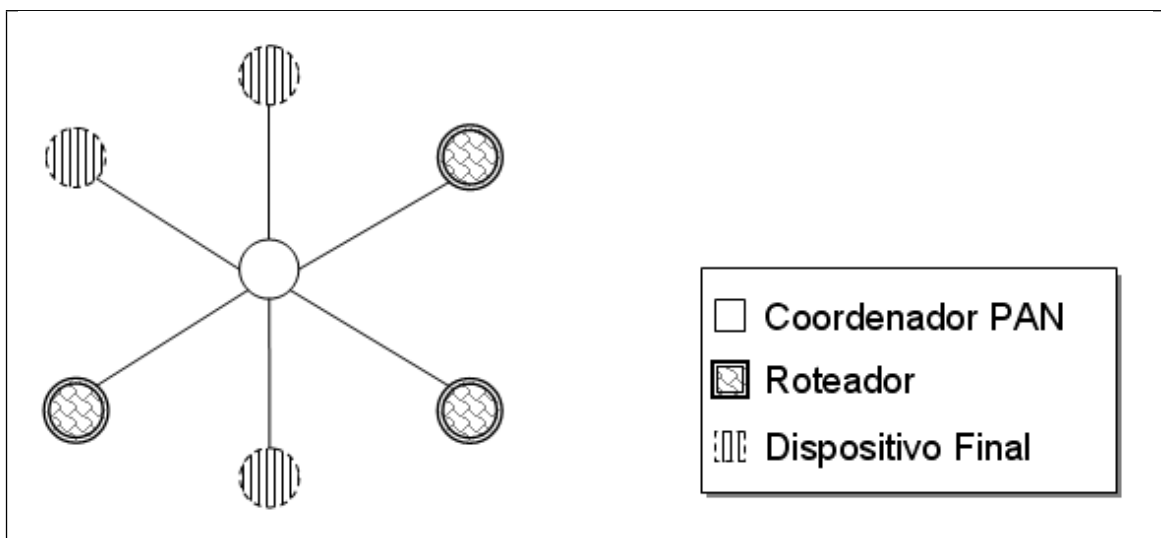


Figura D.1: Exemplo de topologia em estrela.

D.2.2 Topologia em Malha

Nesta topologia, todos os nós da rede podem se comunicar, bastando apenas que eles estejam no alcance do rádio de comunicação um do outro. Embora um nó seja eleito como o Coordenador PAN, ele não tem funções especiais como na topologia em estrela.

Neste tipo de topologia a rede possui a capacidade de entrega de mensagens com múltiplos saltos (*multihop*), onde uma mensagem pode passar por vários nós antes de ser entregue ao seu destino final. Com isso, um mesmo pacote pode ser entregue através de várias rotas distintas, o que aumenta a confiabilidade da rede.

Como a rede é capaz de resistir a falha de um nó e continuar funcionando, ela possui a propriedade de adaptação (*self-healing*). Além disso, a capacidade de

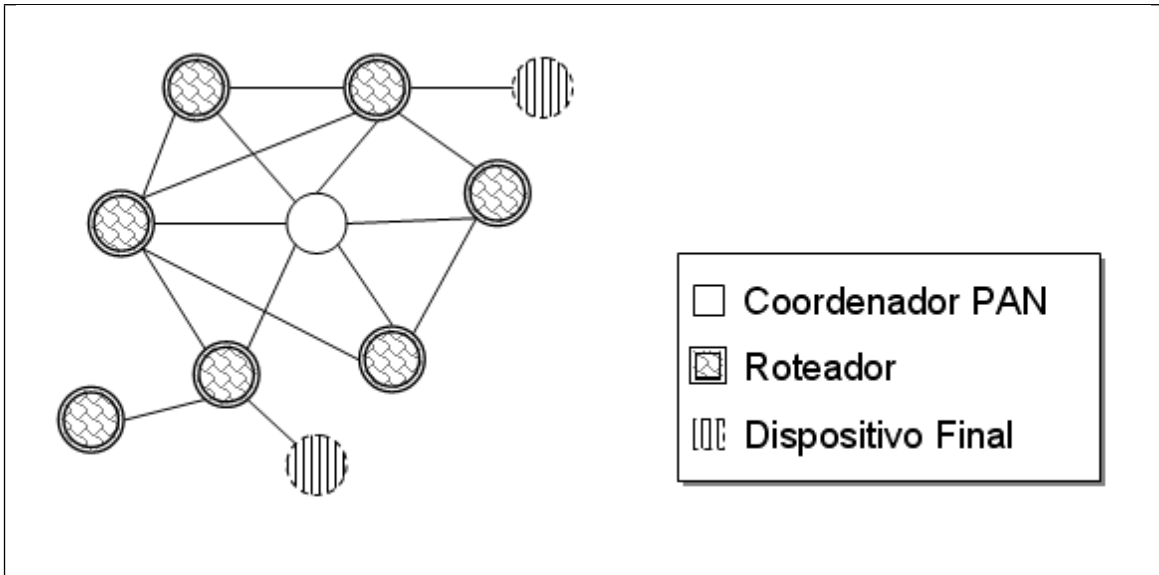


Figura D.2: Exemplo de topologia em Malha.

entrega por múltiplos caminhos permite que ela tenha uma organização automática, onde cada pacote possui um caminho de entrega, sem regras rígidas de roteamento. Esta propriedade é chamada de Auto-organização (*self-organizing*).

Esta topologia é muito usada para aplicações de controle industrial, monitoração e redes de sensores sem fio. A Figura D.2 apresenta um exemplo de uma rede com topologia em Malha.

D.2.3 Topologia de múltiplas árvores

A topologia de múltiplas árvores é um caso particular da topologia em malha. Nesta topologia são formadas múltiplas árvores, derivadas de uma árvore principal que tem o Coordenador PAN como sua raiz. Quando um nó deseja se comunicar com outro nó da mesma árvore, ele encaminha as mensagens para o coordenador da sua árvore, que realiza o roteamento diretamente. Se a mensagem for destinada para um nó de outra árvore, os coordenadores das árvores encaminham as mensagens para o Coordenador PAN, que realiza o roteamento entre todas as árvores da rede.

Em caso de falha do Coordenador PAN, qualquer outro coordenador pode assumir este papel, evitando que toda a rede falhe. A figura D.3 apresenta um exemplo de rede com topologia de múltiplas árvores.

