

MARIO ANTÔNIO ZANCANARO

**AVALIAÇÃO DE ESTRATÉGIAS DE
DISSEMINAÇÃO DE DADOS PARA REDES
DE SENSORES SEM FIO**

Dissertação submetida ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de Mestre em Informática.

Curitiba-PR
Fevereiro de 2011

MARIO ANTÔNIO ZANCANARO

**AVALIAÇÃO DE ESTRATÉGIAS DE
DISSEMINAÇÃO DE DADOS PARA REDES
DE SENSORES SEM FIO**

Dissertação submetida ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de Mestre em Informática.

Área de concentração: *Ciência da Computação*

Orientador: Prof. Dr. Marcelo Eduardo Pellenz

Curitiba-PR
Fevereiro de 2011

Zancanaro, Mario Antônio.

Avaliação de Estratégias de Disseminação de Dados para Redes de Sensores Sem Fio. Curitiba, 2011. 55p.

Dissertação - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Disseminação de Dados 2. Redes de Sensores Sem Fio 3. Códigos Fontanais 4. Data Dissemination 5. Wireless Sensor Networks 6. Fountain Codes. I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática II-t.

*Esta folha deve ser substituída pela ata de defesa devidamente assinada,
que será fornecida pela secretaria do programa após a defesa.*

A Deus, minha esposa e minha família.

Agradecimentos

Ao apoio, paciência, suporte e incentivo, agradeço a minha esposa Caroline Campesatto, a qual foi diretamente responsável pela conclusão deste trabalho. Compartilho contigo esta etapa alcançada.

Dedico em especial aos meus pais, Oscar e Arlete, esta conquista. Foi fundamental para a conclusão deste desafio o incentivo e educação que vocês me proporcionaram. Sou imensamente grato a vocês.

Muito obrigado às minhas irmãs, Luciane e Karina, por acreditarem e confiarem em mim. A participação de vocês foi fundamental.

Ao meu orientador, Marcelo Eduardo Pellenz, obrigado por toda a ajuda, convívio e paciência. A tua contribuição foi essencial na elaboração deste trabalho. Aprendi muito com teus conselhos e ensinamentos.

Aos professores Edson Emílio Scalabrin, Alvaro Machado e Luiz Pavão pelo incentivo inicial, posterior ajuda e apoio.

Dedico também este trabalho para a professora Angela Menegolla. É uma singela forma de agradecimento por você sempre acreditar no meu potencial e fazer parte da minha formação. O papel de professor é muito bem representado por você.

Meu agradecimento a todos os meus amigos e familiares que vibraram com cada etapa vencida e por todas as vezes que me apoiaram.

Por fim, agradeço a todas as pessoas que direta ou indiretamente participaram nesta conquista.

Resumo

Atualmente as redes de sensores sem fio (RSSF) possuem aplicabilidade em diversas áreas, incluindo automação residencial, sistemas de telemetria industrial, monitoramento ambiental, aplicações militares, entre outras. Para que as RSSF mantenham sua conectividade, operacionalidade e trabalhem de forma correta, faz-se necessário em muitos casos que os dispositivos da rede mantenham algumas informações em comum. Podemos citar como exemplo as informações sobre tabelas de roteamento, atualizações do sistema operacional, atualização de aplicativos ou até mesmo informações específicas de configuração do próprio equipamento. A operação de distribuir a mesma informação para todos os nós da RSSF é denominado *disseminação de dados*. Neste trabalho realizamos um estudo comparativo de diferentes estratégias de disseminação de dados, incluindo os métodos baseados em códigos fontanais. Os métodos baseados em códigos fontanais não exigem a necessidade de um canal de retorno para confirmação das mensagens transmitidas entre os nós durante o processo de disseminação, o que pode resultar diretamente na economia de energia dos nós, além de diminuir o número de mensagens trocadas na rede. O objetivo deste estudo é avaliar a eficiência de cada método sob diferentes condições de conectividade do cenário de rede. Também propomos e avaliamos a utilização do conceito de árvore de disseminação conjuntamente com as técnicas convencionais. Um novo método usando códigos fontanais, denominado *replicação de relay* (RR) é proposto e investigado.

Palavras-chave: Disseminação de Dados, Redes de Sensores Sem Fio, Códigos Fontanais.

Abstract

Nowadays, wireless sensor networks (WSN) can be applied in several areas, including domestic automation, industrial telemetry systems, environmental monitoring, military applications and others. To ensure that the WSN maintain the connectivity, operability and work properly, its necessary that in many cases, several network devices keep an amount of common information. The examples of this information can be router tables, operational system updates, application updates or even specific device configuration. The operation responsible to share the same information to each and every WSN nodes is called data dissemination. This work presents a comparative study between different data dissemination strategies, including methods based on fountain codes. These methods does not have the need of a return channel for confirmation of the sent messages between the nodes during the dissemination process, which can impact directly on nodes energy consumption and reducing the exchanged messages inside the network. The objective of this study is to evaluate the efficiency of each method under different connectivity conditions on the network's scenario. Also is proposed and evaluated the concept of the dissemination tree added with conventional techniques. A new method using fountain codes, named Relay Replication (RR) is proposed and investigated.

Keywords: Data Dissemination, Wireless Sensor Networks, Fountain Codes.

Sumário

Resumo	xi
Abstract	xiii
Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Símbolos	xx
Lista de Abreviações	xxi
1 Introdução	1
1.1 Motivação	1
1.2 Proposta	2
1.3 Estrutura do Documento	3
2 Redes de Sensores Sem Fio (RSSF)	5
2.1 Introdução	5
2.2 Características Gerais das RSSF	6
2.3 Arquitetura dos Sensores	8
2.4 Formação da Rede	10
2.5 Aplicações Típicas e Tecnologias para RSSF	12
2.6 Conclusão	14
3 Códigos Fontanais	15
3.1 Introdução	15
3.1.1 Canais com Apagamento	15
3.1.2 Aplicações	16
3.1.3 Alternativas de Códigos Fontanais	17
3.1.4 Método RLF (Random Linear Fountain)	17
3.1.5 Método LT (Luby Transform Codes)	22
3.2 Conclusão	25
4 Trabalhos Relacionados	27
4.1 FBCast	27
4.2 MPR-BASED CODING PROTOCOL	27
4.3 MDeluge	28

4.4	SYNAPSE	28
5	Algoritmo Proposto e Resultados Obtidos	29
5.1	Disseminação de Dados em RSSF	29
5.1.1	Método de Inundação	30
5.1.2	Método Probabilístico	30
5.1.3	Método Fontanal Probabilístico	31
5.2	Cenário de Avaliação	31
5.2.1	TinyOS e TOSSIM	32
5.3	Análise das Estratégias de Disseminação	34
5.4	Resultados da Avaliação	38
5.5	Análise Complementar dos Métodos FLD e RR	43
6	Conclusão	47
A	Qualcomm e Digital Fountain Inc.	53
A.1	Projetos	53
B	Wireless Sensor Networks Research Group	55
B.1	Projetos	55

Lista de Figuras

2.1	Arquitetura Geral do Sensor.	9
2.2	Pilha de Protocolos em Redes de Sensores.	9
2.3	Funcionamento Básico de um Sensor.	10
2.4	Ciclo de Vida das Redes de Sensores.	10
2.5	Formação de uma rede de sensores [Ruiz et al., 2003].	12
2.6	Características dos Padrões IEEE 802.11 e 802.15 - WLANs e WPANs.	13
3.1	Modelo de um Canal com Apagamento	16
3.2	Fluxo dos Pacotes de Dados	18
3.3	Divisão do Arquivo Original em K Pacotes e Geração da Matriz \mathbf{G}	18
3.4	Processo de Geração dos Pacotes Codificados	19
3.5	Fluxograma do Processo de Decodificação	19
3.6	Processo de Decodificação	20
3.7	Desempenho do Método RLF [MacKay, 2005]	21
3.8	Grafo Resultante do Processo de Codificação do Método LT	22
3.9	Comportamento das Distribuições ρ e τ com $K = 10000$, $c = 0,2$ e $\delta = 0,05$ [MacKay, 2005]	24
3.10	Processo de Decodificação do Método LT	26
5.1	Desempenho dos Blocos de Códigos Convolutacional em Canal Binary Symmetric Channel (BSC).	35
5.2	Resultado da Aplicação do Método de Kruskal.	36
5.3	Métodos FLD e FLD+MDST	39
5.4	Métodos FLD e FLD+MDST	39
5.5	Desempenho para diferentes probabilidades de retransmissão	40
5.6	Métodos FBCast e FBCast+MDST com $p = 0.9$	41
5.7	Métodos FBCast e FBCast+MDST com $p = 0.9$	41
5.8	Métodos FLD, FLD+MDST, FBCast e FBCast+MDST com $p = 0.9$	42
5.9	Métodos FLD, FLD+MDST, FBCast e FBCast+MDST com $p = 0.9$	43
5.10	Métodos RR e RR+MDST	44
5.11	Métodos RR e RR+MDST	44
5.12	Métodos FLD, FBCast+MDST com $p = 0.9$, RR e RR+MDST	45
5.13	Topologia de Rede Sequencial	45
5.14	Comparação teórica entre os métodos de disseminação FLD e RR	46

Lista de Tabelas

2.1	Caracterização das RSSF segundo a Configuração	6
2.2	Caracterização das RSSF segundo o Sensoriamento	6
2.3	Características das RSSF segundo a Comunicação	7
2.4	Caracterização das redes de sensores sem fio segundo a comunicação	7
2.5	Caracterização das redes de sensores sem fio segundo o processamento	8
2.6	Características dos Padrões WLANs, WPANs e LR-WPANs.	13
3.1	Probabilidade de Decodificação do Método RLF [Hyytia et al., 2007a]	21
3.2	Exemplo do Processo de Codificação do Método LT	22

Lista de Símbolos

τ	Função Complementar da Sóliton Robusta
$\rho(d)$	Distribuição Sóliton Ideal
$\mu(d)$	Distribuição Robust Soliton
P_a	Probabilidade de Falha
δ	Limite Superior da Probabilidade de Falha na Decodificação LT
E	Excesso de Pacotes
k	Tamanho do Pacote de Dados
G	Matriz Resultante do Processo de Codificação
n	Símbolos Codificados
$t(n)$	Pacotes Codificados
c	Parâmetro da Distribuição Sóliton Robusta
x	Bits de um pacote

Lista de Abreviações

TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XOR	Exclusive OR
LT	Luby Transform
RLF	Randon Linear Fontain
ARQ	Automatic Repeat Request
WSN	Wireless Sensor Networks
Ad-Hoc	Redes Sem Nó Central
RLF	Randon Linear Fountain
LT-Codes	Luby Tranform Codes
RSSF	Redes de Sensores Sem Fio
RR	Relay Replication
RRFB	Relay Replication Fountain Broadcast
I/O	In/Out
WPAN	Wireless Personal Area Networks
PBCast	Probabilistic Broadcast
FBCast	Fountain Broadcast
MDST	Minimum Dissemination Spanning Tree

Capítulo 1

Introdução

Em nosso cotidiano são comuns equipamentos que possuem interface para comunicação sem fio. Equipamentos como celulares, computadores, impressoras, televisores, dentre outros, estão cada dia mais presentes nas residências, empresas e escolas. O crescimento das redes sem fio, pela utilização e popularização de tais dispositivos, origina uma demanda de novos serviços e aplicações, que por sua vez, representam uma mudança no cenário das comunicações em redes sem fio. Em complemento aos ambientes tradicionais de aplicação, é possível observar também o crescimento das redes sem fio em ambientes rurais e industriais.

As redes sem fio sem infraestrutura podem ser classificadas basicamente nas seguintes categorias: Redes de Sensores (Wireless Sensor Networks - WSNs) e Redes Ad-Hoc (Ad-Hoc Networks). Mesmo que desenvolvidas para diferentes aplicações, algumas características permanecem as mesmas nestas redes. Na arquitetura não existe um nó central ou ponto de acesso (AP), são redes autônomas, tolerantes a falhas e os pacotes são transportados em múltiplos saltos. Especificamente, as redes de sensores sem fio (RSSF) possuem aplicabilidades e características interessantes que as diferem de outras redes. Uma das principais características é que as RSSF são criadas com o objetivo de estabelecer conectividade entre os nós que a compõem, sem que haja necessidade de uma infra-estrutura. Além disso, outras características relevantes são a restrição no uso de energia, menor volume no tráfego de informações e conectividade espontânea dos nós. Entre as aplicações, destaca-se a formação de RSSF em ambientes inóspitos para o monitoramento ambiental climático, monitoramento militar, automação industrial e em casos mais cotidianos, a automação residencial [Akyildiz et al., 2002].

Nas RSSF, todo nó que está ativo é responsável também pela operacionalidade da rede como um todo. Partindo desta premissa várias são as técnicas para suprir a necessidade da rede em manter-se operante. Desde a criação de algoritmos que controlam o tráfego, até situações de controle de potência do rádio transmissor, passando pelo controle de energia do nó e o tráfego de dados entre os mesmos.

1.1 Motivação

As redes de sensores sem fio não transmitem grandes volumes de dados como as redes cabeadas. Na verdade, as RSSF não operam com o intuito de prover comunicação entre usuários ou para usuários. O objetivo das RSSF é voltado para funcionalidades focadas em serviços. Ainda não é foco das RSSF viabilizar recebimento de filmes, mensagens de correio eletrônico ou troca de arquivos de cópias de segurança. Em geral, estas redes operam para

prover coleta ou troca de informações em menor escala de tráfego, incluindo a transmissão de informações entre equipamentos móveis de pequeno porte.

Mesmo possuindo um tráfego menor de informações, as RSSF, assim como as redes cabeadas tradicionais, precisam garantir que as informações trafeguem e cheguem ao seu destino por completo. Para garantir a operacionalidade da rede cada nó precisa identificar seus nós vizinhos, realizar controle de acesso ao meio, realizar o encaminhamento de mensagens, dentre outros requisitos de gerência. Dependendo do cenário de aplicação da RSSF, se faz necessária a troca ou atualização de informações em todos os nós que compõe a rede. Tal operação, denominada de *disseminação de dados*, tem por objetivo manter a operacionalidade e conectividade da rede. A disseminação de informações permite que os nós recebam informações relevantes, como por exemplo atualizações do sistema operacional ou de aplicativos, tabelas de roteamento e comandos de reconfiguração dos dispositivos.

As redes de comunicação tradicionais utilizam os protocolos de transporte TCP (Transmission Control Protocol) ou UDP (User Datagram Protocol) para a troca de informações. O protocolo TCP foi desenvolvido para proporcionar a entrega confiável de informação de uma origem a um destino, utilizando um mecanismo de confirmação dos pacotes recebidos. Em contraste, o protocolo UDP não possui mecanismo de controle de entrega e por consequência não garante a entrega confiável dos dados ao destinatário [Tanenbaum, 2003]. Nas transmissões que envolvem um número elevado de nós da rede, como em aplicações de *broadcasting* e *multicasting*, o uso do protocolo TCP não é recomendado pois ele gera um número excessivo de mensagens de confirmação e reenvio de pacotes devido a característica do mecanismo de garantia de entrega. Por outro lado, o uso do protocolo UDP neste mesmo contexto não garante a entrega fidedigna dos dados ao destinatário.

Os protocolos de transporte TCP e UDP não são adequados para uso em RSSF devido as restrições que as RSSF usualmente apresentam: limitação de energia, limitação de memória, ausência de infra-estrutura, baixa capacidade de processamento, dentre outros. Desta forma, existe a necessidade de um envio confiável porém viabilizado para as RSSF. A forma e o método com que as informações são transmitidas na RSSF afeta diretamente o tempo de permanência ativo do nó, em função da energia restante da bateria. Neste sentido, para as aplicações em RSSF que exigem a entrega confiável dos dados surge a necessidade de estratégias alternativas. Este é o caso dos procedimentos que exigem a *disseminação de dados* para todos os nós da rede. Como alternativa aos protocolos TCP/UDP para viabilizar a disseminação de dados nas RSSF, as estratégias baseadas em códigos fontanais unificam algumas propriedades dos protocolos TCP/UDP. Os códigos fontanais permitem que sejam enviadas informações a um grande número de nós através de multicast ou broadcast, de maneira confiável (princípio do TCP) e sem a necessidade do tradicional mecanismo de confirmação de recebimento para cada pacote (princípio do UDP). Neste sentido a motivação principal deste trabalho é comparar o desempenho das estratégias clássicas com os métodos baseados em códigos fontanais para viabilizar a disseminação de informações em uma RSSF.