D.3 A camada física (PHY) do padrão 802.15.4

De acordo com o modelo OSI/ISO, a camada física é a mais próxima do hardware e é a responsável pela transmissão de dados no meio físico. No padrão 802.15.4 [6], a camada PHY oferece dois serviços: Serviços de Dados PHY e Serviço de

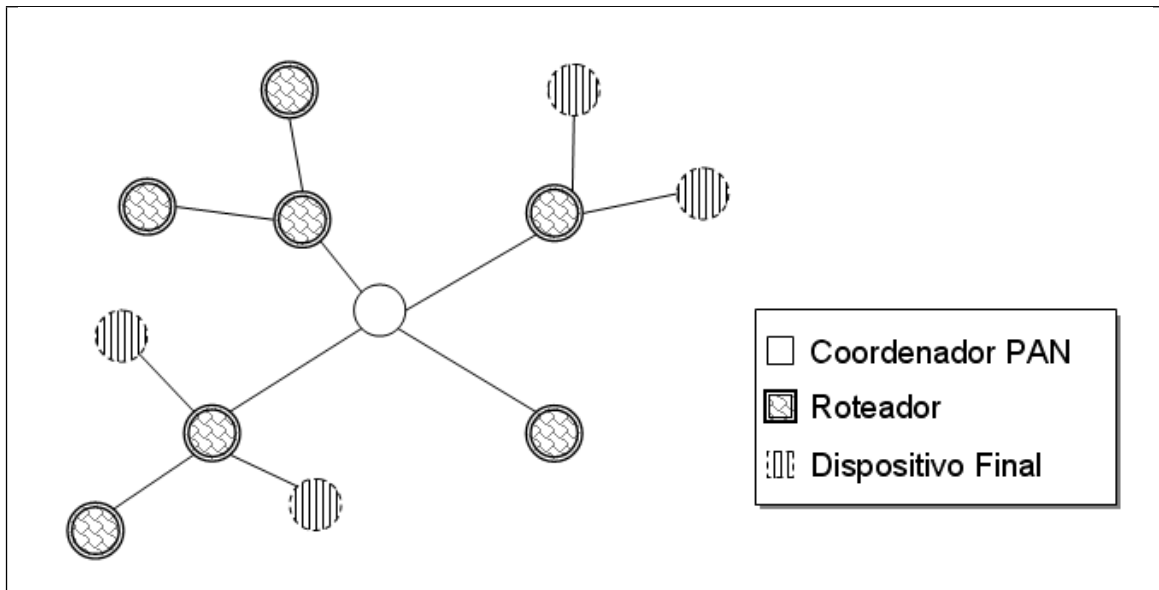


Figura D.3: Exemplo de topologia em Múltiplas Árvores.

Gerenciamento PHY. O Serviço da dados é o responsável pela troca das unidade de dados protocolo PHY (PPDU - *PHY Protocol Data Units*), através de um dos canais do rádio.

A camada PHY do padrão 802.15.4 oferece as seguintes funcionalidades: controle do funcionamento do transceptor de radio, seleção dos canais de transmissão, detecção de energia (ED - *Energy Detection*) em canais de transmissão, indicação de qualidade de link (LQI - *Link Quality Indication*), avaliação de canal livre (CCA - *Clear Channel Assessment*) e transmissão e recepção de pacotes pelo meio físico.

A detecção de energia é uma estimativa da potência do sinal recebido dentro de um canal, representada através de um inteiro de oito bits. Já a indicação de qualidade do link (LQI) é uma medição da qualidade de um pacote recebido e o seu uso fica a cargo das camadas de rede e de aplicação. O LQI também é representado em um inteiro de 8 bits

O número de canais de transmissão varia de acordo com a banda de frequência e a versão do padrão 802.15.4 [81]. Neste texto vamos nos concentrar nas características da versão 2003, pois é a adotada no hardware usado neste trabalho.

A descrição dos canais de transmissão do padrão 802.15.4 na versão 2003, de acordo com a banda de frequência, é a seguinte:

868.0-868.6 MHz (Europa): apenas um canal de comunicação (canal 0), com 2 Mhz de largura de banda e modulação BPSK;

902-928 MHz (EUA e Austrália): dez canais (canais 1-10), com 2 Mhz de largura de banda para cada canal e modulação BPSK;

2400-2483.5 MHz (resto do mundo): dezesseis canais (canais 11-26), com 5 Mhz de largura de banda para cada canal e modulação O-QPSK.

A versão 2003 do padrão oferece dois tipos de camadas físicas, baseadas na técnica *direct sequence spread spectrum* (DSSS): uma que trabalha na banda 868/915 MHz com taxa de transferências de 20 e 40 kbps e -92 dbm de sensibilidade de recepção; e outras na banda de 2450 MHz, com taxa de transferência de 250 kbps e sensibilidade de recepção de -85 dbm.

A Figura D.4, adaptada de [6], apresenta de forma visual as informações da descrição acima.

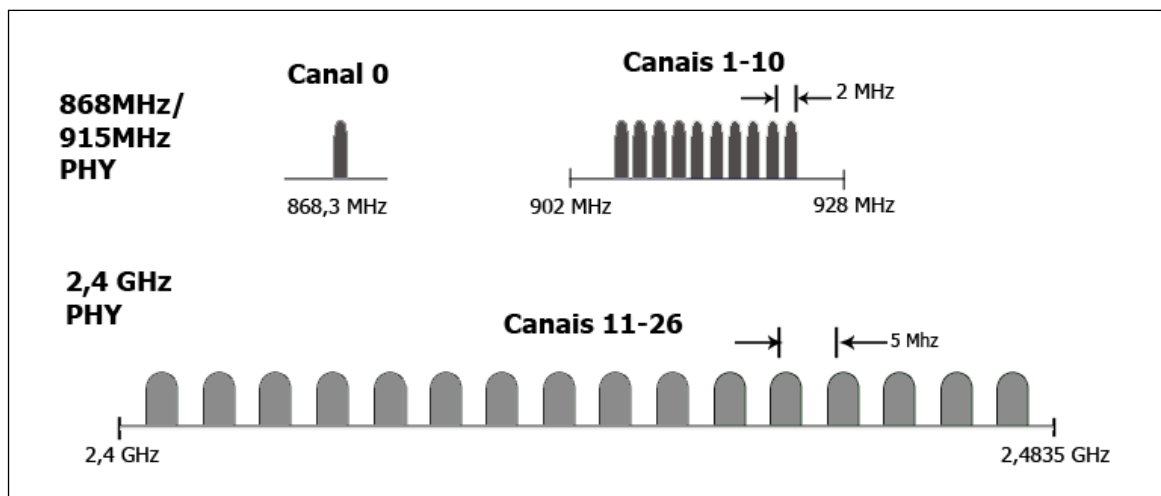


Figura D.4: Canais da camada PHY do padrão 802.15.4, adaptado de [6].

D.4 Camada de Controle e Acesso ao Meio (MAC) do padrão 802.15.4

A camada de Controle e Acesso Meio (MAC) do padrão 802.15.4 é projetada para ser simples e flexível e atender diferentes topologias e tipos de tráfego de dados. Para isso, ela oferece dois serviços: serviço de dados MAC (MCPS - *MAC Common Part Layer*) e serviço de gerenciamento MAC (MLME - *MAC Layer Management Entity*).

O serviço de dados MAC trata os comandos relacionados à troca das unidades de dados MAC (MPDU - *MAC Protocol Data Units*). Já o serviço de gerenciamento MAC controla todos os comandos, respostas, indicações e confirmações usadas para gerenciar uma rede e uma unidade 802.15.4.

A camada MAC do padrão 802.15.4 oferece as seguintes funcionalidades: gerenciamento de quadros de sinalização (*beacon frame*), acesso aos canais, gerenciamento de janelas de tempo com transmissão garantida (GTS - *Guaranteed Time Slot*),

validação de quadros, confirmação de entrega de quadro (*Acknowledged Frame Delivery*), associação e desassociação.

O padrão 802.15.4 foi projetado para contemplar uma ampla gama de aplicações, que possuem vários tipos de tráfego de dados, como, por exemplo: entrega de dados em intervalo fixo, entrega de dados em intervalos alternados, transmissão intermitente, transmissão mediante estímulos de sensores, transmissão constante, entre outros.

Para atender a esta variedade de tipos de tráfegos, a camada MAC do padrão 802.15.4 oferece dois tipos de configurações: com e sem quadros de sinalização. Em uma rede sem quadros de sinalização, os dispositivos podem estabelecer comunicação um com o outro a qualquer hora. O acesso ao canal de comunicação e a conexão são gerenciados através do mecanismo de detecção de colisões CSMA-CA (*Carrier Sense Multiple Access - Collision Avoidance*).

Já em uma rede com quadros de sinalização, o coordenador da rede transmite periodicamente um quadro de sinalização intitulado Beacon, que é utilizado pelos dispositivos para a sincronização e determinação do período de envio e recebimento das mensagens. O intervalo entre mensagens de sinalização é denominado Superframe, que é dividido em dois períodos denominados CAP (*Contention Access Period*) e CFP (*Contention Free Period*).

O CAP é o período destinado a transmissão dos pacotes de dados dos dispositivos sem sincronização, como dispositivos de alarme ou de controle, onde todos os nós que queiram transmitir devem competir com outros dispositivos utilizando o protocolo CSMA/CA MAC. Já o período CFP é destinado à transmissão dos dados com sincronização e baixa latência.

Dentro do intervalo CFP existem janelas de tempo com transmissão garantida, chamadas de GTSs (*Guaranteed Time Slot*), que mapeiam dispositivos ou um conjunto de dispositivos. O coordenador da rede pode alocar até sete destes GTSs. Estes períodos são destinados a aplicações que necessitem coletar dados com intervalos de tempo fixos garantidos.

São previstos quatro tipos de quadros na camada MAC do padrão 802.15.4:

Beacon frame

usado pelo coordenador para envio de mensagens de sinalização usadas para sincronização da rede;

Data frame

utilizado por todos os dispositivos da rede, em todas topologias. Este quadro é usado pela camada MAC para transmissão de dados recebidos das camadas mais altas da pilha de protocolo;

Acknowledgment frame

O *frame* de ACK é transmitido pela camada MAC para confirmar a recepção de dados ao emissor. Este quadro de retorno somente é gerado e enviado se o pacote de origem contiver a solicitação desta confirmação e se o cálculo de CRC for realizado corretamente;

MAC command frame

O *frame* de comando MAC é gerado pela camada MAC e é usado para troca de comandos MAC. Os comando MAC são usados para ações como: pedidos e respostas de associação na rede, notificação de desassociação da rede, sincronização com coordenador e requisição de dados.

D.5 Comparação 802.15.4/ZigBee e BlueTooth

O padrão IEEE 802.15.1/Bluetooth é muito popular e está presente na maioria dos aparelhos celulares disponíveis no mercado. Ele também é utilizado em milhares de produtos [82] em várias áreas, como entretenimento digital, periféricos de computadores, equipamentos médicos e de audiovisual.

O padrão BlueTooth é mantido pela BlueTooth SIG (*Special Interest Group*), uma associação privada e sem fins lucrativos fundada em 1998 e que é formada por mais de onze mil empresas. Entre as suas fundadoras estão empresas como Ericsson, Sony, Apple, Intel, Panasonic e Nokia.

A primeira distinção entre os dois padrões é a taxa de transmissão de dados. Enquanto o 802.15.4/ZigBee apresenta baixas taxas de transmissão, que variam entre 20 e 250 kbps, de acordo com a frequência de operação; o BlueTooth opera com taxa mais altas, com 1 mbps na versão 1.2 do padrão e até 3 mbps na versão 2.0 com a adição de uma tecnologia de transmissão de dados chamada EDR (Enhanced Data Rate) e por isso chamada de 2.0 + EDR.

Outra diferença importante entre os dois é o alcance da transmissão de rádio. Enquanto o alcance do 802.15.4/ZigBee pode chegar até 125 metros em ambientes abertos, o do BlueTooth é determinado pela classe do rádio: classe 1 com alcance de até 100 metros, classe 2 com alcance de até 10 metros e classe 3 com alcance de até 1 metro. A divisão em classes permite maior flexibilidade ao padrão, de acordo com o contexto que será usado.

Em relação ao consumo de energia, o 802.15.4/ZigBee apresenta maior economia, com um consumo médio de 60 mW para transmissões até 125 metros, contra 100 mW do BlueTooth classe 1, com transmissões de até 100 metros. Em geral, os aparelhos com BlueTooth preveem recargas periódicas em suas baterias, enquanto os dispositivos 802.15.4/ZigBee funcionam por longos períodos, de meses e até anos,

sem recargas.

Outra diferença muito importante no contexto deste trabalho é o tempo de despertar dos dois dispositivos. Pois, para fins de economia de energia, ambos entram periodicamente em modos de espera e baixo consumo de energia, enquanto não há troca de dados. Para sair deste estado, um dispositivo BlueTooth leva cerca de 3 segundos, enquanto o 802.15.4/ZigBee leva menos de 15 ms.

Outra diferença fundamental para este trabalho é o tamanho das redes que são possíveis de serem criadas com cada tecnologia. Enquanto o Bluetooth suporta redes de até no máximo 8 dispositivos, o 802.15.4/ZigBee suporta redes com até 65536 dispositivos.

O resumo deste comparativo é apresentado na tabela D.5

Tabela D.1: Quadro com o resumo do comparativo entre o 802.15.4/ZigBee e o BlueTooth.

Característica	BlueTooth	802.15.4/ZigBee
Taxa de transmissão	1 mpbs (versão 1.2), 3 mbps (versão 2.0 + EDR)	Entre 20 e 250 kbps
Consumo de energia	1mW (classe 1), 10 mW (classe 2) e 100 mW (classe 3)	60 mW
Alcance do rádio	1m (classe 1), 10 m (classe 2) e 100m (classe 3)	125 m
Tempo para acordar	3 segundos	menos de 30 ms
Frequência de operação	2.4 GHz	868 MHz (Europa) 900-928 MHz (USA), 2.4 GHz (resto mundo)
Tempo para entrar na rede	10 seg	Menos de 30 ms
Dispositivos na rede	Até 8	Até 65536

Apêndice E

SAMCRA

Para realizar os experimentos apresentados nesse trabalho, desenvolvemos um sistema chamado de SAMCRA (Sistema de Automação, Monitoração e Configuração de Redes Ad Hoc) [29]. Ele é composto de três módulos independentes que oferecem serviços de monitoração, configuração da rede, coleta de dados, rastreamento de mensagens e execuções automatizadas para avaliação experimental da REPI-A.

Para garantir que a execução destes módulos não causa interferências na execução da REPI-A, toda a comunicação dos nós remotos com SAMCRA é feita através da rede TCP/IP cabeada, isolando a REPI-A.

Os três módulos que compõe SAMCRA são apresentados nas próximas Seções.

E.1 Módulo de Coleta de dados

Este módulo é responsável pela coleta dos dados do uso da REPI-A. Esta coleta é feita através das instancias da aplicação mensageira executadas por todos os nós da rede. Elas coletam as informações de cada Tmote e as enviam a cada dez segundo para este módulo, através da tecnologia RMI do Java.

Esses dados são consolidados em nove tabelas diferentes, apresentadas a seguir.