1.2 Proposta

O objetivo principal desta dissertação de mestrado é comparar estratégias clássicas de disseminação de dados com métodos baseados em códigos fontanais, identificando o melhor método para diferentes cenários da RSSF. Também propomos e investigamos um método alternativo aos métodos tradicionais, denominado *replicação de relays* (RR), que é baseado

na combinação de conceitos de escolha de nós estratégicos na rede, ditos nós *relays*, com o conceito de envio de dados sem canal de retorno viabilizado pelo método de códigos fontanais.

Especificamente temos como objetivo realizar um estudo comparativo de desempenho das seguintes técnicas de disseminação de dados: determinística (Flooding), retransmissão probabilística baseada em códigos fontanais (FBCast) e método proposto RR. A comparação é realizada através de simulação, mantendo as mesmas condições de cenário para cada estratégia. Utilizamos em nosso estudo um cenário de topologia em grade, onde são alterados os parâmetros referentes ao espaçamento entre os nós (densidade), quantidade de nós e modelo de descarte de pacotes no canal de rádio.

1.3 Estrutura do Documento

O restante deste documento está estruturado da seguinte forma: O Capítulo 2 apresenta os conceitos fundamentais sobre RSSF, descrevendo as características gerais destas redes, arquitetura dos nós sensores e as aplicações típicas. No Capítulo 3 estão descritos os fundamentos teóricos dos códigos fontanais, incluindo os princípios básicos de codificação e decodificação fontanal. As estratégias clássicas de disseminação de dados são apresentadas e comparadas no Capítulo 4, juntamente com o novo método de *replicação de relays* proposto. Os resultados obtidos também são apresentados no Capítulo 4. Finalmente o Capítulo 5 apresenta as conclusões deste estudo.

Capítulo 2

Redes de Sensores Sem Fio (RSSF)

2.1 Introdução

As redes de sensores sem fio são basicamente um grupo de dispositivos com comunicação de rádio que interconectam-se entre si com o objetivo de executar alguma tarefa colaborativa. Estes dispositivos, denominados de nós, são alimentados por uma bateria e possuem em geral um comportamento autônomo. Assim como as demais redes de comunicação sem fio, as RSSF também apresentaram um amplo crescimento em termos dos cenários de aplicação. Algumas características principais destas redes fomentaram este crescimento, tais como o custo do equipamento que é relativamente menor em relação a outras redes clássicas cabeadas. As RSSF podem ser implantadas em lugares inóspitos e de difícil acesso e podem atender aplicações em diversas áreas. Além disso este modelo de rede, mesmo não possuindo como intuito a substituição das redes clássicas, possui vantagens significativas em relação as mesmas em função de características de mobilidade, independência de infra-estrutura e adaptação em ambientes diferenciados.

No princípio, as aplicações ligadas as RSSF eram fortemente voltadas para a área de monitoramento. É comum encontrar aplicações neste segmento, onde são realizadas basicamente tarefas de coleta e processamento de dados. Porém atualmente as aplicações das RSSF não estão mais restritas ao monitoramento. O surgimento de dispositivos eletrônicos de baixo custo com múltiplas funcionalidades permitiu um avanço em relação ao serviços iniciais. Robôs que interagem entre si, escritórios com circuitos elétricos automatizados, além do acionamento de dispositivos simples de entrada e saída (I/O) em equipamentos eletrônicos são exemplos que diferem do monitoramento.

Manter uma RSSF operante pelo maior tempo possível é assunto de amplo estudo e pesquisa. O desperdício de processamento e o uso do rádio transmissor sem necessidade são dois aspectos que requerem um gerenciamento, uma vez que podem diminuir o tempo de operação de cada nó na rede. Quando um nó deixa de operar, automaticamente existe uma mudança de topologia da rede, que por sua vez afeta aspectos como conectividade, rotas de encaminhamento de mensagens, entre outros.

2.2 Características Gerais das RSSF

Tipicamente uma rede de sensores é formada por um agrupamento de dezenas ou centenas de nós sensores. As RSSF podem ser caracterizadas e classificadas em função de diferentes aspectos. Podemos citar como exemplo o modo de operação da rede, tipo de aplicação, distribuição e disposição de nós no cenário da rede, tipo de informação tratada pelos nós, além dos planos de economia de energia. As Tabelas, 2.1, 2.2 e 2.3 apresentam detalhes de classificação das RSSF segundo as características de configuração, sensoriamento e comunicação [Ruiz et al., 2003]. As Tabelas 2.4 e 2.5 detalham características de classificação em termos do tipo de comunicação e processamento dos nós [Ruiz et al., 2003].

Tabela 2.1: Caracterização das RSSF segundo a Configuração

Configuração		
Composição	homogênea	Rede composta de nós que apresentam a mesma capacidade de hardware. Eventualmente os nós podem executar software diferente.
	heterogênea	Rede composta por nós com diferentes capacidades de hardware.
Organização	hierárquica	RSSF em que os nós estão organizados em grupos (clusters). Cada grupo terá um líder (cluster-head) que poderá ser eleito pelos nós comuns. Os grupos podem Organizar hierarquias entre si.
	plana	Rede em que os nós não estão organizados em grupos
Mobilidade	estacionária	Todos os nós sensores permanecem no local onde foram depositados durante todo o tempo de vida da rede.
	móvel	Rede em que os nós sensores podem ser deslocados do local onde inicialmente foram depositados.
Densidade	balanceada	Rede que apresenta uma concentração e distribuição de nós por unidade de área considerada ideal segundo a função objetivo da rede.
	densa	Rede que apresenta uma uma alta concentração de nós por unidade de área.
	esparsa	Rede que apresenta uma baixa concentração de nós por unidade de área.
Distribuição	irregular	Rede que apresenta uma distribuição não uniforme dos nós na área monitorada.
	regular	Rede que apresenta uma distribuição não uniforme de nós sobre área monitorada.

Tabela 2.2: Caracterização das RSSF segundo o Sensoriamento

Sensoriamento		
Coleta	periódica	Os nós sensores coletam dados sobre o(s) fenômeno(s) em intervalos regulares. Um exemplo são as aplicações que monitoram o canto dos pássaros. Os sensores Farão a coleta durante o dia e permaneceram desligados durante a noite.
	contínua	Os nós sensores coletam os dados continuamente. Um exemplo são as aplicações de exploração interplanetária que coletam dados continuamente para a formação de Base de dados para pesquisas.
	reativa	Os nós sensores coletam dados quando ocorrem eventos de interesse ou quando solicitado pelo observador. Um exemplo são as aplicações que detectam a presença de objetos na área monitorada.
	tempo real	Rede em que os nós não estão organizados em grupos.

Tabela 2.3: Características das RSSF segundo a Comunicação

Classificação segundo a comunicação		
Disseminação	programada	Os nós disseminam em intervalos regulares.
	contínua	Os nós disseminam os dados em intervalos regulares.
	sob demanda	Os nós disseminam os dados em resposta à consulta do observador e à ocorrência de eventos.
Conexão	simétrica	Todas as conexões existentes entre os nós sensores, com exceção do nó sorvedouro têm o mesmo alcance.
	assimétrica	As conexões entre os nós comuns têm alcance diferente.
Transmissão	simplex	Os nós sensores possuem transceptor que permite apenas transmissão da informação.
	half-duplex	Os nós sensores possuem transceptor que permite transmitir ou receber em um determinado instante.
	full-duplex	Os nós sensores possuem transceptor que permite transmitir ou receber dados ao mesmo tempo.

Tabela 2.4: Caracterização das redes de sensores sem fio segundo a comunicação

Classificação segundo a comunicação		
Alocação de Canal	estática	Neste tipo de rede se existirem "n" nós, a largura de banda é dividida em "n" partes iguais na frequência (FDMA - Frequency Division Multiple Access), no tempo (TDMA - Time Division Multiple Access), no código (CDMA - Code Division Multiple Access), no espaço (SDMA - Space Division Multiple Access) ou ortogonal (OFDM - Orthogonal Frequency Division Multiplexing). A cada nó é atribuída uma parte privada da comunicação, minimizando interferência.
	dinâmica	Neste tipo de rede não existe atribuição fixa de largura de banda. Os nós disputam o canal para comunicação dos dados.
Fluxo de Informação	flooding	Neste tipo de rede, os nós sensores fazem broadcast de suas informações para seus vizinhos que fazem broadcast desses dados para outros até alcançar o ponto de acesso. Esta abordagem promove um alto overhead mas está imune às mudanças dinâmicas de topologia e a alguns ataques de impedimento de serviço (DoS - Denial of Service).
	multicast	Neste tipo de rede os nós formam grupos e usam o multicast para comunicação entre os membros do grupo.
	unicast	Neste tipo de rede, os nós sensores podem se comunicar diretamente com o ponto de acesso usando protocolos de roteamento multi-saltos.
	gossiping	Neste tipo de rede, os nós sensores selecionam os nós para os quais enviam os dados.
	bargaining	Neste tipo de rede, os nós enviam os dados somente se o nó destino manifestar interesse, isto é, existe um processo de negociação.

Uma característica desejável em toda RSSF é fazer com que a mesma opere durante o maior tempo possível. Tal propósito é obviamente viabilizado através da economia de energia, que pode ser obtida através de diversas formas: controle e gerenciamento de potência do sinal do rádio transmissor, economia em número de mensagens que a rede transmite aos seus nós,

Tabela 2.5: Caracterização das redes de sensores sem fio segundo o processamento

Classificação segundo o processamento		
Cooperação	infra-estrutura	Os nós sensores executam procedimentos relacionados à infra-estrutura da rede como por exemplo, algoritmos de controle de acesso ao meio, roteamento, eleição de líderes, descoberta de localização e criptografia.
	localizada	Os nós sensores executam além dos procedimentos de infra-estrutura, algum tipo de processamento local básico como por exemplo, tradução dos dados coletado pelos sensores baseado na calibração.
	correlação	Os nós estão envolvidos em procedimentos de correlação de dados como fusão, supressão seletiva, contagem, compressão, multi-resolução e agregação.

roteamento eficiente das informações, controle do reenvio de dados ocasionados por colisões nas transmissões, dentre outros.

Conforme [Akyildiz et al., 2002], mobilidade e escalabilidade também estão presentes em uma RSSF, que pode operar com dezenas ou centenas de nós, dependendo da aplicação em que está envolvida. Redes de sensores sem fio de alta densidade exigem um bom controle de fluxo de informações e de auto-organização, caso contrário comprometem o desempenho da rede desperdiçando principalmente recursos de transmissão e processamento dos nós. Por outro lado, o fato de existir uma alta densidade de nós em uma RSSF, agrega uma maior autonomia e ao mesmo tempo deixa a rede preparada para eventuais falhas dos nós. Em um cenário onde existam muitos nós próximos, é possível que quando um destes deixa de funcionar, seja por problemas de hardware ou interferências climáticas, outro nó se habilita para fazer a mesma tarefa e manter a rede operante naquele ponto.

2.3 Arquitetura dos Sensores

Cada sensor que compõe uma RSSF possui uma arquitetura que simplificada é composta por um transmissor, um transdutor, uma unidade de processamento (memória e processador) e uma bateria. A Figura 2.1 adaptada de [Akyildiz et al., 2002] demonstra esta arquitetura.

Em síntese, a pilha da arquitetura pode ser descrita da seguinte forma: O transmissor, responsável pela comunicação sem fio entre os nós pode operar em diferentes padrões de transmissão (Laser, Rádio Frequência e Infravermelho). O transdutor é a parte do equipamento responsável pela coleta de informação, podendo realizar diferentes tipos de coleta como acústicas, sísmicas e de imagem. O processador responde pelo processamento e envio das informações, bem como todo o esquema de comunicação e manipulação de dados. Por fim, a bateria é a responsável por manter o equipamento ativo na rede.

A pilha de protocolos das redes de sensores pode ser demonstrada conforme a Figura 2.2 adaptada de [Akyildiz et al., 2002]. Na pilha vertical frontal, a RSSF segue basicamente o padrão clássico. A ressalva está na pilha horizontal lateral que prevê gerenciamento de tarefas, mobilidade e energia. É justamente nesta parte que as RSSF se diferenciam das redes comuns cabeadas. O impacto causado por uma mau gerenciamento de qualquer um destes itens compromete a rede como um todo. O plano de gerenciamento de energia aparece em primeiro

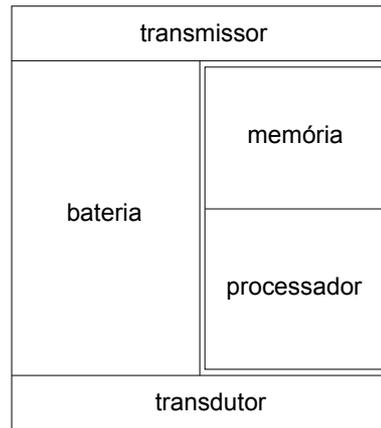


Figura 2.1: Arquitetura Geral do Sensor.

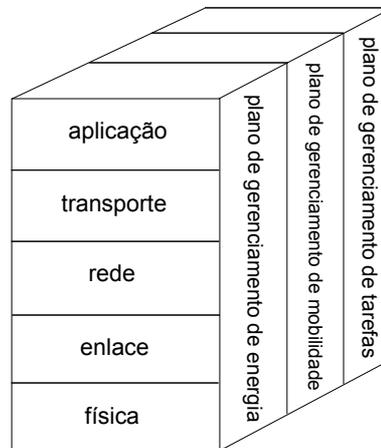


Figura 2.2: Pilha de Protocolos em Redes de Sensores.

plano pois é possível afirmar que é um dos mais importantes, uma vez que determina o tempo de operação da rede.

O gerenciamento de mobilidade e de tarefas está após o plano de energia, mas não têm sua importância minimizada pois, a ausência do gerenciamento de tarefas acarretaria em uma duplicidade de processamento para um determinado nó causando um gasto maior de energia. Ainda, a falta de um plano de mobilidade faz com que um ou mais nós fiquem fora do foco do monitoramento e, além do gasto da energia, o nó não participaria da aplicação a qual a rede foi submetida. Um plano de mobilidade recoloca ou então ajusta este nó para um melhor aproveitamento deste na aplicação.

O funcionamento de um nó sensor é elucidado na Figura 2.3 conforme [Bischoff et al., 2009], e as funcionalidades podem ser resumidas pelas etapas ilustradas. O sensor, que pode ser de diferentes modelos (acústico, pressão, velocidade) coleta as informações e envia para o armazenamento temporário e posterior processamento. No processamento é feita a análise da informação e classificada segundo aplicação em que a rede e o nó estão submetidos. Em função da análise da informação pode ser disparado algum aviso ou alerta. Além disso, cada nó precisa processar e monitorar informações de controle gerencial de si pró-

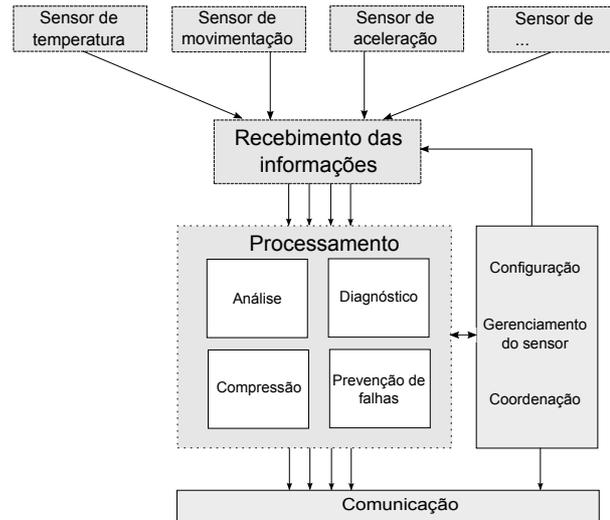


Figura 2.3: Funcionamento Básico de um Sensor.

prio. Esta operação pode ser um simples agendamento de tarefa, processamento de algoritmo ou controle de sua própria energia restante. Um nó sensor deve ainda processar o recebimento de comunicação de nós vizinhos e reenviar pacotes de dados. Por fim, ainda é necessário que o nó controle a sua comunicação com os nós próximos no intuito de desempenhar o papel colaborativo para a rede toda.

2.4 Formação da Rede

Em qualquer aplicação envolvida, a formação e ciclo de vida das redes de sensores segue a classificação em etapas de estabelecimento da rede, manutenção, sensoriamento, processamento e comunicação. Estas fases são simultâneas em suas ocorrências e podem estar ativas em diferentes momentos do tempo de vida das redes de sensores [Ruiz et al., 2003]. A Figura 2.4 demonstra o esquema de fases na qual a rede de sensores tem seu ciclo de vida estabelecido.

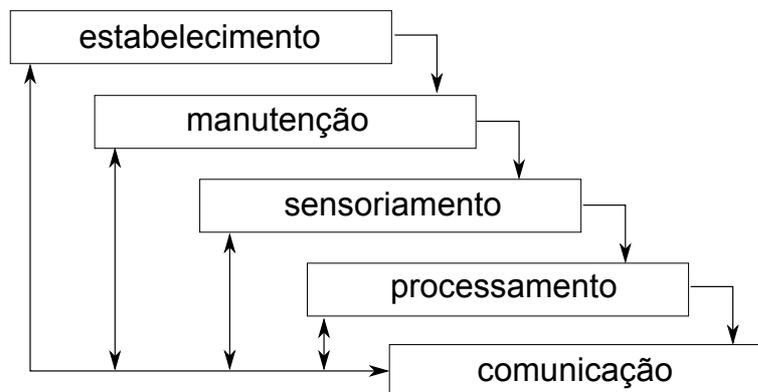


Figura 2.4: Ciclo de Vida das Redes de Sensores.

Conforme [Ruiz et al., 2003], o estabelecimento da rede ocorre após os sensores serem lançados de uma maneira aleatória ou posicionados nos lugares desejados, iniciando o processo de formação da rede. Este passo é o despertar dos sensores, na qual existe uma identificação dos vizinhos. Na fase seguinte ao reconhecimento, os sensores estabelecem sua organização enquanto rede propriamente dita e então acontece a formação de sub-redes ou agrupamento em *cluster*. Por fim, existe a troca de informações entre os nós ou auto-organização. Nesta última etapa, iniciam os algoritmos de controle que estabelecem a autosuficiência da rede. A Figura 2.5 demonstra as etapas de formação de uma RSSF.

A etapa de estabelecimento, assim como as demais, acontece repetidamente pelo fato da rede, dependendo de onde é estabelecida, sofrer com perda de nós por problemas de comunicação, problemas no equipamento, ruído, barreiras naturais, dentre outros. Diante deste fato, os nós precisam novamente reestabelecer a rede repetindo as etapas. É sabido também que, quando encontrado uma alta densidade na formação dos nós da rede - quando os nós estão muito próximos uns dos outros - a rede pode utilizar estes nós na substituição de outros próximos que venham a falhar.

A fase de manutenção, como o nome propriamente relata, é a fase em que existe uma preocupação em manter a rede por mais tempo ativa. Devido ao dinamismo encontrado neste tipo de rede, a manutenção está envolvida em todas as fases ditas anteriormente. Normalmente, a fase de manutenção entra em cena quando um ou mais nós perde energia, fica temporariamente inoperante ou simplesmente para de funcionar.

A fase de sensoriamento é a fase onde a coleta de informações é realizada. É também a etapa de percepção do ambiente que identifica a capacidade dos nós envolvidos de transmitir as informações, identifica a posição geográfica e ruídos e volume de informações, além de ajustar as redundâncias dos nós próximos de modo que a rede suporte a suposta transmissão. No sensoriamento também existe uma colaboração entre os nós, onde um nó poderá utilizar um vizinho próximo para substituí-lo no envio das informações. Neste momento, existe automaticamente uma mudança na topologia da rede e, por tal motivo, é que o processo de arranjo e demais é contínuo.

O processamento envolvido nos nós de uma rede de sensores é dual. Processamento de suporte: diz respeito e realiza toda a execução de manutenção do nó, da rede, de protocolos e demais vinculados ao estado da rede como um todo. É o processamento motor do gerenciamento da rede. Processamento de informações: diz respeito as informações coletadas pelo nó, processadas a partir de uma aplicação específica. É neste processamento que está a parte colaborativa da rede. Também é neste processamento que os protocolos são executados e as execuções dependem de um gatilho para disparar. Por exemplo, um aumento excessivo de uma temperatura pode desencadear, por sua vez, a execução de um aplicativo.

A comunicação é a parte de estabelecimento da rede de sensores que mantém contato com o usuário final. E nesta etapa que são enviadas as informações pertinentes e esperadas das coletas realizadas. De forma geral, os nós executam suas tarefas comunicam-se entre si e enviam seus resultados ao nó denominado *sink*. Um nó *sink*, é um nó responsável por receber todos os dados coletados e processados e, por sua vez, entregar as informações ao usuário. Na comunicação, ambientes com ruído e interferência fazem com que o tempo de energia do nó diminua pelo fato de existir um número maior de transmissões até que a informação trafegue na rede. Em função do limite de alcance das transmissões, a comunicação é geralmente feita em múltiplos saltos (multi-hop), através de diferentes nós. Os protocolos utilizados para comuni-

cação entre os nós variam em função do padrão ou tecnologia utilizada. Podemos citar como exemplo os padrões HomeRF2, Bluetooth, ZigBee e MicaMotes.

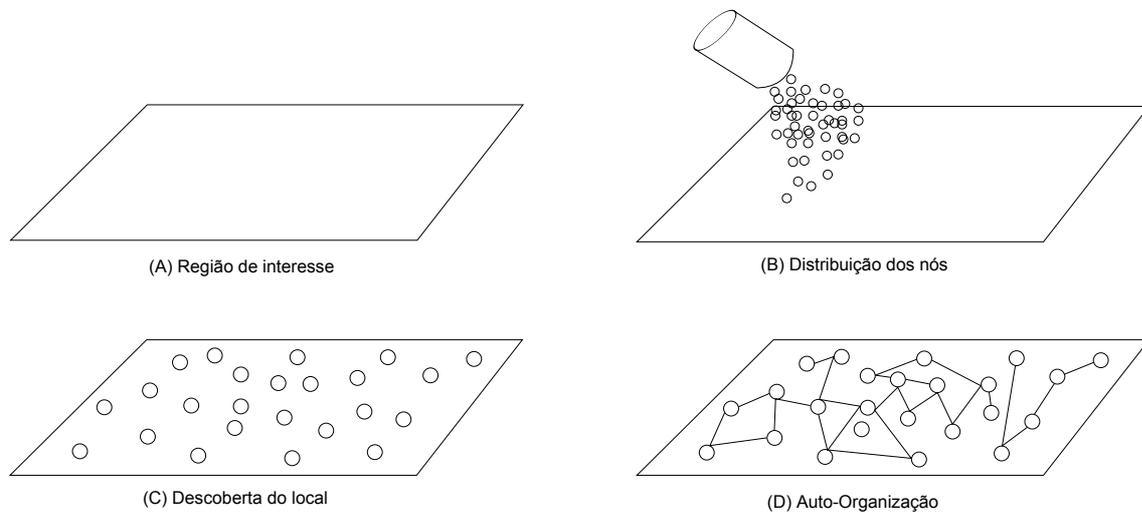


Figura 2.5: Formação de uma rede de sensores [Ruiz et al., 2003].

2.5 Aplicações Típicas e Tecnologias para RSSF

As redes de comunicação pessoal sem fio (Wireless Personal Area Networks - WPANs) [Toh, 2001] são uma extensão do padrão de redes locais sem fio (Wireless Local Area Networks - WLANs) [Santamaria, 2001] e foram desenvolvidas para uso em escala de cobertura menor, como ambientes internos, e para atender usuários finais e suas aplicações locais. Este padrão também é caracterizado por possuir uma pequena ou nenhuma infra-estrutura de rede. A IEEE categoriza em três diferentes classes as WPANs, considerando características de transferência, energia, e qualidade de serviço (Quality of Service - QoS). Desta forma, o padrão WPAN IEEE 802.15.3 é indicado para aplicações que necessitam maior taxa de transferência e um alto QoS. Na categoria mediana de taxa de transferência e QoS, está o padrão WPAN IEEE 802.15.1 Bluetooth, aplicado a equipamentos móveis como telefones celulares ou equipamentos eletrônicos em geral. Por fim, o padrão LR-WPAN (Low Rate Wireless Personal Area Networks) IEEE 802.15.4, apesar da mudança de nomenclatura é também uma extensão das WLANs, sendo o que possui menor taxa de transferência e não tem como foco QoS em suas aplicações [Gutierrez et al., 2003]. A Figura 2.6 e a Tabela 2.6 adaptada de [Gutierrez et al., 2003], são complementares e demonstram a categorização acima citada em função das características de cada padrão.

Em [Bischoff et al., 2009], a categorização dos dispositivos utilizados nas RSSF é realizada de acordo com critérios de: *equipamentos para propósitos gerais, módulos de sensores embutidos e sistema de circuito integrado ou chip*. As plataformas envolvidas no primeiro caso são de baixa potência e estão em computadores pessoais rodando Windows ou Linux com dispositivos de comunicação sem fio nos padrões IEEE 802.11 e IEEE 802.15.1. Oferecem maior capacidade de processamento e, portanto, permitem execução de algumas linguagens de mais alto nível. Por não possuir muitas restrições de programação e linguagem, a consequência é um processamento maior e por este motivo um alto consumo de energia. O dispositivo mais comum

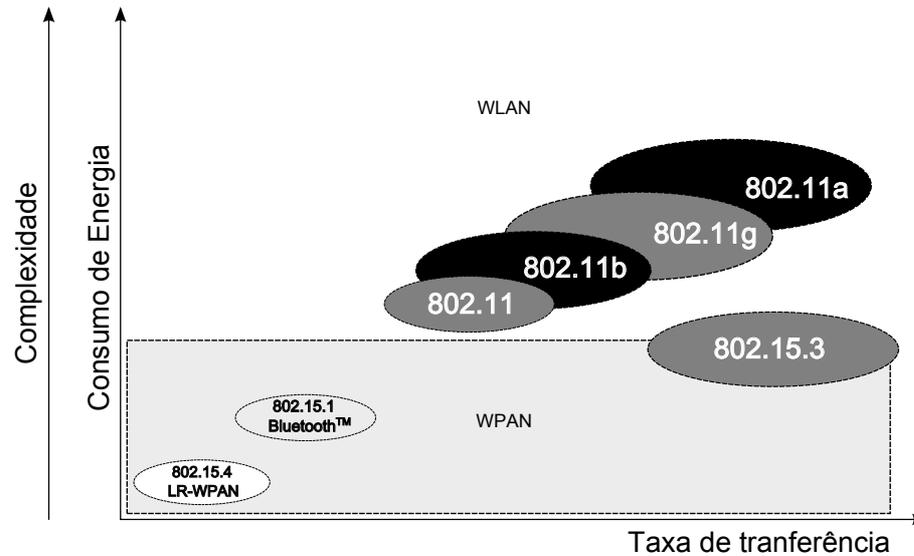


Figura 2.6: Características dos Padrões IEEE 802.11 e 802.15 - WLANs e WPANs.

Tabela 2.6: Características dos Padrões WLANs, WPANs e LR-WPANs.

	802.11b (WLAN)	802.15.1 (WPAN)	802.15.4 (LR-WPAN)
Intervalo	100m	10 - 100m	10m
Taxa de transferência	2-11 Mbits/s	1 Mbits/s	< 0.25 Mbits/s
Consumo de energia	Médio	Baixo	Muito baixo
Tamanho	Grande	Pequena	Muito pequeno
Custo/Complexidade	Alto	Médio	Muito baixo

neste ambiente é o Bluetooth. No segundo caso, os módulos de sensores embutidos são dispositivos de custo financeiro mais baixo, podem ser embutidos em equipamentos diversos e possuem como característica o fato de serem limitados em relação a memória e processamento. Utiliza como programação a linguagem C, para maior controle de memória e processamento e os dispositivos mais comuns neste caso são os da família Mica (Mica, MicaZ, Mica2) e Imote2. A última classificação contempla dispositivos muito pequenos, são os de menor capacidade de processamento e o equipamento todo é um único chip. Um exemplo deste dispositivo é o Smart Dust. Não é objetivo deste trabalho esclarecer os detalhes técnicos e características dos principais dispositivos utilizados pelas redes de sensores sem fio. Para detalhes sobre Bluetooth consulte [Gangali, 2002], assim como para ZigBee vide [Gislason, 2008], Smart Dust e família Mica podem ser encontrados respectivamente em [Ilyas and Mahgoub, 2006] e [Jin et al., 2007].

As redes de sensores sem fio podem ser aplicadas nos mais diversos segmentos ou áreas [Akyildiz and Vuran, 2010], sendo mais visíveis e difundidas as aplicações para WPAN. Porém, existem aplicações mais restritas em que as RSSF estão presentes. Na área de automação, as RSSF podem ser utilizadas na integração entre equipamentos eletrônicos, domésticos ou comerciais para fins diversos, desde informar sobre avisos de abertura ou fechamento de portões até temperatura de fornos ou caldeiras. No segmento militar [Akyildiz and Vuran, 2010], as RSSF auxiliam na detecção de gases nocivos ou avisam sobre uma possível intrusão em território restrito. Na área de defesa civil as RSSF ajudam na detecção de eventuais movimentos sísmicos, atividades de vulcões ou até mesmo na percepção de pequenos tremores em áreas

desejadas. No setor de segurança, uma RSSF pode ser aplicada para monitorar edifícios, casas, indústrias e estabelecimentos comerciais [Gutierrez et al., 2003], com o objetivo de fornecer informações sobre a movimentação nestes ambientes. O trânsito das cidades também é alvo das RSSF [Dahlman et al., 2009], podendo monitorar como está o tráfego das rodovias, informar em tempo real o deslocamento dos veículos e evitar congestionamentos indicando o melhor caminho. Por fim, na área médica [Otto et al., 2006], uma RSSF pode auxiliar médicos e enfermeiros no monitoramento do paciente a distância. Sensores podem ser acoplados no corpo de uma pessoa para medir os índices e taxas de glicose, batimentos cardíacos ou até mesmo monitorar a situação de alguns órgãos vitais. Esta aplicação se estende, da mesma forma, para a área veterinária.

2.6 Conclusão

Este capítulo apresentou as características básicas, detalhes de equipamentos e aplicações das RSSF. A diversidade de aplicações em diferentes áreas e segmentos em que as RSSF atuam, demonstra a evolução das redes de sensores sem fio. Por este motivo, é preciso ter em mente que novas alternativas de gerenciamento, metodologias e aplicações serão necessárias para otimizar e viabilizar ainda mais o uso das RSSF no cotidiano das pessoas. Também foi apresentado neste capítulo a importância de metodologias de configuração, operação e gerenciamento que viabilize a economia de energia dos nós que compõe a RSSF, bem como alguns parâmetros dos principais padrões atualmente utilizados.

Capítulo 3

Códigos Fontanais

3.1 Introdução

Códigos fontanais são uma metodologia para transmissão de dados com confiabilidade, independente do meio de propagação dos dados (com ou sem fio) e do modo de transmissão (*unicast*, *broadcast* ou *multicast*). Através de uma transmissão com códigos fontanais, um usuário recebe pacotes de dados codificados e, uma vez que estes pacotes sejam suficientes para proceder com a decodificação, o usuário deve ser capaz de reconstruir o pacote original [MacKay, 2005].

Um analogia comum feita para o entendimento dos códigos fontanais é o de uma fonte natural, onde um usuário recebe gotas (pacotes codificados) até que sua caneca (arquivo) esteja completa. Enquanto a caneca vai enchendo, é irrelevante ao usuário a quantidade de gotas perdidas (pacotes descartados na rede), o que realmente importa é que em algum momento a mesma estará completa (arquivo recuperado).

Em teoria, para fazer uma transmissão utilizando o método de códigos fontanais, é preciso particionar o arquivo de dados (mensagem) em K pacotes, codificar um subconjunto destes pacotes usando a operação XOR dos bits, e enviar o pacote codificado resultante ao receptor. A quantidade possível de codificações para os K pacotes do transmissor é relativamente grande [Mitzenmacher, 2005], por este motivo é feita a comparação com a fonte natural. Do lado do receptor, não importa quais pacotes de dados foram perdidos, o que importa é que em algum momento, com os pacotes que foram recebidos, será possível montar o arquivo original. Isso justifica a ausência de aviso sobre os pacotes eventualmente perdidos durante a transmissão.

3.1.1 Canais com Apagamento

No processo de comunicação digital o efeito do ruído e das interferências externas provoca a degradação do sinal recebido, causando erros na recepção dos dados ou mesmo a perda dos pacotes transmitidos. Na comunicação sem fio os efeitos de propagação do canal de rádio mais severos para o processo de comunicação. Para minimizar estes problemas os sistemas de comunicação empregam estratégias para detecção e correção de erros, conjuntamente com mecanismos e protocolos de retransmissão de dados. Os códigos fontanais são uma alternativa interessante neste sentido, uma vez que podem ser realizadas inúmeras transmissões codificadas da mensagem para o destino, até que o mesmo consiga receber integralmente a informação

original. Para tanto, não se faz necessário nenhum mecanismo para correção de erros ou a utilização de canal de retorno para retransmissão de pacotes perdidos.