E.1.1 Tabela Dados de participação por Tmote

A tabela Dados de participação por Tmote é acessível na segunda aba de SAMCRA, como mostra a Figura E.1. Nela são consolidados os dados sobre a participação de cada Tmote na rede. Ela contabiliza os valores de trinta e cinco variáveis coletadas por cada nó e que contém informações sobre as trocas de mensagens, erros e participação na rede de cada dispositivo. Na primeira coluna de cada linha está o endereço TinyOs do Tmote que enviou as informações. A seguir, são apresentadas as descrições destas variáveis:

Servidor de Instrumentação																	
Parametrização & Execução Automatizada																	
Coleta de dados						Campos coincidentes por quantidade						Endereçadas Rx (US)				Desc. Perfil R:	
ID	RX Total	Stat Rx	Pre... RX	Cu... Rx	Re... Rx	Ou... Rx	En... total	Pre... TX	Stat Tx	Cu... Tx	Re... Tx	De... Total	De... HTL	De... Pró...	De... Me...	Des... Perfil	De... Enc...
1	825	1	249	0	0	0	94	94	0	0	0	576	0	125	451	0	0
2	1530	1	351	0	0	0	0	0	0	0	0	1352	0	0	1179	173	0
3	965	1	339	0	0	0	0	0	0	0	0	706	0	0	626	80	0
4	1379	1	258	0	0	0	90	90	0	0	0	1210	0	226	895	89	0
5	1006	1	345	0	0	0	0	0	0	0	0	661	0	0	661	0	0
6	1215	1	347	0	0	0	0	0	0	0	0	868	0	0	868	0	0
7	1107	1	346	0	0	0	0	0	0	0	0	851	0	0	761	90	0
8	926	1	342	0	0	0	0	0	0	0	0	670	0	0	584	86	0
9	1291	1	348	0	0	0	0	0	0	0	0	1114	0	0	943	171	0
10	1055	1	345	0	0	0	0	0	0	0	0	710	0	0	710	0	0
11	1243	1	349	0	0	0	0	0	0	0	0	983	0	0	894	89	0
12	988	1	346	0	0	0	0	0	0	0	0	730	0	0	642	88	0

Figura E.1: Visão parcial da tabela que consolida os dados sobre as mensagens trocadas por cada Tmote.

Mensagens recebidas

Número total de mensagens recebidas por cada Tmote e o número de mensagens de acordo com o seu tipo: mensagens de status do Tmote; mensagens para canais públicos; mensagens para canais customizados; mensagens para o canal reservado; e mensagens de outro tipo, que indicam se há mensagens de outras redes TinyOS interferindo na REPI-A;

Mensagens enviadas

A aplicação contabiliza o número total de mensagens enviadas por cada Tmote, além do número de mensagens de acordo com os mesmos tipos das mensagens recebidas;

Mensagens descartadas

Número de mensagens recebidas por um Tmote que não foram reencaminhadas para a rede, que neste texto são chamadas de mensagens descartadas. Além do número total, a aplicação contabiliza o número de mensagens descartadas de acordo com o motivo do descarte: descarte em razão do fim do valor do HTL; descarte por ter sido uma mensagem enviada pelo próprio Tmote que a recebeu; descarte por não atender aos critérios de controle de pacotes, apresentados na Seção E.2; descarte pela mensagem não atender aos critérios de encaminhamento do Protocolo REPI-A;

Mensagens não exibidas para o usuário

Número de mensagens que foram recebidas pelo Tmote mas não foram exibidas para o usuário por não haver casamento de campos na parte de interesses (distribuição Zipf) do prefixo da mensagem, ou seja, que não eram endereçadas para ele. Este valor é contabilizado em três variáveis, de acordo com o tipo

de mensagem: mensagem de canal público, mensagem do canal reservado e mensagem de canal customizado;

Mensagens encaminhadas

Número de mensagens recebidas e reenviadas para a rede pelo Tmote. Existem dois tipos de mensagens encaminhadas na rede: o encaminhamento padrão, em que houve algum casamento de campos tanto no prefixo C quanto na parte de interesses da mensagem e do Tmote receptor; e o encaminhamento colaborativo, em que o Tmote repassa a mensagem tendo somente casamento no prefixo C, ou seja, a mensagem não foi aproveitada pela aplicação, mas é encaminhada para colaborar com a rede;

Erros no Tmote

Número total de erros ocorridos no Tmote, conforme tabela de códigos apresentada na Tabela 2.3, e uma variável para cada tipo de erro, em um total de sete.

Novos interesses customizados

Número de canais customizados criados pelo usuário;

Número médio de saltos

Média aritmética do número de saltos de todas as mensagens recebidas, inclusive as descartadas. Além da média aritmética do número de saltos apenas das mensagens entregues aos seus destinatários;

Tempo médio de entrega de mensagens

Média aritmética do tempo, em milissegundos, de entrega das mensagens que circularam na rede. O tempo é calculado com o *overhead* da aplicação messageira e sem ele. No segundo caso, o tempo é calculado pelo próprio Tmote.

E.1.2 Tabelas com detalhamento de mensagem recebidas por Tmote

As próximas seis abas do servidor contém tabelas que exibem o detalhamento do número de mensagens recebidas por cada Tmote. As três primeiras apresentam a quantidade de mensagens recebidas a partir dos seus vizinhos imediatos, ou seja, de acordo com o último Tmote por onde a mensagem passou; e a três últimas detalham a quantidade de mensagens de acordo com o Tmote que gerou esta mensagem na rede, ou seja, o envio fim-a-fim.

Em ambos os casos, essas três tabelas apresentam os seguintes nomes e contém as seguintes informações: Endereçadas Rx, apresenta a quantidade de mensagens com casamento de campos de características do prefixo; Descartadas Rx, apresenta o

Servidor de Instrumentação									
Parametrização & Execução Automatizada		Coleta de dados		Campos coincidentes por quantidade			Endereçadas Rx (US)		Desc. Perfil R:
Mote	Gaus 0	Gaus 1	Gaus 2	Gaus 3	Gaus 4	Gaus 5	Gaus 6	Gaus 7	Gaus 8
1	0	542	180	0	0	0	0	0	0
2	840	404	319	0	0	0	0	0	0
3	201	452	4	329	0	0	0	0	0
4	448	399	332	0	0	0	0	0	0
5	0	318	474	235	0	0	0	0	0
6	0	265	670	307	0	0	0	0	0
7	335	533	263	0	0	0	0	0	0
8	279	478	190	0	0	0	0	0	0
9	659	384	281	0	0	0	0	0	0
10	0	542	545	0	0	0	0	0	0
11	360	353	279	0	278	0	0	0	0
12	277	271	238	0	223	0	0	0	0
13	0	1333	0	351	0	0	0	0	0
14	243	0	549	0	0	0	0	0	0

Figura E.2: Uma das tabelas que detalham o número de mensagens recebidas por Tmote.

número de mensagens descartadas; Todas Rx, apresenta a quantidade de mensagens recebidas no total.

Nesta tabela, cada linha representa o Tmote que recebeu as mensagens e nas colunas estão os números dos Tmote que as enviaram. As tabelas que representam o último salto são identificadas com a sigla US, e as que representam o envio fim-a-fim são representadas com a sigla FF.

Com as informações obtidas nestas tabelas é possível criar um grafo que apresente a troca de mensagens da rede, como o mostrado na Seção 4.5.1. Neste grafo, cada Tmote é um vértice e os valores das trocas de mensagens entres estes são as arestas.

E.1.3 Tabela Quantidade de Campos coincidentes por Tmote

A Tabela Quantidade de Campos coincidentes por Tmote é acessível na nona aba do Servidor REPI, como mostra a Figura E.3. Ela consolida o número de campos coincidentes do prefixo local com os prefixos das mensagens recebidas. Ela possui dezessete colunas, que indicam quantos campos foram coincidentes nas duas partes do prefixo: os campos biométricos e os campos de interesse.

A primeira coluna apresenta o número do Tmote que enviou os dados, e a partir da segunda, são mostradas dez colunas que mostram quantas mensagens foram recebidas com cada quantidade de características coincidentes, começando em zero e indo até nove, em números sequenciais incrementados de uma unidade. As próximas seis colunas indicam o número de interesses coincidentes, começando em zero e indo até cinco, também de forma sequencial.

Portanto, se, por exemplo, um nó receber vinte mensagens que não tenham nenhuma característica coincidente, trinta mensagens com uma e mais quarenta

com três, a sua linha na tabela será preenchida com os valores 20, 30 e 40 e todas as outras colunas com o valor zero.

Mote	Gaus 0	Gaus 1	Gaus 2	Gaus 3	Gaus 4	Gaus 5	Gaus 6	Gaus 7	Gaus
1	0	0	54	20	0	11	0	0	0
2	0	132	87	66	0	0	0	0	0
3	34	0	144	0	0	44	0	0	0
4	0	0	50	142	58	0	0	0	0
5	0	61	83	121	0	0	0	0	0
6	0	0	127	89	0	0	0	0	0
7	0	54	82	41	0	0	0	0	0
8	0	0	130	46	0	0	0	0	0
9	0	43	62	62	0	0	0	0	0
10	40	108	0	102	0	0	0	0	0
11	0	0	86	53	0	58	0	0	0
12	0	34	39	96	0	0	0	0	0
13	0	0	94	51	153	0	0	0	0
14	0	27	57	45	37	48	0	0	0
15	0	44	138	0	0	0	0	0	0

Figura E.3: Visão parcial da tabela que consolida o número de campos coincidentes nos prefixos das mensagens recebidas .

E.1.4 Tabela com rastreamento de mensagens (*trace*)

Esta tabela, acessível na décima aba do servidor REPI, como mostra a Figura E.4, consolida metadados sobre o recebimento de todas as mensagens da rede.

O principal objetivo do rastreamento de mensagens é permitir o acompanhamento da troca de mensagens na rede de forma detalhada, permitindo a análise de todas as mensagens que circularam na rede. Além disso, através da manipulação desses dados, é possível inferir os impactos causados por alterações nos parâmetros da rede. Por exemplo, é possível verificar qual seria o impacto ao diminuir o valor de um parâmetro como HTL, filtrando as mensagens que não seriam recebidas com o novo valor.

A Figura E.4 apresenta o trace de uma execução em uma REPI. Nesta tabela são detalhados os seguintes campos:

Hora de Recebimento

Horário, com dia, mês, ano, hora, minutos, segundo e milissegundos, de recebimento da mensagem;

Tmote Receptor

Endereço TinyOS do Tmote que recebeu a mensagem;

Tmote Gerador

Endereço TinyOs do Tmote que originou a mensagem(fim-a-fim);

Servidor de Instrumentação					
Parametrização & Execução Automatizada		Coleta de dados	Campos coincidentes por quantidade	Endereçadas Rx (US)	Descartadas R:
Horário Recebimento	Tmote Receptor	Tmote Gerador	Último salto	HTL	Núme
20/02/10 - 19:02:08,421	14	11	11	20	
20/02/10 - 20:02:28,937	10	11	11	20	
20/02/10 - 20:02:28,937	6	11	11	20	
20/02/10 - 19:02:08,500	12	11	6	19	
20/02/10 - 19:02:08,421	13	11	11	20	
20/02/10 - 20:02:10,703	2	11	11	20	
20/02/10 - 18:59:51,718	17	11	11	20	
20/02/10 - 20:02:28,937	8	11	11	20	
20/02/10 - 20:02:29,031	5	11	10	19	
20/02/10 - 18:59:51,703	19	11	11	20	
20/02/10 - 19:02:08,578	14	11	13	19	
20/02/10 - 20:02:29,093	6	11	13	19	
20/02/10 - 20:02:29,093	5	11	13	19	
20/02/10 - 20:02:10,859	2	11	13	19	
20/02/10 - 20:02:29,109	6	11	17	19	

Figura E.4: Tabela com o trace de uma execução na rede REPI.

Último Tmote

Endereço TinyOs do último Tmote por onde a mensagem passou (último salto);

HTL

Valor do HTL da mensagem ao chegar no destino;

Número Seq.

Número sequencial da mensagem;

Descarte

Indica se o pacote foi descartado, pode ter dois valores: Sim ou Não;

Tipo Descarte

Caso o pacote tenha sido descartado, indica o motivo;

Campos Coinc. Gaussianos

Indica quantas campos do prefixo C foram coincidentes;

Campos Coinc. Interesses

Indica quantos campos foram coincidentes na parte de interesses do prefixo;

E.2 Módulos de Configuração e Execução Automatizada

O Módulo de Parametrização e de Execução Automatizada, apresentado na Figura E.5, possui duas funções: parametrizar a rede e permitir execuções automatizadas, que promovem o envio de mensagens na rede sem a intervenção humana. A interface

gráfica deste servidor é dividida em quatro painéis, que são descritos a seguir na ordem em que aparecem na interface, de cima para baixo.