O canal de comunicação sem fio pode ser modelado como um *canal com apagamento*. Neste modelo temos um conjunto de símbolos de entrada (alfabeto de entrada) de tamanho $t = 2^l$, onde l é o número de bits que representa cada símbolo. Existe uma probabilidade $1 - p_a$ de um dado símbolo ser recebido corretamente e uma probabilidade p_a do símbolo não ser recebido (apagamento). A Figura 3.1 ilustra um exemplo do modelo do canal com apagamento para um alfabeto de entrada representado por $\{00, 01, 10, 11\}$. Na saída do canal o símbolo ? representa o não recebimento ou recebimento incorreto do símbolo enviado (apagamento).

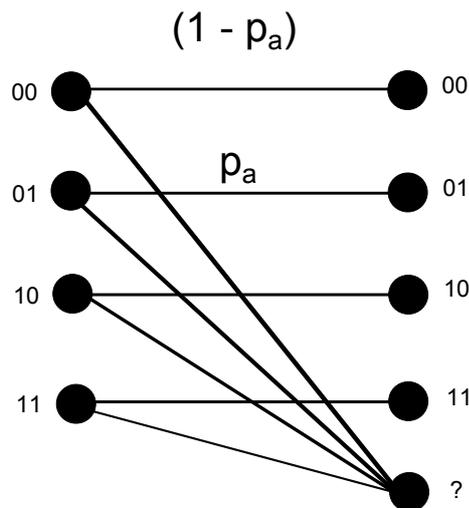


Figura 3.1: Modelo de um Canal com Apagamento

3.1.2 Aplicações

Pela característica de entrega confiável sem necessitar de um canal de retorno, o uso dos códigos fontanais, na maioria dos casos, envolve aplicações que necessitam transmitir dados através de canais sem fio. Normalmente, este tipo de canal de comunicação pode apresentar altas taxas de erro no envio dos pacotes de dados, devido aos efeitos de propagação do canal de rádio. Exemplos típicos do uso de códigos fontanais podem ser encontrados em [Hongpeng Zhu, 2008] e [Casari et al., 2008]. A seguir são apresentadas algumas aplicações em que a utilização de códigos fontanais é indicada, bem como os principais benefícios que o método agrega em cada uma delas.

No recebimento de dados paralelos, códigos fontanais podem prover eficiência no sentido de que cada receptor pode coletar de diferentes fontes e a diferentes taxas, os dados que fazem parte do arquivo desejado. Cada fonte pode prover distintas codificações dos dados originais, sem correr o risco de duplicidade de pacotes. O usuário também pode encerrar sua conexão com uma das fontes a qualquer momento. Aplicações de *multicast* confiável consistem em enviar pacotes de dados a um grupo de usuários. Os códigos fontanais representam uma boa alternativa para *multicast* em relação ao método convencional TCP na transmissão confiável de arquivos. A utilização do mecanismo de retorno de confirmação usado no TCP (mensagens de ACK), fica inviabilizado quando o grupo de usuários é elevado. O uso de códigos fontanais

é indicado porque possui a garantia de entrega das informações sem a necessidade de retorno de confirmação de cada pacote recebido. O método é ainda mais indicado para redes sem fio que possuem um canal de comunicação mais limitado, com maior número de colisões e que consequentemente necessitam mais retransmissões.

Para uma situação em que vários usuários desejam baixar quase que simultaneamente o mesmo arquivo, a utilização de códigos fontanais é proveitosa no sentido de que os pacotes que fazem parte deste arquivo original são duplicados apenas quando as conexões se dividem. Além disso, no caminho comum da fonte até o usuário não é necessário transmissões separadas de cada e para cada usuário. Se de alguma forma não existir simultaneidade ao baixar arquivos, e um usuário por acaso iniciou o processo de *download* de um arquivo a mais tempo que outro, o último pode começar a receber imediatamente pacotes codificados, uma vez que não é necessário receber os pacotes em ordem.

A velocidade para baixar os arquivos, usando códigos fontanais, também pode ser vantajosa se por acaso os usuários compartilharem em algum momento a rota até o servidor de arquivos. O arquivo pode ser enviado a uma taxa mais alta de transmissão até o momento em que a rota se torna individual a cada usuário. Neste caso, a velocidade é reduzida conforme a capacidade do canal de cada usuário.

Nas transmissões multicast usando o protocolo TCP (ponto-multiponto), um servidor qualquer mantém abertas diferentes conexões para diferentes usuários e serve a cada um deles alocando recursos de memória, *buffer* para eventuais retransmissões, dentre outros. Consequentemente, tornando limitado o número de conexões por servidor [Kurose and Ross, 2007]. Utilizando códigos fontanais para este caso, não se faz necessário manter um *buffer* de retransmissões para cada conexão, bem como, não é necessário fazer o envio separado dos pacotes para cada usuário. Outra aplicação dos códigos fontanais é a transmissão de vídeo em tempo real. A idéia proposta para compartilhamento de vídeo em tempo real através de códigos fontanais é segmentar o vídeo. Enquanto uma parte é exibida, a parte subsequente é baixada, e assim sucessivamente, até o final da exibição. Ao usar este método é necessário considerar a taxa de download e taxa de execução, a fim de não executar alguma parcela do vídeo que ainda não foi recebida.

3.1.3 Alternativas de Códigos Fontanais

Os códigos fontanais podem ser implementados através de quatro métodos principais: RLF - Random Linear Fountain, [MacKay, 2005], LT - Luby Transform Codes [Luby, 2003], Tornado Codes [Khisti,] [Luby, 1998], derivados dos códigos LDPC [Leiner, 2005], e Raptor Codes [Shokrollahi, 2006]. Nesta seção apresentamos detalhes sobre o processo de codificação e decodificação dos métodos clássicos, RLF e LT.

3.1.4 Método RLF (Random Linear Fountain)

O método RLF [MacKay, 2005] é o mais simples dos métodos e sua representação é através de uma matriz de dados. A Figura 3.2 ilustra o fluxo dos pacotes de dados entre origem e destino. Na origem os pacotes são codificados pelo código fontanal e por um código detetor/corretor de erros, que possibilita descartar pacotes codificados corrompidos no receptor ou ainda corrigir eventuais erros causados pelo canal de comunicação.

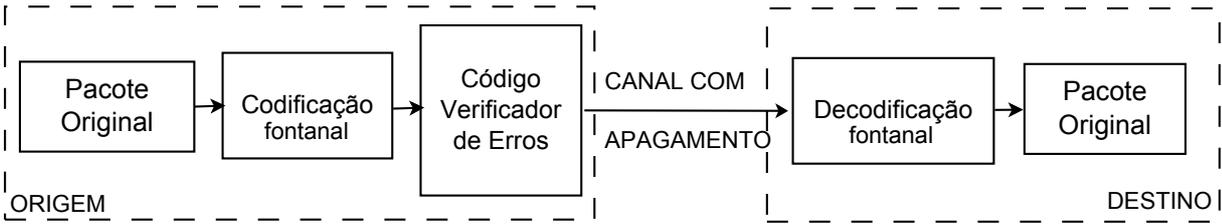


Figura 3.2: Fluxo dos Pacotes de Dados

Processo de Codificação

A seguir é demonstrado didaticamente como é realizada a codificação no método RLF. Considere a transmissão de um arquivo com $K \cdot l$ bits. O processo de codificação se inicia com a divisão do arquivo a ser enviado em K pacotes de l bits cada, denominados por $s_1, s_2, s_3, \dots, s_K$, conforme ilustrado na Figura 3.3a. O codificador gera a cada ciclo de codificação, n , uma palavra aleatória de K bits para a codificação, G_{kn} , e as concatena (coluna a coluna) montando uma matriz \mathbf{G} . O pacote codificado t_n no n -ésimo ciclo é resultado da operação:

$$t_n = \sum_{k=1}^K s_k \cdot G_{kn} \tag{3.1}$$

As operações entre os bits são operações XOR (\oplus). Em síntese, os bits iguais a 1 na palavra G_{kn} representa quais pacotes de informação serão combinados. As Figuras 3.3 e 3.4 ilustram o processo por completo, para $K = 3$ e três ciclos realizados $n = 1, 2, 3$. É importante lembrar que cada ciclo acontece separadamente, e ao final de cada ciclo o pacote codificado é enviado.

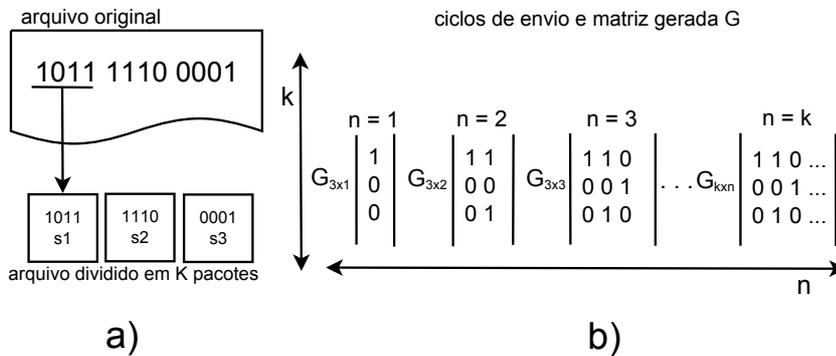


Figura 3.3: Divisão do Arquivo Original em K Pacotes e Geração da Matriz \mathbf{G}

Processo de Decodificação

Para proceder com a decodificação, de alguma forma o decodificador deve saber como foram gerados os bits da matriz \mathbf{G} . As maneiras propostas para tal são: enviar a semente geradora ou a função geradora embutida no cabeçalho do pacote de dados ou enviar os identificadores dos pacotes combinados que representam o pacote enviado em questão. Assumindo que o codificador sabe a sequência geradora dos bits utilizada na codificação, o processo de

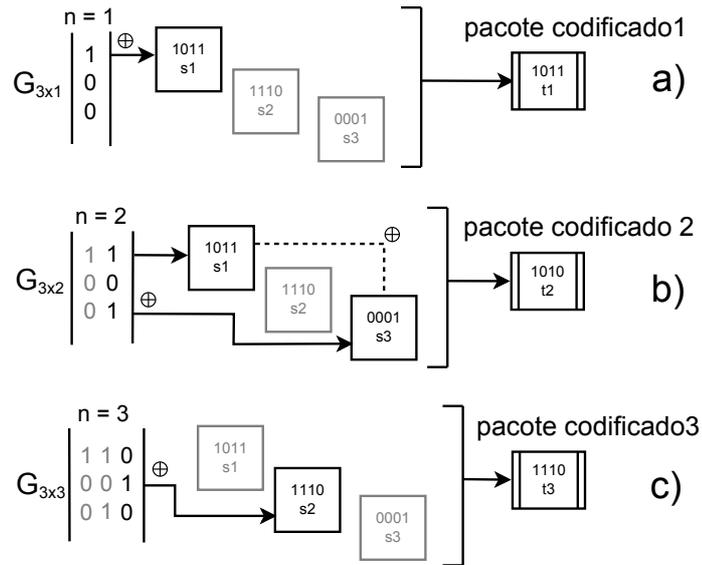


Figura 3.4: Processo de Geração dos Pacotes Codificados

decodificação é estabelecido a partir das seguintes regras: (1) Enquanto n for menor que K não é possível iniciar a tentativa de decodificação, necessitando aguardar que $n \geq K$. (2) Quando n atingir o tamanho K , significa que a matriz G é quadrada e por consequência uma tentativa de decodificação pode ser realizada. (3) Se a matriz G for inversível, a mensagem pode ser decodificada e processo finaliza. As regras podem ser acompanhadas no fluxograma da Figura 3.5. Sempre que n for maior que K é preciso fazer a combinação entre as colunas da matriz G , a fim de se encontrar uma matriz quadrada inversível.

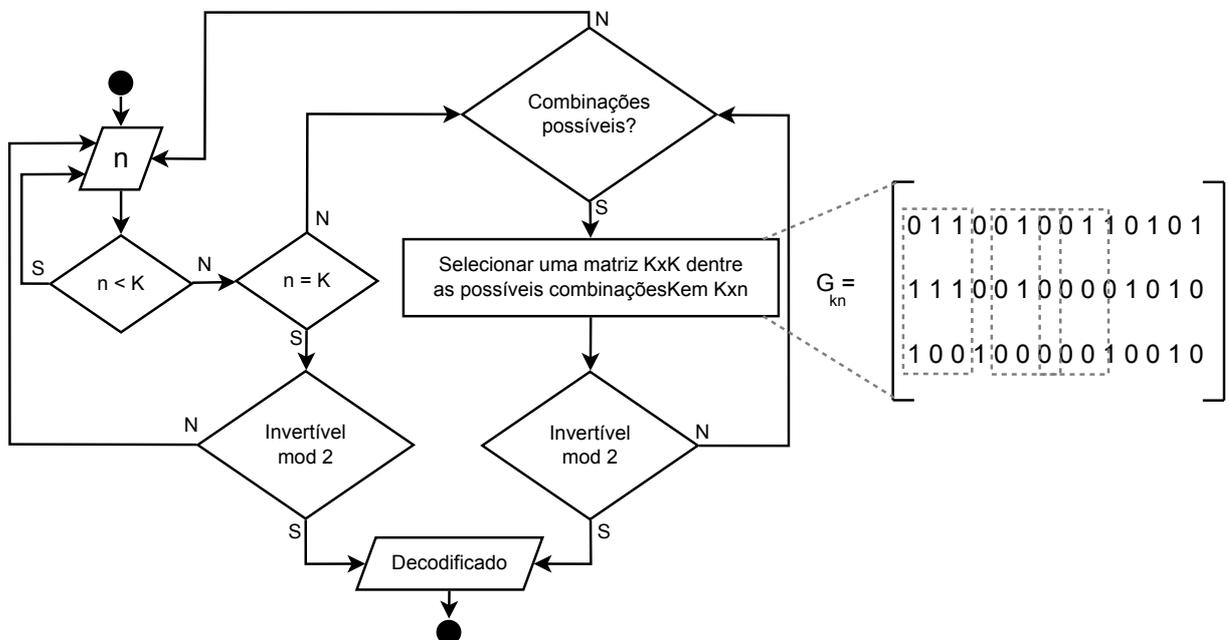


Figura 3.5: Fluxograma do Processo de Decodificação

A medida em que a codificação é realizada, a matriz envolvida no método RLF vai aumentando suas proporções, e como a decodificação dos dados envolve operações de combina-

ções para determinação de uma matriz inversível, o custo computacional para a decodificação dos pacotes aumenta em função do crescimento da matriz. Portanto, o método não é considerado muito eficiente. O esforço computacional para localizar na matriz \mathbf{G} a existência de uma matriz inversível $K \times K$ aproxima-se de $K^2/2$ operações por pacote, e o custo computacional de aplicar a operação para cálculo da matriz inversa, pré-requisito do método, aproxima-se de K^3 operações binárias [MacKay, 2005].

Finalizando o exemplo, no terceiro envio de pacotes codificados, ou seja, quando n é igual a K , a inversa é testada e existe, logo, o pacote é decodificado usando a equação

$$s_k = \sum_{n=1}^N t_n \cdot G_{nk}^{-1} \tag{3.2}$$

onde G_{nk}^{-1} representa as colunas de \mathbf{G}^{-1} e as operações também são realizadas com XOR (módulo 2). A Figura 3.6 ilustra um exemplo do processo de decodificação. Na Figura 3.6a uma parcela do pacote original é estabelecida, na Figura 3.6b mais uma parcela é obtida, e por fim na Figura 3.6c o pacote é decodificado por completo. Ao contrário da codificação, onde um pacote é enviado para cada G_{kn} gerado, a tentativa do processo de decodificação é realizado depois que $n \geq K$ pacotes foram recebidos.

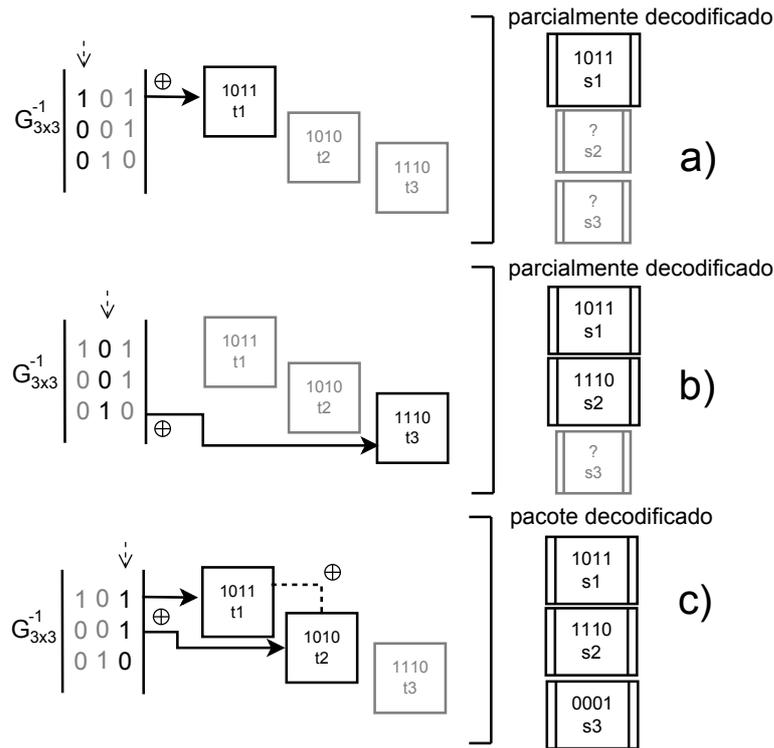


Figura 3.6: Processo de Decodificação

Particularmente, no exemplo da Figura 3.6, a decodificação acontece de fato quando $n = K$, ou seja, é o mínimo de informação suficiente para a tentativa de decodificação, considerada a situação ideal. A probabilidade, P_i , de uma matriz $K \times K$ ser inversível é estimada pela equação 3.3. Significa a probabilidade de cada coluna gerada da matriz \mathbf{G} ser linearmente independente das demais. As colunas de uma matriz são ditas linearmente independentes se, e somente se,

nenhuma das colunas pode ser obtida a partir da combinação linear das outras [Leon, 1999]. Para $K > 10$ a probabilidade de existir uma matriz inversa tende a ser aproximadamente 0,289.

$$P_i = (1 - 2^{-K}) \cdot (1 - 2^{-(K-1)}) \cdot \dots \cdot (1 - \frac{1}{8}) \cdot (1 - \frac{1}{4}) \cdot (1 - \frac{1}{2}) \quad (3.3)$$

No caso de uma matriz $K \times n$, o valor de n é o valor de $K + E$, onde E é o excesso de pacotes recebidos. Em uma matriz $K \times n$ a probabilidade de achar a inversa é dada por $1 - \delta$ e a de não encontrar é δ . Para qualquer K , a probabilidade de falha pode ser estimada por:

$$\delta(E) \leq 2^{-E} \quad (3.4)$$

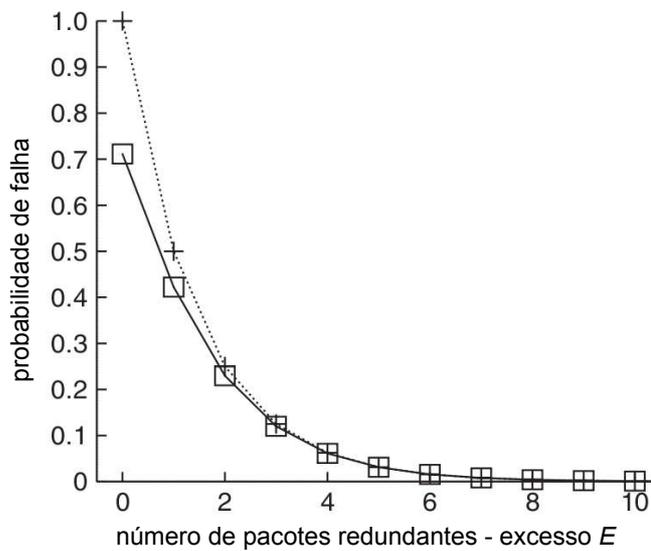


Figura 3.7: Desempenho do Método RLF [MacKay, 2005]

Tabela 3.1: Probabilidade de Decodificação do Método RLF [Hytyia et al., 2007a]

$n + E$	$1 - \delta$	$n + E$	$1 - \delta$	$n + E$	$1 - \delta$	$n + E$	$1 - \delta$
0	0,289	3	0,880	6	0,984	9	0,998
1	0,578	4	0,939	7	0,992	10	0,999
2	0,770	5	0,970	8	0,996	11+	1,000

A Figura 3.7 juntamente com dados da Tabela 3.1 representam alguns valores da probabilidade de falha (eixo y) em função do número de pacotes em excesso (eixo x), sem considerar a perda de pacotes no canal de comunicação. É possível observar que conforme aumenta o número de pacotes em excesso E , a probabilidade de não se conseguir decodificar o pacote diminui significativamente. A linha pontilhada indica o limite superior, 2^{-E} , da probabilidade de erro e a linha sólida indica que a completa decodificação não é possível como função do número de pacotes em excesso [MacKay, 2005].

3.1.5 Método LT (Luby Transform Codes)

No método LT [Luby, 2003] o custo computacional e a complexidade sobre as operações de codificação e decodificação são menores, tornando o método LT o primeiro método viável para implementação prática. Como notação o método LT utiliza grafo bipartido, diferentemente do RLF que utiliza notação matricial.

Processo de Codificação A codificação do método LT é obtida da seguinte forma:

1 - A partir de uma distribuição de probabilidade, sorteie um número, denominado grau d_n . A escolha da melhor distribuição de probabilidade para o sorteio do grau é descrita no item *Distribuições de Grau* adiante.

2 - Faça um segundo sorteio, a partir de uma distribuição uniforme, e escolha d_n nós, dentre s_1, \dots, s_K nós, a fim de obter os nós que combinados usando a operação XOR, formarão o nó codificado t_n .

Da mesma forma que no método RLF, o processo inicia com a divisão do arquivo a ser enviado em K pacotes de x bits, denominados de $s_1, s_2, s_3, \dots, s_K$, já ilustrados na Figura 3.3a. Neste caso, por simplicidade, é assumido que cada pacote tem apenas 1 bit, ou seja, o arquivo completo possui 5 bits. A Tabela 3.2 e o grafo da Figura 3.8 são complementares e demonstram as etapas da codificação do método LT.

Tabela 3.2: Exemplo do Processo de Codificação do Método LT

n	XOR (bits) S	Grau d_n (sorteio)	XOR (combinação)	t_n (codificado)
1	1 0 0 1 1	2	$s_1 \oplus s_2$	1
2	1 0 0 1 1	3	$s_1 \oplus s_2 \oplus s_4$	0
3	1 0 0 1 1	4	$s_1 \oplus s_2 \oplus s_3 \oplus s_5$	0
4	1 0 0 1 1	4	$s_1 \oplus s_3 \oplus s_4 \oplus s_5$	1
5	1 0 0 1 1	1	s_5	1
6	1 0 0 1 1	2	$s_2 \oplus s_5$	1

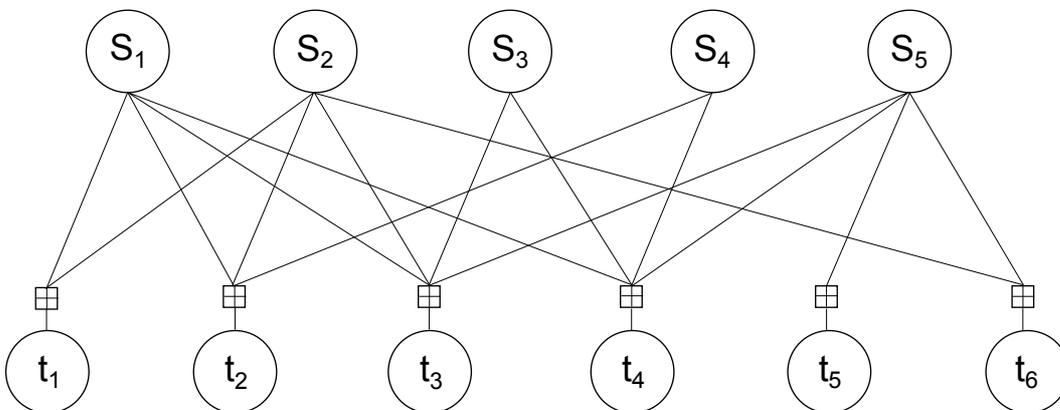


Figura 3.8: Grafo Resultante do Processo de Codificação do Método LT

Contextualizando: Na iteração n_1 o grau sorteado d_1 foi **2** (Tabela); Isso indica que partem do nó t_1 (grafo) duas arestas. Para saber aonde chegam as duas arestas que partem de t_1 foi realizado o segundo sorteio (distribuição uniforme), que por acaso indicou os nós s_1 e s_2 (Tabela); Então foi traçado as arestas de t_1 para s_1 e s_2 e combinados os nós $s_1 \oplus s_2$ (XoR) que resultaram no nó codificado t_1 com valor **1**. Da mesma forma o processo de codificação ocorre para $n = 1, 2, 3, \dots, \infty$.

Distribuições de Grau

O processo de codificação do método LT envolve dois sorteios e duas diferentes distribuições de probabilidades. O segundo sorteio é realizado com uma distribuição de probabilidade uniforme. O foco está voltado para o primeiro sorteio da codificação, por ser a parte crítica do método. Este sorteio fundamentalmente determina o início, a complexidade computacional, o andamento correto do processo e conseqüentemente o sucesso da decodificação, porque:

- a) Caso o sorteio do grau resulte num valor baixo, é provável que se tenha nós sem arestas ou conexões, impedindo a continuidade do método;
- b) Caso o sorteio do grau resulte num valor alto, a complexidade computacional aumenta, pois serão necessários mais operações para codificar e posteriormente decodificar;
- c) Caso o sorteio do grau resulte em um valor nem tão grande, nem tão pequeno, corre-se o risco do processo de decodificação não ser iniciado, uma vez que é preciso pelo menos um nó com apenas uma conexão para início da decodificação.

O sorteio ideal do grau deve garantir o início do processo, ao mesmo tempo manter a decodificação operante e evitar um custo computacional maior na codificação e decodificação. As distribuições empregadas são:

1 - Distribuição Sóliton Ideal

A distribuição Ideal Sóliton proposta em [MacKay, 2005], tem como principal objetivo garantir que a cada iteração um nó, pelo menos, tenha uma aresta. Critério este, de início e continuidade do método. Na prática, porém, ocorre que em alguns momentos da decodificação existem nós codificados que ficam sem ligação alguma, interrompendo o método. O comportamento da distribuição é apresentado na Figura 3.9. A distribuição é definida como:

$$\rho(d) = \frac{1}{K} \text{ para } d = 1 \quad (3.5)$$

$$\rho(d) = \frac{1}{d \cdot (d-1)} \text{ para } d = 2, 3, \dots, K \quad (3.6)$$

2 - Distribuição Sóliton Robusta

A fim de corrigir o problema de nós órfãos ao longo do processo de decodificação para a distribuição Ideal Sóliton, uma nova distribuição foi proposta. A distribuição Sóliton

Robusta é composta pela normalização das distribuições ρ e τ . A distribuição τ , conta com dois parâmetros a mais que ρ (c e δ), e é definida por:

$$\tau(d) = \begin{cases} \frac{s}{K} \frac{1}{d} & \text{para } d = 1, 2, \dots, (K/S) - 1 \\ \frac{s}{K} \log(S/\delta) & \text{para } d = K/S \\ 0 & \text{para } d > K/S \end{cases} \quad (3.7)$$

Sendo assim, a distribuição Sóliton Robusta é composta por:

$$\mu(d) = \frac{\rho(d) + \tau(d)}{Z}, \text{ onde } Z = \sum_d \rho(d) + \tau(d) \quad (3.8)$$

A Figura 3.9 também ilustra o comportamento de $\tau(d)$. Existem estudos concentrados em encontrar uma melhor distribuição de probabilidade para o sorteio dos graus da codificação com o intuito de tornar o método mais robusto [Rossi et al., 2008] e [Hyytia et al., 2007b].

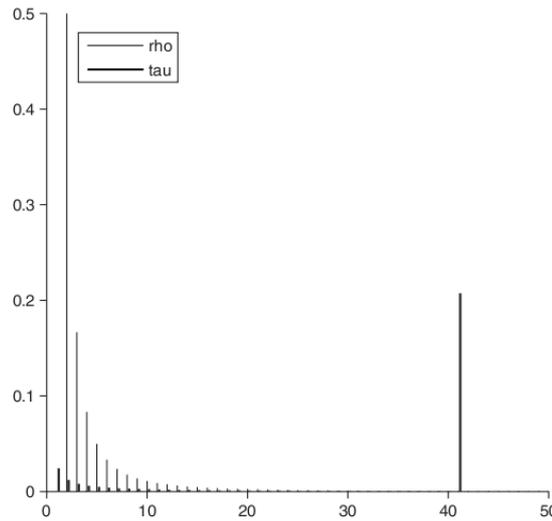


Figura 3.9: Comportamento das Distribuições ρ e τ com $K = 10000$, $c = 0,2$ e $\delta = 0,05$ [MacKay, 2005]

Processo de Decodificação

Para proceder com a decodificação, de alguma forma o decodificador deve conhecer as sementes geradoras das distribuições envolvidas na codificação ou os valores obtidos pelas mesmas. Assumindo que o codificador sabe a sequência geradora do grau e dos nós combinados na codificação, o processo de decodificação é estabelecido a partir das seguintes regras: (1) Procurar um nó t_n com apenas uma aresta conectada a um nó s_n . Caso este nó não exista, não é possível iniciar a tentativa de decodificação, necessitando aguardar mais símbolos codificados. (2) Caso o nó com apenas uma aresta existir, então, o valor do nó t_n é atribuído ao nó s_n , ou seja, $s_n = t_n$. (3) Todos os nós t_n conectados ao nó s_n sobrescrevem seus valores pelo valor da soma em módulo 2 entre t_n e s_n . (4) Remover todas as arestas conectadas ao nó s_n , e como

consequência os nós t_n conectados decrementam seu número de arestas (grau) em 1. (5) Repita as operação até que todos os nós s_n sejam determinados.

A seguir é demonstrado didaticamente como é realizada a decodificação do método LT. Considere como exemplo o grafo apresentado na Figura a. Na Figura **a**) o grafo representa o estado original do pacote codificado. A primeira execução indica que o único nó t_n com uma aresta até algum s_n é t_5 . Então, na Figura **b**), o valor de t_5 é passado para s_5 . A todos os nós t_n , conectados ao nó s_5 , é feito a soma bit a bit em módulo 2 e removidas as respectivas arestas, representado na Figura **c**). Por consequência, é diminuído em 1 o número de arestas (grau) de cada um dos nós t_n da Na Figura **c**.

Para a segunda execução, na Figura **c**), o nó t_n que tem apenas uma aresta ligando a até algum s_n é t_6 . Da mesma forma que anteriormente, $s_2 = t_6$, representado na Figura **d**).

Fazendo a soma bit a bit módulo 2 para cada nó t_n conectado a s_2 e removendo as arestas conectadas ao mesmo, obtemos o grafo representado na Figura **e**). O nó t_n com apenas uma aresta agora é t_1 em **e**). Então, $s_1 = t_1$, e o nó t_1 é desconectado. Representado na Figura **f**).

Fazendo a soma bit a bit módulo 2 aos nós conectados a s_1 , em **g**), e removendo as arestas conectadas ao mesmo obtemos o grafo da Figura **g**). Desta forma, o nó que possui apenas uma aresta é t_2 , Figura **g**). Logo, $s_4 = t_2$, elimina-se o nó t_2 , na Figura **h**).

Somando o valor de s_4 aos nós conectados, e removendo as arestas conectadas ao mesmo obtemos o grafo representado na Figura **i**). Por fim, na Figura **j**), os nós t_3 e t_4 estão conectados a s_3 e atribuem o mesmo valor ao nó s_3 , portanto o processo é finalizado com sucesso e o arquivo original recuperado.

3.2 Conclusão

Este capítulo abordou conceitos fundamentais sobre códigos fontanais bem como detalhou tecnicamente sua codificação e decodificação. O capítulo demonstrou detalhadamente os métodos LT-Codes e RLF bem as distribuições de grau envolvidas no processo do método LT-Codes. Os códigos fontanais surgem como uma interessante opção para aplicações em redes sem fio. No caso das RSSF, por exemplo, a aplicação de códigos fontanais é uma alternativa viável nas transmissões ou em aplicações de disseminação de dados entre os nós da rede. A viabilização acontece devido ao fato de os códigos fontanais possuírem baixa complexidade e custo computacional em suas codificações e decodificações, além de não utilizarem canal de retorno. Neste sentido, as RSSF podem economizar energia, principal requisito para operacionalidade, além de operar com confiabilidade na entrega das informações.