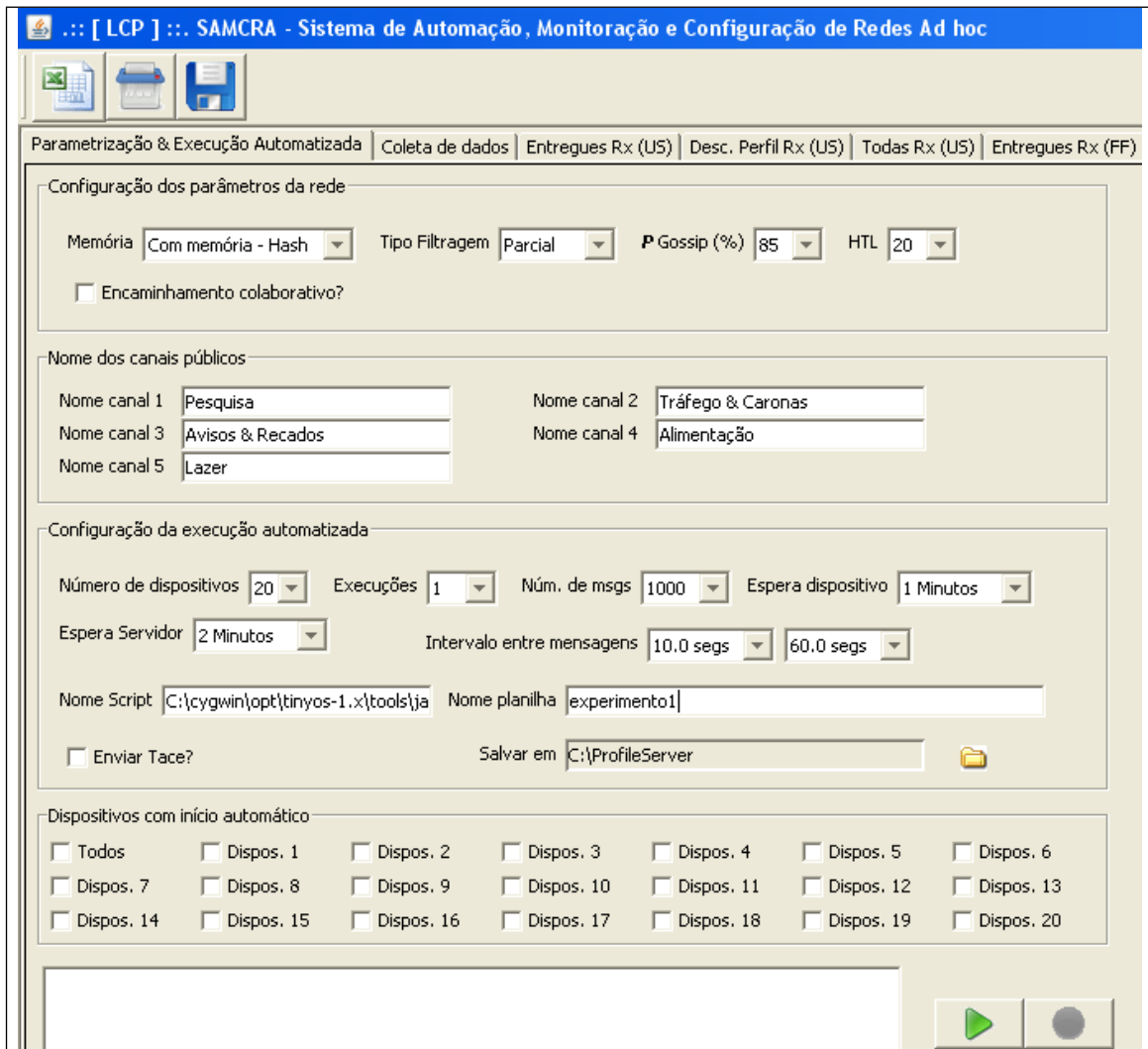


Figura E.5: Tela com as opções de parametrização e execução automatizada na REPI-A.

E.2.1 Configuração dos parâmetros da REPI-A

Neste painel o usuário determina os parâmetros que definirão o comportamento da REPI-A. Estas informações são transmitidas para os nós remotos através de um arquivo XML, que é disponibilizado em um servidor web e é lido por cada aplicação mensageira durante sua inicialização.

O primeiro destes parâmetros é a memória, que define o tipo de controle que é usado para evitar a duplicação de mensagens na rede. Para isso, os Tmotes controlam as mensagens que passam por eles e descartam aquelas que são repetidas. A versão atual possui três valores para este campo: Sem Memória, Número sequencial e Hash.

O próximo parâmetro determina qual será a política de filtragem de mensagens de acordo com a comparação do prefixo da mensagem e do Tmote que a recebe. Esta configuração impacta diretamente o roteamento de mensagens na rede, pois ela determina qual será o critério para o encaminhamento de mensagens da REPI-A. Ela pode ter três valores:

Sem filtro

Nesta configuração não há filtragem de pacotes e todos os pacotes recebidos são encaminhados. Esta configuração é similar a um algoritmo de inundação (*flooding*);

Parcial

Nesta configuração os pacotes são encaminhados se pelo menos um dos campos da parte de características do prefixo da mensagem for igual ao do prefixo do Tmote que a recebeu. Em caso contrário, a mensagem é descartada;

Completo

Nesta configuração os pacotes só são encaminhados se todo o conjunto de características dos dois prefixos for igual, caso contrário a mensagem é descartada;

Gossip

Nesta configuração os pacotes são encaminhados seguindo um algoritmo probabilístico (*gossip*), em que cada mensagem tem um probabilidade P de ser encaminhada. O parâmetro P é determinado pela opção “ P Gossip (%)”.

A próxima configuração indica qual será o HTL (*Hops to live*) da mensagem, ou seja, por quantos Tmotes a mensagem pode passar antes de ser descartada.

A quarta e última configuração determina se haverá encaminhamento colaborativo na rede. Ou seja, determina se os Tmotes encaminharão as mensagem que atendem aos critérios de filtragem de mensagens mas não são endereçadas para eles, atuando para colaborar com a formação da rede.

E.2.2 Nome dos canais públicos

Neste painel o usuário pode definir os nomes dos cinco canais públicos que serão comuns a todos os nós da REPI-A, como apresentado na Seção 3.

E.2.3 Configuração da execução automática

Os dois painéis inferiores, chamados “Configuração da execução automática” e “Tmotes com início automático”, fazem parte da ferramenta de execução automatizada desenvolvida para a realização dos experimentos apresentados neste trabalho.

O principal objetivo desta funcionalidade é permitir a realização de execuções na rede em que sejam enviadas milhares de mensagens por até vinte Tmotes simultaneamente sem nenhuma intervenção do usuário.

Além disso, a ferramenta também permite que essas execuções sejam repetidas várias vezes, para aumentar a confiabilidade dos dados coletados. Os resultados dos experimentos realizados são apresentados no Capítulo 4, nesta Seção apresentamos apenas as funcionalidades da ferramenta.

Para realizar uma execução automatizada na REPI-A, o usuário precisa especificar onze parâmetros distintos, que determinarão como será feita esta execução, a saber:

Número de Tmotes

Determina quantos Tmotes serão usados na execução automatizada;

Execuções

Determina quantas execuções serão realizadas;

Núm de Msgs

Determina quantas mensagens os Tmotes com início automático enviarão;

Espera Tmotes

Determina quanto tempo cada Tmote irá esperar para notificar ao servidor que sua execução terminou e que uma nova execução de experimentos pode ser iniciada;

Espera Servidor

Determina quanto tempo o servidor vai esperar para iniciar uma nova execução de execuções depois que o último Tmote comunicar que terminou a sua execução;

Intervalo entre mensagens

Determina o intervalo mínimo e máximo, em segundos para o envio de duas mensagens consecutivas na rede. O tempo de envio será um número randômico entre o mínimo e o máximo especificado;

Nome Script

Nome do script que será executado para iniciar as aplicações messageiras nos nós remotos;

Nome planilha

Nome da planilha eletrônica que será gravada com todos os dados coletados durante os experimentos. Quando mais de uma execução é realizada no mesmo experimento, a aplicação cria automaticamente uma planilha para cada

execução e, ao final da última execução, cria uma planilha com os dados consolidados, com média aritmética e desvio padrão;

Salvar em

Determina em qual diretório as planilhas geradas serão salvas;

Enviar Trace?

Determina se os Tmotes remotos enviarão o trace da sua execução. O trace é salvo em um arquivo de texto separado por vírgulas csv (*comma separated values*);

Tmotes com início automático

Determinar quais serão os Tmotes com início automático, ou seja, quais Tmote enviarão mensagens durante a execução. Os Tmotes sem início automático podem atuar na rede apenas como receptores e repetidores de mensagens de outros Tmotes.