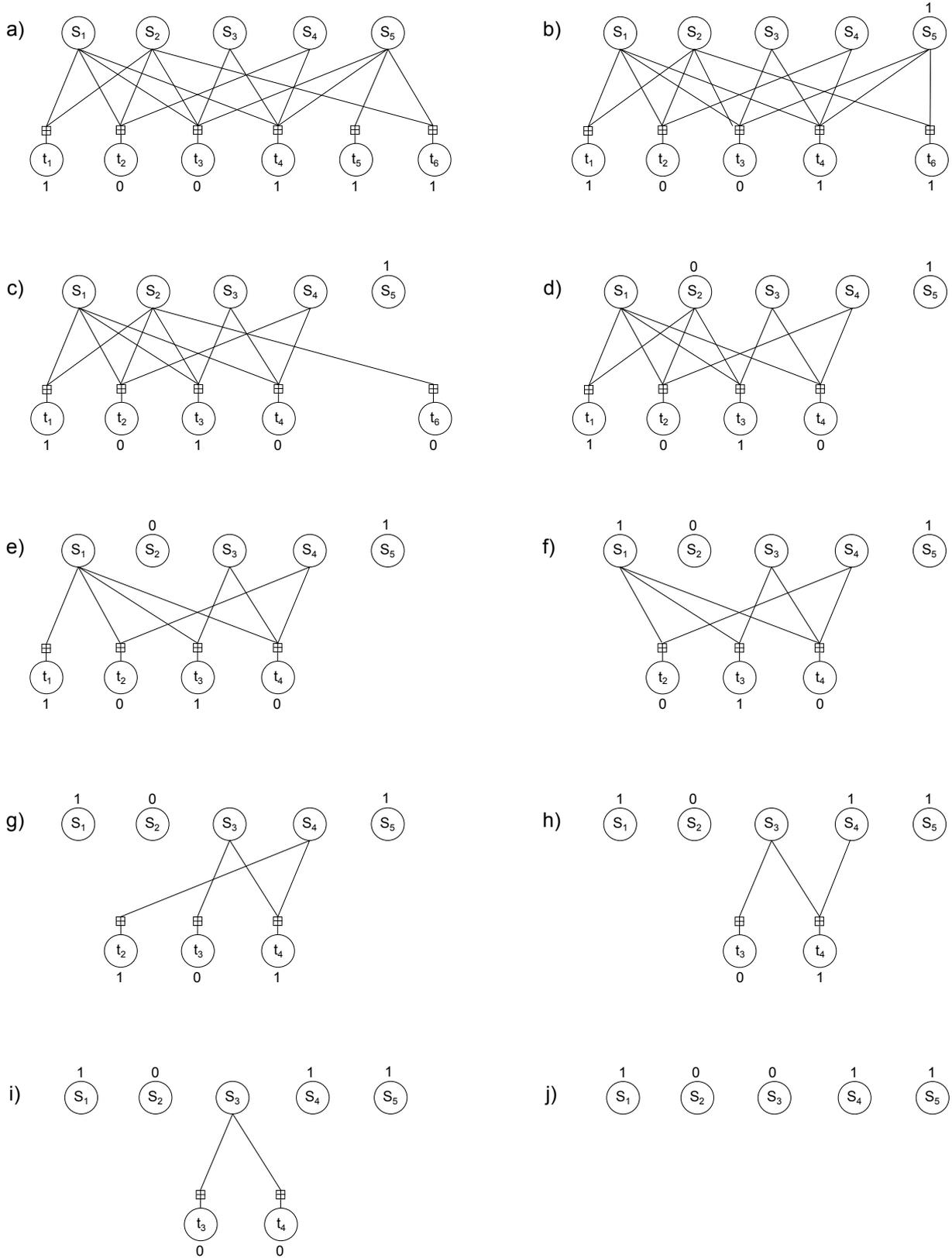


Figura 3.10: Processo de Decodificação do Método LT

Capítulo 4

Trabalhos Relacionados

Este capítulo aborda os métodos de disseminação de dados existentes e que foram considerados para o desenvolvimento do método proposto neste trabalho.

4.1 FBCast

Em [Kumar et al., 2005], foi desenvolvido e avaliado o método de disseminação FBCast que é uma extensão do método probabilístico PBCast. A idéia principal foi aliar conceitos de disseminação probabilística as estratégias de códigos fontanais. Neste mesmo trabalho duas premissas são consideradas a respeito do cenário de rede. A primeira delas diz respeito aos nós e ao conteúdo disseminado. O autor assume que não é necessário que todos os nós tenham a informação disseminada e a métrica determinante avaliada foi a confiabilidade representada pelo percentual de nós que receberam as informações disseminadas. A segunda premissa assumida em [Kumar et al., 2005], é em relação aos cenários simulados, aonde o autor distribui e incrementa o número de nós em uma mesma área (*grid*) de tamanho fixo.

4.2 MPR-BASED CODING PROTOCOL

No estudo de [Kadi, 2009] o autor propõe um novo protocolo para disseminação de informações denominado MPR-BASED CODING PROTOCOL para redes *Ad-Hoc* móveis (MANET). Tal método combina técnicas de *network coding* com o protocolo OLSR e técnicas de MPR *multipoint relay*. O estudo demonstra as vantagens do método proposto pelo autor em relação as duas técnicas de disseminação em suas formas originais.

Em um primeiro momento o autor avalia uma operação de *flooding* utilizando o método OLSR, o qual utiliza técnicas de MPR. Na segunda situação o autor realiza a mesma operação utilizando apenas técnicas de *network coding*. Por fim, são associadas pelo autor as duas técnicas (*network coding* e OLSR) para então formar a estratégia de disseminação MPR-BASED CODING PROTOCOL. O cenário de avaliação é a partir de redes simuladas com topologia aleatória e o número de nós variando entre 30 e 200. As métricas avaliadas em [Kadi, 2009] são o número de transmissões para a operação de *flooding*, o *delay* e a utilização de memória dos nós. As colisões no canal de comunicação são evitadas pelo autor em

função de que cada nó realiza a transmissão dos pacotes em uma unidade de tempo determinada.

4.3 MDeluge

O protocolo de disseminação MDeluge (*multicast Deluge*, proposto em [Zheng and Sarikaya, 2009], utiliza em sua estratégia conceitos de algoritmos de árvores e sub-árvores de disseminação (*tree based multicast algorithm*) e pode ser considerado uma extensão do Deluge [Hui and Culler, 2004], porém, viabilizado para categorias de redes de sensores estáticas e móveis. O protocolo proposto pelos autores tem como objetivo disseminar informações a partir de uma fonte para subconjuntos de nós na rede. No estudo de [Zheng and Sarikaya, 2009], a energia dos nós, a velocidade da disseminação e a mínima utilização de recurso de memória dos nós são os requisitos que o método prioriza na execução. O cenário utilizado em [Zheng and Sarikaya, 2009] foram redes em grade (*grid*) com base nos equipamentos TelosB e Stargate e simuladas em NS-2.

Em [Crepaldi et al., 2007] é desenvolvido um protocolo para disseminação de dados para RSSI que utiliza estratégia de códigos fontanais modificada para atender especificamente este tipo de rede. A adaptação realizada no protocolo objetiva alta eficiência em termos de recursos do nós e tamanhos de pacotes. O autor considera o uso de códigos fontanais adequado em função do baixo custo computacional de codificação e decodificação proporcionado pela técnica.

4.4 SYNAPSE

No estudo de [Crepaldi et al., 2007] foi desenvolvido um simulador gráfico aonde é monitorado o comportamento da rede em função de um processo de disseminação de dados na rede. O simulador demonstra em tempo real o tempo do processo de disseminação, número de pacotes transmitidos e tempo necessário para transmissão de cada bloco de dados além de algumas outras métricas do ciclo da disseminação.

Em extensão ao próprio estudo em [Crepaldi et al., 2007], os autores desenvolveram outro protocolo para disseminação denominado SYNAPSE [Rossi et al., 2008]. O SYNAPSE, assim como o protocolo FRP criado em [Crepaldi et al., 2007], mantém a utilização dos códigos fontanais no processo de disseminação. No estudo proposto em [Rossi et al., 2008], aborda e resalta a eficiência do método SYNAPSE para a fase de correção de erros dos nós. O envio de informações redundantes é o recurso que proporciona a recuperação das informações.

O cenário conta com implementações de redes em (*grid*), os nós são representações dos equipamentos TelosB com sistema operacional TinyOS. O método proposto é comparado com o método Deluge [Hui and Culler, 2004] em termos de tempo decorrido da disseminação, perda de pacotes por transmissões e em função do custo de transmissão entre os nós. O SYNAPSE é um protocolo de código fonte aberto e pode ser encontrado em [Telecomunicazioni, 2007].

Capítulo 5

Algoritmo Proposto e Resultados Obtidos

Este capítulo aborda métodos de disseminação de dados, detalhes de cenário de simulação, algoritmos propostos deste estudo, as análises de desempenho das estratégias clássicas para disseminação e, por fim, os resultados obtidos das medições simuladas.

Como proposta inicial este estudo comparou o comportamento e desempenho das estratégias clássicas de disseminação nas RSSF, FBCast e o *flooding determinístico ou inundação* que, por simplicidade, neste estudo é referenciado também como FLD. A topologias de rede adotada para as comparações entre os métodos foram realizadas em cenários com variações em densidade e tamanho de rede. Como segundo objetivo neste trabalho foi criado um método alternativo aos estudados para disseminação em RSSF que utiliza códigos fontanais e conceitos de nós estratégicos *relays* chamados RR - *Relay Replication*, que objetiva melhorar a eficiência da disseminação de dados para determinados cenários nas RSSF.

As comparações das estratégias foram validadas através de simulações, e os critérios utilizados para tal, assim como a plataforma de simulação, gráficos resultantes e algoritmos estão apresentados a seguir.

5.1 Disseminação de Dados em RSSF

As redes de sensores sem fio são utilizadas em uma variedade de aplicações de monitoramento e automação, sendo comum a existência de RSSF com dezenas de sensores. Também é frequente que a distribuição dos nós aconteça em regiões de difícil acesso, um dos motivos pelo qual as RSSF operem, em geral, em modo dito sem infra-estrutura. A inviabilidade de acesso ou manutenção física dos equipamentos, seja pela localização dos nós sensores ou pela quantidade de sensores da rede, torna o processo de operação e manutenção diferente dos aplicados às redes convencionais.

Uma das operações comuns utilizadas para manutenção das RSSF é a disseminação de informações entre os nós, que pode ser utilizada para diferentes fins, desde atualização de programas executados pelos nós, até atualização das tabelas que controlam o roteamento das informações que trafegam na rede.

Uma operação de disseminação de informações é regida por dois objetivos principais: economia de energia e entrega confiável das informações [Kumar et al., 2005]. Um dos fatores que impede a economia de energia, por exemplo, é o fato de que em uma rede densa, os nós necessitam fazer mais transmissões de um mesmo pacote devido ao alto número de colisões normalmente presente em um canal de comunicação sem fio. No que se refere ao segundo obje-

tivo, a entrega confiável, sua inexistência deixaria um ou mais nós sem receber as informações de atualização por completo, o que poderia resultar em uma rede inoperante.

Basicamente os métodos aplicados para realizar uma disseminação de informações são classificados em determinísticos e probabilísticos. O método determinístico mais comum é denominado *flooding determinístico ou inundação* [Lim and Kim, 2001]. Este método possui como critério de funcionamento o reenvio de cada pacote de dados recebido aos seus vizinhos, e estes sucessivamente, até que todos os nós recebam o conteúdo distribuído. Nos métodos probabilísticos, o processo é semelhante ao *flooding determinístico*, exceto pelo modo de decisão de reenviar o pacote. Cada nó que utilize método probabilístico, retransmite os pacotes recebidos de acordo com uma distribuição de probabilidade pré-definida [Lim and Kim, 2001].

Uma estratégia promissora envolvendo disseminação de informações nas RSSF é a utilização de códigos fontanais [MacKay, 2005] nos protocolos de comunicação. Códigos fontanais são uma metodologia de envio de informações com entrega confiável e que não exigem a necessidade de um canal de retorno para confirmação das mensagens transmitidas (*feedback channel*), normalmente utilizado em protocolos de transmissão em redes.

5.1.1 Método de Inundação

O método de disseminação de dados denominado inundação, também conhecido como *flooding determinístico* (FLD), é um método que em sua forma mais simples possui seu funcionamento regido da seguinte maneira: toda mensagem que um nó recebe deve repassar a seus vizinhos. Quando aplicado desta forma pura nas RSSF, o método é considerado proibitivo pelo fato de gerar um alto tráfego na rede. Tal tráfego excessivo, resulta em um consumo desnecessário de energia em cada nó e, como consequência, o desligamento de alguns destes. Uma demonstração de transmissões desnecessárias em uma RSSF densa que utilize o *flooding determinístico* é o *looping*, onde um nó **A** envia para um nó **B** vizinho os dados recebidos, e o nó **B**, por sua vez, encontra **A** como vizinho e repassa novamente os dados a este. O envio de dados de **A** para **B** e vice-versa, pode perdurar indefinidamente até a energia dos nós acabar.

Ainda assim, mesmo possuindo um tráfego demasiado de mensagens, em casos de RSSF não densas, o método de *flooding determinístico* pode ser considerado viável e tem aplicabilidade. Para tanto, é preciso contornar problemas básicos como o *looping*, conforme mencionado anteriormente. Tal situação pode ser contornada através de um controle de identificação de pacotes de dados recebidos pelos nós.

5.1.2 Método Probabilístico

A idéia geral envolvida em um método probabilístico é que o envio dos pacotes de dados de um nó para outro está associada a uma distribuição de probabilidade. A probabilidade da transmissão acontecer pode variar em função da distribuição e dos parâmetros configurados. Pelo fato de algumas transmissões não ocorrerem, os métodos probabilísticos como, por exemplo, o PBCast (*probabilistic broadcasting*) [Birman et al., 1999] são menos proibitivos nas RSSF, principalmente nas redes de maior densidade. Em RSSF de alta densidade os métodos do tipo PBCast limitam que um mesmo pacote de dados seja enviado mais de uma vez a um mesmo destinatário, com isso, reduzem o número de transmissões, propiciam economia de energia nos nós e melhoram a largura de banda de comunicação.

A regra envolvida no PBCast basicamente determina que todo nó que receba um pacote de dados realize um sorteio de acordo com uma distribuição de probabilidade, onde, p é a probabilidade de realizar a transmissão ou retransmitir o pacote e $(1 - p)$ é a probabilidade de não realizar a transmissão ou não reenviar o pacote recebido.

O controle de reenvio realizado pela regra probabilística em uma RSSF de alta conectividade, evita a redundância de transmissão e otimiza em número de transmissões o desempenho do PBCast em relação ao FLD. No entanto, é possível também que, em um pior caso do método PBCast, pelo sorteio resultante da distribuição de probabilidade, aconteça o mesmo número de transmissões do método FLD. Ainda assim, em redes menos densas que possuem canais de comunicação com um elevado número de colisões, a redundância do envio realizado pelo *flooding determinístico* pode ser uma alternativa viável.

Apesar do bom desempenho que o PBCast possui em redes mais densas, o método utiliza em sua comunicação a estratégia de um canal de retorno para confirmação das mensagens que são recebidas pelos nós. Dependendo da topologia da rede, um canal de retorno pode elevar o número de colisões no canal de comunicação e gerar um demasiado tráfego de informações a ponto de comprometer a rede.

5.1.3 Método Fontanal Probabilístico

Mantendo o critério dos métodos probabilísticos, o método FBCast (*Fountain Broadcast*) [Kumar et al., 2005] também utiliza em seus princípios decisões de repassar pacotes de dados aos vizinhos associado a uma probabilidade p . A diferença do FBCast em relação aos demais é que este agrega também a estratégia de códigos fontanais.

Basicamente o mecanismo de funcionamento dos métodos fontanais probabilísticos acontece da seguinte forma: Os m pacotes a serem transmitidos para os demais nós da rede são codificados em n pacotes e enviados aos nós conectados. O processo de codificação e decodificação é parte do método fontanal, conforme demonstrado no Capítulo 3. Depois de enviados aos nós vizinhos, os pacotes codificados são reenviados por estes às suas respectivas conexões se: forem pacotes considerados diferentes de outros uma vez recebidos e se satisfazem a condição de reenvio vinculada a uma probabilidade p . Do lado do receptor, uma vez recebido o pacote, este pode tentar a decodificação e reconstrução dos dados quando $k \geq n$. O FBCast também considera que o nó receptor conheça a semente geradora dos pacotes codificados ou então saiba a lista de pacotes combinados que gerou o pacote codificado recebido.