A execução automatizada é realizada em 8 passos, que são detalhados a seguir:

1. O usuário configura os parâmetros mostrados anteriormente, ajustando os parâmetros da REPI-A e da execução automatizada. Após esta configuração, o usuário aciona o botão de execução, apresentado na Figura E.5 com um ícone verde, na parte inferior da interface;
2. As configurações feitas pelo usuário são salvas em um arquivo XML e disponibilizadas em um servidor Web, que é acessado pelas aplicações mensageiras durante a sua inicialização;
3. O script, especificado pelo usuário no campo “Nome Script” é executado. Esse script é o responsável pela inicialização das aplicações mensageiras nos nós remotos. Após a execução do script, o servidor entra em modo de espera, aguardando que todos os Tmotes com início automático notifiquem o término de suas execuções;
4. Os Tmotes com endereço TinyOS com execução automática enviam o número de mensagens especificado pelo parâmetro “Núm de Msgs”, com a frequência especificada pelo parâmetro “Intervalo entre mensagens”;
5. Ao terminar de enviar todas suas mensagens, os Tmotes com início automático aguardam o período especificado pelo parâmetro “Espera Tmotes” e, ao terminar o prazo, enviam uma notificação de término para o servidor. Esta espera é importante para garantir que todas as mensagens da rede serão entregues, visto que existe um atraso na entrega que, dependendo da quantidade de mensagens e a contenção na rede, pode chegar até a algumas horas;

6. Após receber todas as notificações de término, o servidor aguarda o tempo especificado pelo parâmetro “Espera Servidor”. Ao fim deste período, o servidor considera que a execução de experimentos foi encerrada;
7. O servidor salva todos os dados coletados na execução em uma planilha eletrônica no disco e inicia uma nova execução;
8. Ao final da última execução, o servidor cria uma planilha eletrônica com a média aritmética e o desvio padrão dos dados coletados em todas as execuções.

Apêndice F

Sistema Operacional TinyOS

O TinyOS [83] é um sistema operacional desenvolvido especificamente para controlar dispositivos de comunicação em redes de sensores sem fio. Ele é codificado com a linguagem de programação NesC [84], um dialeto da linguagem C para programação em redes de sensores.

Com o crescimento e a popularização do TinyOS, foi criado um consórcio internacional, chamado TinyOS Alliance [85], cujo objetivo é manter o desenvolvimento do sistema operacional e da comunidade envolvida com ele.

A versão 1.1 do TinyOS é utilizada neste trabalho, pois esta era a versão mais madura e melhor documentada no início da sua elaboração. Atualmente, o TinyOS está na versão 2.1.0, que possui uma série de mudanças em relação a versão 1.1 e não tem compatibilidade total de código. Neste trabalho são apresentadas as características da versão 1.1.

F.1 Arquitetura do TinyOS 1.1

A arquitetura do TinyOS é baseada em componentes, que são construídos separadamente e depois agrupados e compilados para criar um único arquivo executável, com a aplicação e o sistema operacional. Um componente é um módulo de software reusável que realiza uma tarefa específica, como, por exemplo, controlar os leds do Tmote, acesso a memória, enviar e receber pacotes ou contabilizar o tempo.

Uma das principais vantagens desse modelo é que as aplicações só utilizam os módulos do sistema operacional que necessitam, pois somente esses serão adicionados ao código final durante a compilação.

Os componentes do TinyOS podem ser de dois tipos: módulos, que implementam as funcionalidades do componente; e configurações, que são usados para especificar a forma como os componentes são ligados para construir uma aplicação. Esse processo de ligação de componente é chamado de amarração (*wiring*).

A arquitetura de componentes do TinyOS é hierárquica, com componentes que estão em níveis mais baixos, ou seja, mais próximos do hardware; e outros que estão em níveis mais altos, ou seja, próximos ou na própria camada de aplicação.

Todo componente possui uma interface que é a descrição dos serviços que ele pode realizar e é o seu único ponto de acesso. Com isso, cada componente provê uma interface, para que ele seja ligado a outros componentes, e utiliza as interfaces de outros componentes. Isso faz com que um módulo não precise conhecer o funcionamento interno de outro para utilizá-lo, basta conhecer sua interface. Dessa forma, é possível que um módulo tenha seu funcionamento interno alterado sem impactar o restante da aplicação, desde que sua interface seja a mesma. Para aumentar a eficiência em tempo de execução, todas as ligações de componentes são estáticas.

Cada interface oferece uma série de funções que podem ser de dois tipos: comandos e eventos. Os comandos e eventos, assim como as interfaces, também são fornecidos e utilizados pelos componentes. A seguir são descritas as principais características e diferenças entre eventos e comandos:

Comandos

Os Comandos são funções que devem ser implementados pelos componentes que o fornecem para outros componentes. Em geral, eles são utilizados para realizar ações no sistema operacional e no hardware do Tmote, tal como mudar o estado de um led ou enviar um pacote na rede. Por isso, geralmente são executados na camada de aplicação e descem na hierarquia de componentes até os níveis mais baixos;

Eventos

Os Eventos são funções que devem ser implementadas pelo usuário do componente e geralmente são utilizados para avisar ao programador alguma ocorrência de hardware ou software, como a chegada de um pacote da rede ou o disparo de algum contador de tempo. Com isso, os eventos geralmente são gerados nos níveis mais baixos da hierarquia de componentes e vão subindo até serem tratados pela aplicação.

F.2 Modelo de concorrência do TinyOS

O TinyOS é um sistema operacional monoprocessado, pois executa apenas uma aplicação de cada vez. As aplicações são compostas pelos componentes do sistema operacional que são necessários para o seu funcionamento (o escalonador de tarefas é sempre obrigatório) e os componentes funcionais desenvolvidos para a aplicação.

O modelo de concorrência do TinyOS é simples e prevê apenas dois tipos de fluxo de execução, chamados de tarefas (*tasks*) e tratadores de eventos de hardware

(*hardware event handlers*).

As tarefas são funções que não são preemptivas entre si: uma vez escalonadas, elas executam até terminar (*run to completion*) e não podem ser removidas do processador por outra tarefa. Por isso, elas devem ser curtas para evitar a inanição (*starvation*) de outras tarefas. Além disso, elas também não podem bloquear o uso de recursos nem entrar em modo de espera ocupada.

O escalonador de tarefas padrão do TinyOS é implementado através de uma política FIFO (*First In First Out*) e executa uma tarefa sempre que o processador está disponível.

Os tratadores de eventos de hardware são funções executadas em resposta a uma interrupção de hardware e, ao contrário das tarefas, são preemptivos entre si e também em relação a tarefas. Ou seja, ao receber uma interrupção de hardware, o TinyOS executa um tratador de eventos que será executado imediatamente, tirando da execução uma tarefa ou outro tratador de eventos que esteja sendo executado.

Em função da preempção das tarefas e eventos, o TinyOS está sujeito a condições de corrida no acesso aos dados compartilhados. Existem duas estratégias para evitar estas situações: acessar variáveis compartilhadas exclusivamente em tarefas, que não são preemptivas entre si; ou utilizar a cláusula *atomic* da linguagem NesC, que permite a criação de blocos de código com execução atômica, ou seja, com garantia de que serão executadas na sequência exata em que estão escritas sem serem interrompidas.

Para evitar a criação de ciclos de chamadas, os eventos podem executar comandos, mas comandos não podem sinalizar eventos.

F.3 Operações divididas em fase

Como visto na subseção anterior, as tarefas do TinyOS não são preemptivas entre si, o que faz com que elas tenham que ser curtas para evitar que elas sofram de inanição de recursos. Para resolver esse problema, o TinyOS oferece um modelo de execução operações divididas em fases (*split-phase operations*).

Neste modelo, as operações são realizadas em duas etapas: a requisição da execução de uma tarefa e o aviso do seu término. Em geral a requisição é feita através de um comando e o aviso do término é feito através de um evento.

Um exemplo dessa situação é o envio de um pacote na rede: ao enviar um pacote, o comando *send* (enviar) deve ser executado pela aplicação e, ao concluir o envio, um evento chamado *sendDone* (envio concluído) é disparado e a aplicação recebe o resultado da operação e pode tomar as medidas cabíveis.