Conforme [Kumar et al., 2006], o FBCast minimiza a sobrecarga da rede e do canal de comunicação além de manter a alta confiabilidade de entrega das informações. O autor reforça o uso do FBCast para redes mais esparsas, uma vez que os nós tornam-se repetidores, e desta forma, atingem os vizinhos mais extremos e distantes da fonte.

5.2 Cenário de Avaliação

Na análise do trabalho desenvolvido por [Kumar et al., 2005] foi identificado que nosso estudo pode contribuir e estender o desenvolvimento do trabalho dos autores, ainda que, assumindo as duas situações estipuladas pelos mesmos. Em relação a primeira premissa, consideramos que uma disseminação deve acontecer por completo em toda a rede. Um processo de disseminação para atualizar tabelas de roteamento ou corrigir sistema operacional deve ser realizado em todos os nós da rede, do contrário, a rede pode ficar comprometida.

No que diz respeito a segunda premissa, a densidade da rede, consideramos restritivo e não usual para RSSF um cenário em que o número de nós é incrementado enquanto a área de distribuição dos nós permanece estática. Neste caso, estendemos o estudo de [Kumar et al., 2005] para cobrir cenários menos restritivos considerando aplicações práticas.

Objetivando representar, comparar e posteriormente propor alterações nos métodos de disseminação, utilizamos neste trabalho as mesmas fontes e critérios adotados em [Kumar et al., 2005], a fim de atingir resultados comparativos fieis ao método desenvolvido. As informações base para as simulações realizadas neste estudo tem origem em uma ferramenta denominada TOSSIM, mais especificamente em um módulo do TOSSIM [Community, 2010b] chamado *lossyBuilder*, que é responsável por gerar uma matriz de conectividade entre os nós da rede.

Para estender as estratégias de disseminação inicialmente abordamos o método mais simples, FLD (*flooding determinístico*), porém com alteração sob sua forma pura onde consideramos na implementação do mesmo o controle de pacotes recebidos pelos nós. Em sequência, foi explorado o método FBCast criado por [Kumar et al., 2005]. Neste caso foi utilizado o método em sua forma original e uma variação que adiciona ao seu formato inicial conceitos de nós estratégicos (*relays*). Por fim, foi proposto uma alternativa criada intitulada RRFB (*Relay Replication Fountain Broadcast*) para a disseminação de informações que, assim como o FBCast, utiliza conceitos de códigos fontanais e também tem seu desempenho medido quando aliado ao conceito de nós estratégicos *relays*. Os métodos são comparados a partir de simulações realizadas em iguais condições de cenário de rede, critérios probabilísticos, ambiente de simulação, parâmetros dos códigos fontanais e métodos para identificar nós estratégicos ou *relays*.

5.2.1 TinyOS e TOSSIM

O TinyOS [Community, 2010a] é um sistema operacional de código fonte aberto desenvolvido especificamente para equipamentos de redes de sensores sem fio. Assim como outros sistemas operacionais, o TinyOS é o responsável por todas as operações internas dos equipamentos sensores, controle de memória, processamento, concorrência, agendamento de tarefas, energia, dentre outros detalhes são controlados pelo TinyOS. A documentação, customização e especificações técnicas deste sistema operacional podem ser encontradas em [Levis et al., 2005] e [Levis et al., 2009] além do portal de ajuda e especificações em [Community, 2010a].

O TOSSIM é o simulador do sistema operacional TinyOS. Diversos estudos adotam o TOSSIM por ser um simulador que representa com fidelidade a execução do TinyOS em equipamentos sensores reais. Detalhes técnicos sobre o TOSSIM bem como documentação podem ser encontrados em [Levis et al., 2003] e [Levis and Lee, 2003] bem como o portal em [Community, 2010b] que contempla todos os detalhes deste simulador.

O modelo de erro de canal de comunicação, gerado pelo módulo *lossyBuilder* do TOSSIM, é uma matriz de dados com probabilidades de sucesso de transmissão entre os nós da rede. A utilização do *lossyBuilder* tem por base o estudo de [Ganesan et al., 2002] que realizou análises e simulações em uma RSSF real com equipamentos sensores utilizando o sistema operacional TinyOS. Em [Ganesan et al., 2002] foi avaliado o desempenho da disseminação de informações através do método FLD em topologia de grade com nós variando a potência do rádio em densidades muito alta, alta, média, baixa e muito baixa. Os autores avaliaram separadamente as camadas da pilha de protocolos explorando na camada física e de comunicação a

perda de pacotes e o comportamento da comunicação entre os nós. Na camada MAC (*Media Access Control*) as análises contemplaram colisão e latência e, por fim, na camada de rede e aplicação, foi analisado a estrutura das árvores enquanto do uso do FLD.

O cenário do estudo de [Ganesan et al., 2002] utilizou sensores do tipo *Rene Mote* com sistema operacional *TinyOS*, uma variação do CSMA (*carrier sense multiple access*) como protocolo de acesso ao canal de comunicação, pacotes de 38 bytes de tamanho com paginação de 30 bytes e *backoff* com intervalos entre 6 e 100 ms. As demais configurações dos sensores utilizados foram: processador com 4MHz, 8KB de memória de programação e 512B de memória para dados, 916 MHz de rádio em um canal simples e 10 kbps de largura de banda utilizando modulação OOK (*on-off Keying*).

Ainda em [Ganesan et al., 2002], foram propostos dois cenários de simulação. No primeiro deles os autores têm como resultado do experimento a perda de pacotes em função da configuração de potência de sinal dos rádios. Na segunda situação, foi avaliada propagação das informações na rede bem como a árvore resultante desta propagação e a partir de registros em *logs* da hora das transmissões foram levantadas métricas como tempo de *backoff* e número de colisões. O autor conclui que embora o método *flooding* apresente relativa simplicidade enquanto sua implementação, seu comportamento demonstra complexidade de análise. Por fim, o autor ressalta que a perda de pacotes aumenta em função da distância entre os nós, a demora para receber as informações acontece em função da colisão da camada MAC e que a lentidão da comunicação nas redes sem fio é resultante da combinação das colisões e da elevada distância entre os nós.

O estudo realizado por [Kumar et al., 2005] utilizou o modelo de canal gerado pelo *lossyBuilder*. Em nosso estudo adotamos o mesmo modelo por critérios de comparação. As densidades de rede foram classificadas em muito alta, alta, media, baixa e muito baixa, e representadas no simulador através de *grids* espaçados em 20, 25, 30, 35 e 40 pés. Os *grids* variam em número de nós em 9, 16, 25 e 36. Apesar do TOSSIM permitir toda a representação de equipamentos de RSSF, neste trabalho utilizamos especificamente apenas a representação da camada física e de conectividade da rede gerada pelo módulo *lossyBuilder* do TOSSIM. As simulações deste trabalho foram realizadas em ambiente de programação Matlab[®] e o controle de acesso ao meio (MAC) não é avaliado nas simulações [Williams and Camp, 2002].

Assim como em [Williams and Camp, 2002], justificamos a ausência da camada MAC para que seja possível comparar e avaliar o desempenho dos métodos enquanto a eficiência destes para disseminação de dados na rede. A omissão da camada MAC permite que efeitos de mobilidade e congestionamento, presente em RSSF, sejam desconsiderados. Em segundo caso, não é objetivo deste trabalho avaliar redes móveis ou efeitos da camada MAC no processo de disseminação. A idéia é determinar através de comparações qual o método possui melhor desempenho em um processo de disseminação em função do cenário da rede. A rede em grade (*grid*) auxilia a avaliação isolada das estratégias de disseminação, uma vez que, seu comportamento é uniforme e, ao contrário de topologias aleatórias, exigem um número menor de variações. Por fim, entendemos que uma disseminação de dados pode ser realizada em etapas, ao longo da operação normal da rede e não necessariamente deve ocorrer em um tempo pré-estabelecido. Também é possível realizar este processo aproveitando uma possível ociosidade da rede. Portanto, nas comparações das estratégias de disseminação e para o método proposto, a métrica avaliada é o número médio de mensagens transmitidas na rede até que as informações estejam em todos os nós da rede.

Para o processo da disseminação de informações foi assumido neste estudo que as mensagens disseminadas são divididas em K pacotes. Estes K pacotes são codificados e enviados a todos os nós da rede. Quando um nó recebe K' pacotes distintos codificados, onde $K' = (1 + \varepsilon)K$, a mensagem original pode então ser reconstruída pelo nó receptor. O parâmetro ε é um número configurado para codificação da fonte. Foi adotado neste trabalho, assim como em [Kumar et al., 2005], $K = 10$ pacotes e para a tentativa de decodificação $K' = 14$ pacotes recebidos. Os parâmetros são configurados antes do início do processo de disseminação.

Para uma topologia em *grid* com N nós, o *lossyBuilder* gera uma matriz canal de probabilidades, p_{ij} , que representa a probabilidade de erro de BIT entre os nós i e j , conforme definido na equação 5.1.

$$\mathbf{P} = \begin{pmatrix} p_{11} & \dots & p_{1N} \\ p_{21} & \dots & p_{2N} \\ \vdots & \ddots & \vdots \\ p_{N1} & \dots & p_{NN} \end{pmatrix}, \quad (5.1)$$

Dependendo do esquema de correção de erros FEC (*Forward Error Correction*) empregado no nó sensor [Wicker, 1995] é possível determinar a probabilidade de erro com base no modelo de canal simétrico BSC (*Binary Symmetric Channel*). Os dois tipos de códigos de blocos BC (*Block Code*) mais comuns utilizados nos nós sensores possuem parâmetros $(n, k) = (13, 8)$ e $(16, 8)$, que corrigem 1 e 2 erros respectivamente.

O parâmetro k representa o tamanho de bloco de entrada do codificador em bits e o parâmetro n representa o tamanho de bloco de saída do codificador em bits. Analisamos o desempenho em termos da probabilidade de erro de quadro FER (*Frame Error Rate*) para os dois códigos de bloco citados acima. Também foi investigado o desempenho de FER para um código corretor de erros do tipo código convolucional de baixa complexidade com taxa igual a $1/2$, em um canal BSC com probabilidade de transição p .

O gráfico da Figura 5.1 demonstra que para uma probabilidade p , os códigos convolucionais (CC) possuem um melhor desempenho em relação aos códigos de bloco. Apesar do método convolucional possuir complexidade de decodificação maior do que ambos os métodos de códigos de blocos e utilizar mais energia neste processo, a redução FER proporcionada pelo método CC compensa este gasto de energia.

5.3 Análise das Estratégias de Disseminação

Os métodos FLD e FBCCast que foram propostos respectivamente em [Lim and Kim, 2001] e [Kumar et al., 2005] possuem seu funcionamento especificado pelo pseudo-código do Algoritmo 1. Ambos os métodos implementados neste trabalho respeitam as mesmas regras dos métodos em sua forma original, conforme Seção 4.1, salvo pelo fato de possuírem em sua implementação o controle dos pacotes recebidos (ID) e, portanto, contornam o problema de ciclos infinitos de retransmissão de pacotes. Esta regra é descrita na linha 4 do pseudo-código que representa ambos os algoritmos, FLD e FBCCast. Os demais métodos propostos neste trabalho também possuem implementados o controle de identificação de pacotes recebidos (ID).

Neste estudo comparativo, conforme proposta inicial, também foi associado aos métodos FLD e FBCCast o conceito de nós estratégicos *relays*, resultantes da aplicação do algoritmo

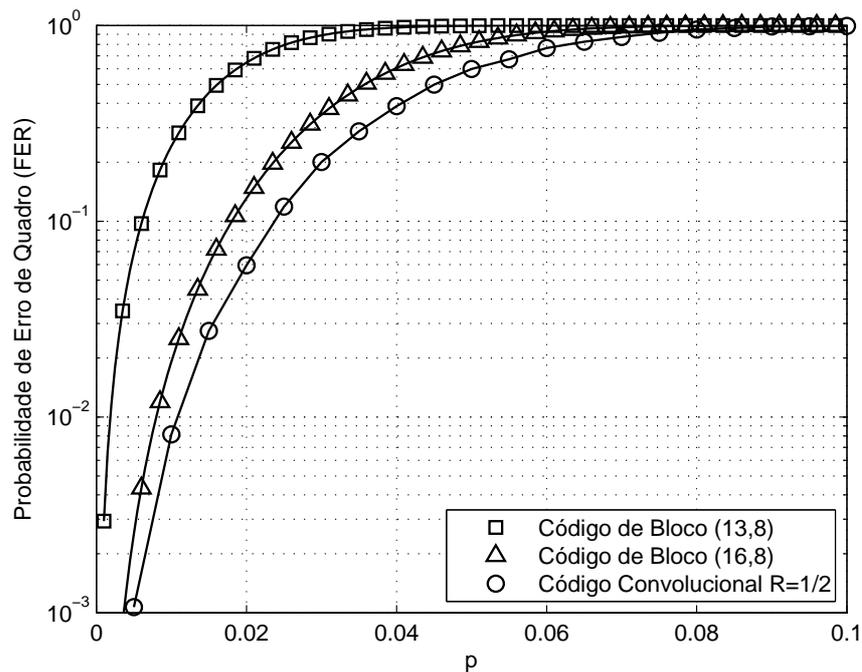


Figura 5.1: Desempenho dos Blocos de Códigos Convolutivo em Canal Binary Symmetric Channel (BSC).

de Kruskal e intitulado MDST (*Minimum Dissemination Spanning Tree*). Os dois métodos adaptados FLD+MDST e FBCast+MDST também estão representados no pseudo-código do Algoritmo 1 e diferenciam-se de suas características originais em função do uso do MDST. O parâmetro \mathbf{T} , linha 2 em Algoritmo 1, representa a árvore binária construída a partir da topologia da rede e, em \mathbf{L} , são atribuídos os nós folhas de \mathbf{T} e que não serão nós fontes repetidores conforme linha 9 do Algoritmo 1.

Nos métodos que utilizam o MDST, algumas das conexões dos nós do *grid* são eliminadas pela criação da árvore mínima que o algoritmo de Kruskal estabelece. A Figura 5.2 demonstra um exemplo do resultado da utilização deste algoritmo. Na Figura 5.2 **a)** está ilustrada uma rede em grade com 9 nós e suas conexões originais. Na Figura 5.2 **b)** esta ilustrada a rede resultante após a utilização do algoritmo de Kruskal. Para formar a árvore mínima o algoritmo de Kruskal elimina arestas de maior peso e desconsidera as arestas que possivelmente formam ciclos entres os nós.

Neste trabalho, o algoritmo de Kruskal estabelece a árvore em função dos pesos vinculados nas arestas que representam as probabilidades de sucesso das transmissões representadas pela matriz canal gerada pelo módulo *lossyBuilder* do TOSSIM.

De acordo com a Figura 5.2 **a)** e a utilização do método de FLD, tem-se todos os nós repassando os pacotes de dados recebidos. Já no cenário da Figura 5.2 **b)**, tem-se a limitação de algumas conexões e, portanto, nem todos os nós distribuem os pacotes recebidos. O mesmo princípio se aplica para o método FBCast que utiliza a MDST, lembrando apenas que no FBCast cada nó ainda retransmite com uma probabilidade p .



Figura 5.2: Resultado da Aplicação do Método de Kruskal.

Algorithm 1: FLD / FLD+MDST / FBCast

```

1 Inicialização dos parâmetros para os nó  $i$ :
2    $L \leftarrow$  Determinar os nós folhas de  $T$ ;
3    $p_i \leftarrow$  Probabilidade de retransmissão do nó;
4   Define  $R_i = \{\}$ : Lista dos identificadores dos pacotes recebidos  $c_k$ ;

5 Procedure Recebimento( $c_k$ )
6 {
7   if ( $c_k \notin R_i$ ) then
8      $R_i \leftarrow R_i \cup c_k$ ;
9     if ( $i \notin L$ ) then
10      | Retransmitir  $c_k$  com probabilidade  $p_i$ ;
11      end
12   end
13 }
```

Após a adaptação dos métodos FLD e FBCast, foi desenvolvido neste estudo um novo método de disseminação baseado em códigos fontanais entitulado RR. O método RR em sua essência tem por princípio repassar a fonte disseminadora para os demais nós da rede. Diferentemente dos outros métodos fontanais em que a fonte codificadora é única e mantém seu envio para toda rede a partir de um único nó, neste método, a fonte se replica na rede e os nós vão tornando-se fontes ao longo da disseminação. No caso de uso do RR com MDST, o critério de funcionamento do método permanece, salvo novamente ao fato do número de nós diminuir devido a aplicação do algoritmo de Kruskal.

O pseudo-código do Algoritmo 2 especifica o método RR desenvolvido neste estudo. A principal diferença em relação aos demais métodos está no início de uma nova fonte codificadora, conforme especificado na linha 11 do Algoritmo 2.