F.4 Modelo de memória do TinyOS

O modelo de memória do TinyOS é estático e por isso não há alocação dinâmica de memória, não há ponteiros para funções e todas as chamadas de funções são definidas em tempo de compilação. Com isso, é possível ao compilador executar otimizações mais agressivas do que em modelos dinâmicos, além de evitar situações de desperdício de memória que ocorrem quando há alocação dinâmica em uma memória muito fragmentada.

A aplicação conta com uma pilha que é compartilhada por todas as suas funções e que armazena todas as variáveis locais da aplicação.

Apêndice G

Tmote Sky

O Tmote Sky [2] é um dispositivo de comunicação que utiliza ondas de rádio frequência para a formação de redes sem fio. Ele é compatível com padrões de comunicação da indústria: padrão USB (*Universal Serial Bus*) para a comunicação com fio; e ZigBee (IEEE 802.15.4) para a comunicação sem fio. O que permite sua interoperabilidade com outros tipos de dispositivos, inclusive com computadores pessoais.

Além da capacidade de formação de redes de comunicação de propósito geral, o Tmote Sky também possui dispositivos de sensoriamento ambiental, que permitem a coleta de informações relacionadas ao ambiente onde ele está instalado.

Em sua versão padrão, o Tmote Sky possui sensores de medição de temperatura, umidade, intensidade de luz e intensidade de luz no espectro necessário para a realização de fotossíntese vegetal, além de possuir conectores para adição de sensores adicionais. Por isso, as redes formadas por estes dispositivos também são tratadas na literatura como RSSF- Redes de Sensores Sem Fio (WSN - *wireless sensor network*).

As principais características técnicas do Tmote Sky são:

- Transceptor sem fio Chipcon, com largura de banda de 250 kbps, compatível com 802.15.4 e frequência de operação de 2.4GHz;
- Interoperabilidade com outros dispositivos 802.15.4;
- Microcontrolador MSP430 [86] da Texas Instruments, com frequência de relógio (*clock*) de 8MHz, 10k de memória RAM (*Random Access Memory*) e 48k de memória Flash;
- Antena integrada com alcance de 50 metros para ambiente fechados (*indoor*) e 125 metros para ambientes abertos (*outdoor*);
- Programação e coleta de dados através da interface USB;

- Dois leds (diodo emissor de luz - *light emitting diode*) para indicar a transmissão e recepção de sinal da porta USB. Três leds para uso livre da aplicação e depuração;
- Botão para reiniciar o Tmote. Botão usuário, que pode ser customizado pela aplicação;
- Suporte ao TinyOS.

A Figura G.1, adaptada de [2], apresenta a parte frontal de um Tmote Sky e a localização de seus principais elementos de hardware.

O Tmote Sky é projetado para operar com tensões elétricas que variam de 2,1V até 3,6V com corrente contínua. Essa alimentação é obtida através de duas pilhas AA ou através da porta USB.

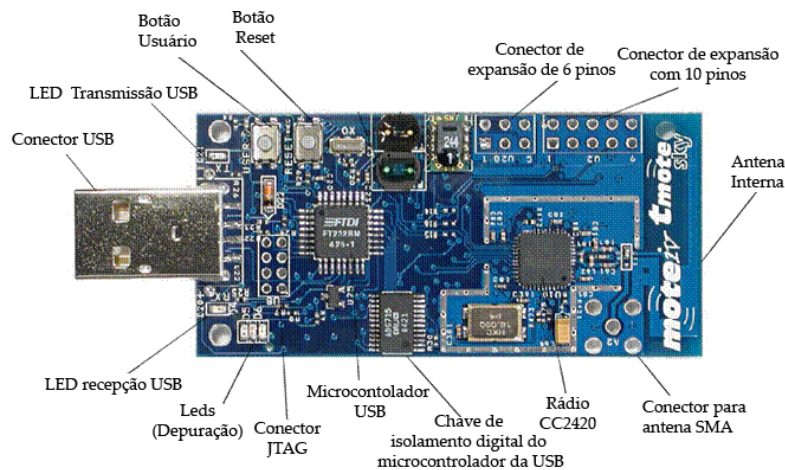


Figura G.1: Dispositivo de comunicação Tmote Sky, adaptado de [2].

G.1 Rádio do Tmote Sky

O rádio de comunicação sem fio do Tmote Sky é controlado pelo chip CC2420 [87] da Chipcon. Este chip é compatível com o padrão IEEE 802.15.4 e atua em duas camadas da pilha de protocolos: provê todo o controle da camada física, e algumas funções da camada MAC.

O CC2420 é controlado diretamente pelo microcontrolador MSP430 através da porta SPI (*Serial Peripheral Bus*), uma porta padrão da indústria que oferece ligações (*links*) seriais síncronas de troca de dados em modo de comunicação bidirecional simultânea (*full-duplex*), e uma série de linhas de entradas e saídas digitais e interrupções.

Uma funcionalidade importante do CC2420 é que o microcontrolador pode regular a sua tensão elétrica de operação, permitindo que ele entre em modos de baixo consumo de acordo com a situação.

O CC2420 também oferece dois indicadores de qualidade de sinal: o indicador de força de sinal de recepção (RSSI - *received signal strength indicator*); e o indicador de qualidade de ligação (LQI - *link quality indicator*).

Tanto o RSSI quanto o LQI podem ser usados para tomada de decisão de roteamento e para construção de sistemas de localização espacial de Tmotes [88]. A principal diferença entre os dois é que o RSSI é baseado na força do sinal de rádio durante a recepção de um pacote. Já o LQI pode ser definido como a taxa de erros na entrega de pacotes detectada pelo CC2420.

Alguns trabalhos alegam que o RSSI é impreciso e volátil [46] e que o LQI é superior para determinar a qualidade do sinal e para localização de Tmotes.

Mas no trabalho apresentado em [89], os autores alegam que os problemas do RSSI eram consequências das limitações tecnológicas de chips de rádio antigos como o CC1000 e TR1000. Neste trabalho há comparações que indicam que em chips mais modernos, como o CC2420, o RSSI também pode ser um bom indicador de qualidade de sinal em algumas situações.

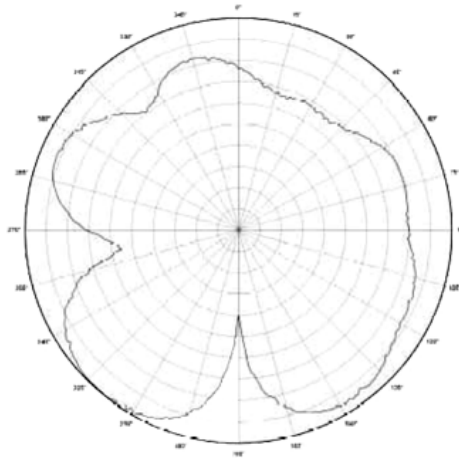


Figura G.2: Padrão de propagação de rádio-frequência da antena do Tmote Sky, adaptado de [2]

G.2 A antena do Tmote Sky

A antena interna do Tmote Sky é do tipo F-invertido, que é um monopolo em que a seção superior é dobrada para baixo para ficar paralela com o plano horizontal. A antena é inserida no circuito impresso da placa do Tmote com a técnica Microstrip.

Embora não seja perfeitamente omnidirecional, o sinal da antena pode alcançar

até 50 metros de distância em ambientes fechados e 125 metros em ambientes abertos.

A Figura G.2, retirada de [2], demonstra o padrão típico de propagação de rádio frequência da antena do Tmote Sky.