Para a análise dos métodos de disseminação este trabalho adotou um cenário RSSF específico representado por um *grafo* $G = (V, E)$, onde V representa os nós (vertices) e E representa as conexões entre os nós (arestas), que pode ser monitorada nos nós sensores através do RSSI (*Received Signal Strength Indicator*) quando em uma rede real. Normalmente nas redes sem fio, a qualidade de sinal está diretamente ligada a confiabilidade da conexão entre os nós e tal métrica pode ser especificada pela probabilidade de sucesso ou falha de transmissão. A partir da rede especificada em formato de grafo $G = (V, E)$ foi adotado para cada vértice $e \in E$

Algorithm 2: RR / RR+MDST

```

1 Inicialização dos parâmetros para os nó  $i$ :
2    $\mathbf{L} \leftarrow$  Determinar os nós folhas de  $\mathbf{T}$ ;
3    $p_i \leftarrow$  Probabilidade de retransmissão do nó;
4   Defina  $\mathbf{R}_i = \{\}$ : Lista dos identificadores dos pacotes recebidos  $\mathbf{c}_k$ ;
5 Procedure Tecebimento( $\mathbf{c}_k$ )
6 {
7 if ( $\mathbf{c}_k \notin \mathbf{R}_i$ ) then
8    $\mathbf{R}_i \leftarrow \mathbf{R}_i \cup \mathbf{c}_k$ ;
9   if ( $i \notin \mathbf{L}$  AND  $|\mathbf{R}_i| \geq K'$ ) then
10    Decodificar mensagem:  $\{\mathbf{m}_1, \dots, \mathbf{m}_K\}$ ;
11    Iniciar novo codificador fontanal  $\rightarrow \mathbf{c}_j$ ;
12    Transmitir  $\mathbf{c}_j$  com probabilidade  $p_i$ ;
13  end
14 end
15 }
```

um peso $w_e \in \mathbb{R}$ que representa a probabilidade de sucesso de transmissão. Para os métodos que utilizam MDST a identificação dos nós estratégicos é realizada através do método de Kruskal [Cormen, 2001] que encontra a árvore mínima $\mathbf{T} = (\mathbf{V}, \mathbf{E}')$ onde $\mathbf{E}' \subseteq \mathbf{E}$, determinada pelo menor valor de soma do peso de todos os nós ($\sum_{e \in \mathbf{E}'} w_e$) sem formação de ciclos.

A utilização do algoritmo de Kruskal exige um processamento centralizado para a construção da árvore mínima. Esta não é uma metodologia para uma aplicação prática em RSSF. Contudo a montagem da árvore mínima pode ser feita de forma distribuída durante a operação normal de uma RSSF utilizando algoritmos como os propostos em [Zh and Sarikaya, 2007]. Não é o objetivo deste trabalho desenvolver tais algoritmos distribuídos para a criação de MDST.

Do ponto de vista de avaliação de desempenho das técnicas de disseminação de dados propostos, podemos partir da premissa que a MDST foi previamente estabelecida. Desta forma, estabelecemos a MDST usando o algoritmo de Kruskal no início do processo de simulação.

Algorithm 3: Algoritmo de Kruskal

```

1  $\mathbf{T} = \{\}$ : Definir as arestas do MDST;
2 while ( $\mathbf{E} \neq \{\}$ ) do
3   Selecione  $e \in \mathbf{E}$  aonde o peso é o menor ;
4   Remover  $e$  de  $\mathbf{E}$ ;
5   if ( $\mathbf{T} \cup \{e\}$ ) não forma ciclo no MDST then
6      $\mathbf{T} \leftarrow \mathbf{T} \cup \{e\}$ ;
7   end
8 end
```

5.4 Resultados da Avaliação

Na etapa inicial de nosso estudo, realizamos uma comparação do método clássico FLD com o método modificado utilizando a MDST, que denominamos FLD+MDST. A avaliação foi realizada através de simulação assumindo as topologias, parâmetros e caracterizações do canal conforme especificados anteriormente. A métrica de comparação utilizada é o número médio de mensagens transmitidas por nó até a completa disseminação das informações para toda a rede.

Inicialmente avaliamos o desempenho das duas estratégias em topologia de grade *grid* com 16 nós, assumindo espaçamentos entre os nós de 20, 25, 30, 35 e 40 pés. Os resultados obtidos são apresentados na Figura 5.3. É importante ressaltar que os valores obtidos da métrica de desempenho estão representados na figura de forma normalizada e em função de cada espaçamento do *grid*.

Podemos observar que para cenários de rede com maior densidade, ou seja, espaçamentos de 20 e 25 pés o método FLD+MDST proporciona uma redução no número de mensagens transmitidas até a completa disseminação das informações para todos os nós da rede. Contudo, para cenários com média e baixa densidade (conectividade) o método FLD é significativamente mais vantajoso.

O método FLD demonstra melhor desempenho nos cenários de 30, 35 e 40 pés em relação ao FLD+MDST, uma vez que nestes cenários a conectividade dos nós é prejudicada devido a distância ou interferência do sinal de comunicação e, portanto, as conexões redundantes tornam-se uma alternativa interessante na disseminação de dados. Em contrapartida, quando a rede possui uma alta densidade, 20 e 25 pés, a proximidade dos nós mantém em boas condições a conectividade da rede, os nós têm suas transmissões sem falhas e acabam por processar de diversos vizinhos em seu alcance os pacotes disseminados. Neste caso, a utilização do algoritmo de MDST limita algumas destas conexões e, a partir desta restrição, alguns nós não repassam mais as informações que recebem. Assim, a rede consegue fazer a disseminação com uma troca menor de mensagens entre os nós que, por sua vez, representa economia de energia, recurso essencial nas RSSF.

Embora os resultados apresentados na Figura 5.3 seja para uma topologia em *grid* com 16 nós, ($N = 4$), também foram realizadas simulações para os casos de topologias com 9, 25 e 36 nós na rede. Os resultados e conclusões obtidas nestes casos são similares ao caso de 16 nós e portanto não apresentados.

Para exemplificar a real quantidade de pacotes transmitidos na rede por cada nó no cenário avaliado, apresentamos na Figura 5.4 os valores não normalizados do número de mensagens coletados a partir das simulações. Observa-se que para o caso de topologias com densidade de nós muito baixa, 40 pés, o processo de disseminação pode gerar um número excessivo de mensagens para o método FLD+MDST.

A comparação dos métodos FBCast e FBCast+MDST foi feita sob os mesmos cenários de rede citados anteriormente, salvo que assumimos $p = 0,9$ em função do parâmetro exigido para o FBCast. Os resultados obtidos são apresentados nas Figuras 5.6 e 5.7. Neste caso podemos observar que o método FBCast+MDST apresenta apenas uma vantagem para a topologia com alta densidade de nós (20 pés). Estão ocultas as variações do método FBCast na Figura 5.6 em função da normalização dos resultados da simulação. De forma análoga a Figura 5.4 também apresentamos na Figura 5.7 os valores não normalizados no número de mensagens trocadas entre os nós na rede.

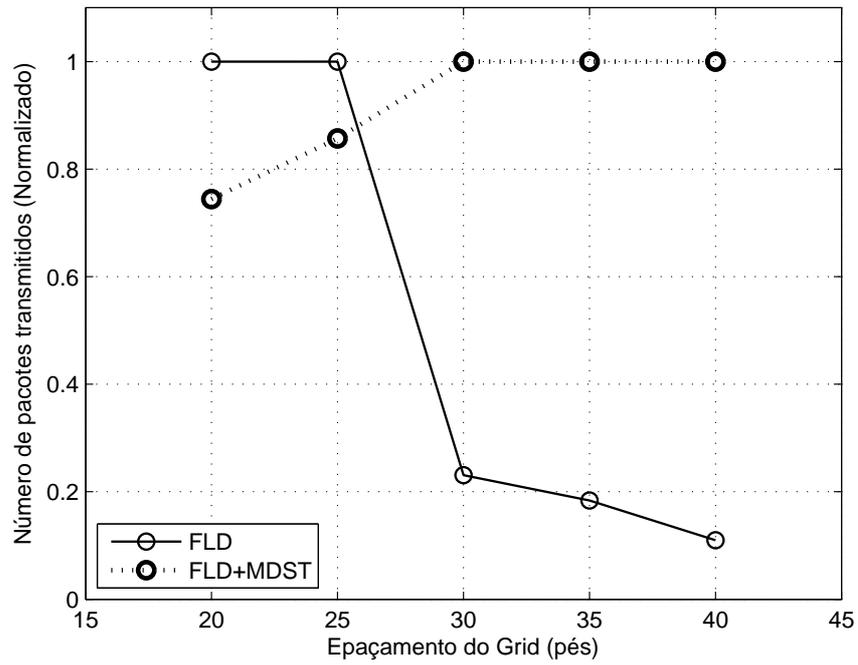


Figura 5.3: Métodos FLD e FLD+MDST

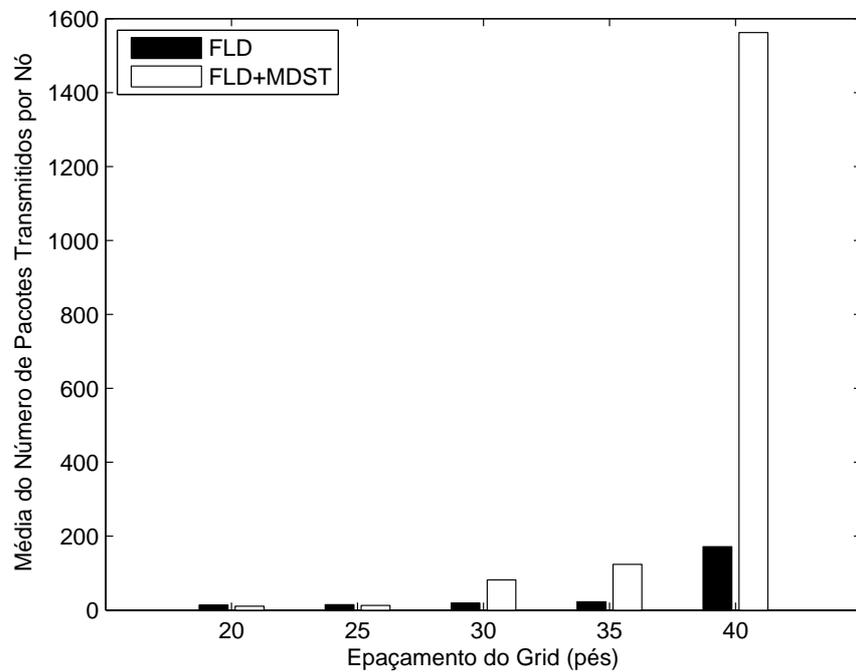


Figura 5.4: Métodos FLD e FLD+MDST

Em uma segunda etapa realizamos uma avaliação dos métodos FBCast e FBCast associado com a técnica MDST. Conforme já mencionado anteriormente, o método FBCast proposto em [Kumar et al., 2005] utiliza códigos fontanais conjuntamente com uma probabilidade p de retransmissão para cada nó da RSSF. No estudo realizado em [Kumar et al., 2005] foi utilizado um valor $p = 0,9$. Contudo, realizamos um estudo complementar do método FBCast para avaliarmos o seu desempenho em termos da probabilidade de retransmissão p . Simulamos topologias de diferentes densidades de nós, caracterizadas de muita alta, alta, média, baixa e muito baixa e representadas nas simulações por 20, 25, 30, 35 e 40 pés para valores de p variando entre 0,3 e 1. Os resultados obtidos são apresentados na Figura 5.5.

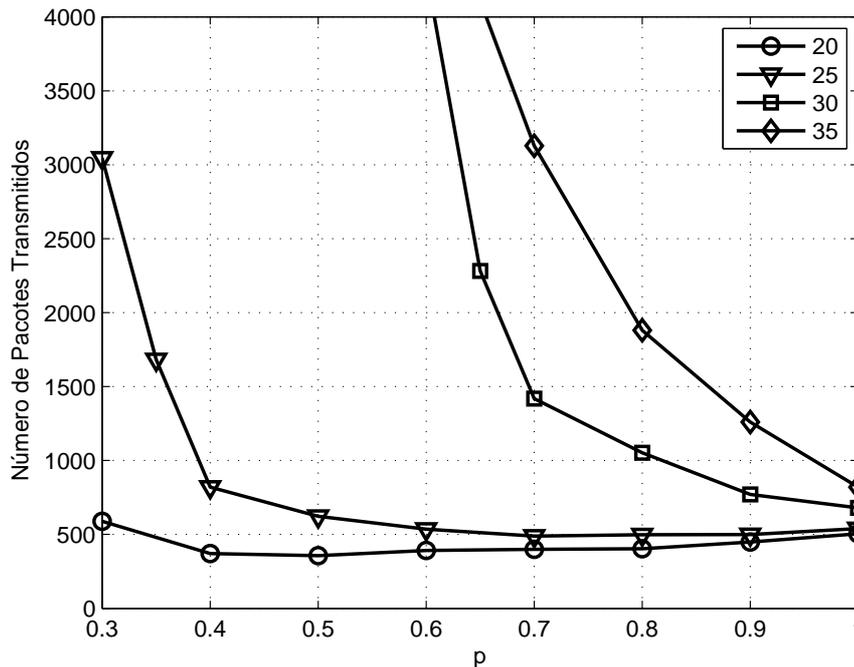


Figura 5.5: Desempenho para diferentes probabilidades de retransmissão

Para cada densidade existe um valor de p que resulta em um valor mínimo de mensagens transmitidas para disseminação na rede. Nos espaçamentos de topologia de 20, 25, 30 e 35 pés estes valores de p corresponde aproximadamente a 0,6, 0,8, 1 e 1. Identificamos portanto que existe um valor específico de p ótimo para cada densidade da rede. Apesar do resultado de p nas topologias de 20 e 25 pés não confirmar um valor ótimo em 0,9, adotamos este valor primeiramente em função dos critérios comparativos ao trabalho de [Kumar et al., 2005] que utiliza o mesmo valor. Além disso, o valor em $p = 0,9$ pode ser considerado ótimo para a maioria dos cenários simulados. Mesmo que não confirmado para cenários de 20 e 25 pés um valor mínimo em $p = 0,9$, a diferença é sutil e nos permite adotar tal valor sem comprometer os resultados das simulações.

Os gráficos demonstrados nas Figuras 5.8 e 5.9 comparam os quatro métodos apresentados FLD, FLD+MDST, FBCast e FBCast+MDST. É possível visualizar que nas redes densas, 20 e 25 pés, os métodos FBCast+MDST e FLD+MDST possuem um melhor desempenho para a métrica simulada. Dentre as estratégias que utilizam o MDST, o FLD+MDST obteve melho-

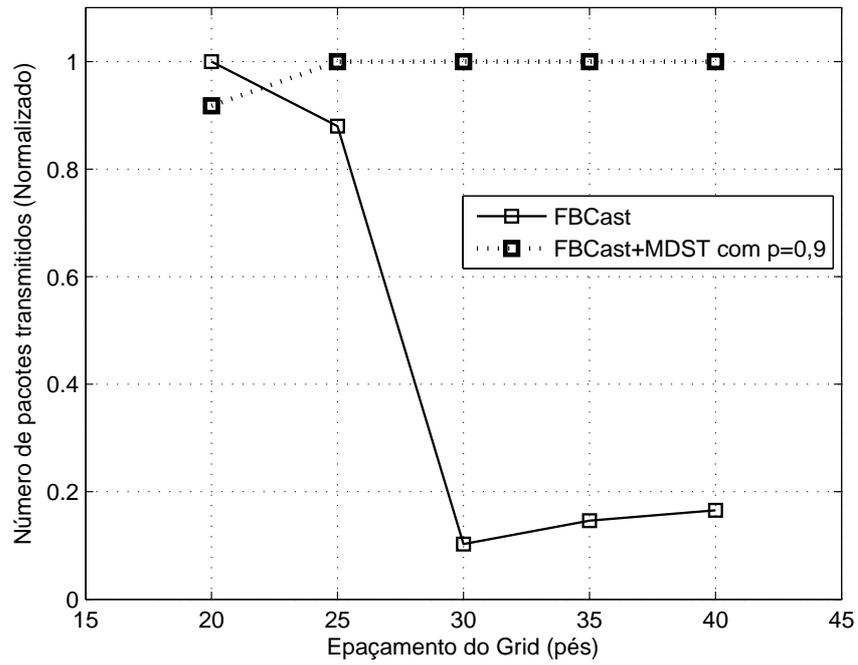


Figura 5.6: Métodos FBCast e FBCast+MDST com $p = 0.9$

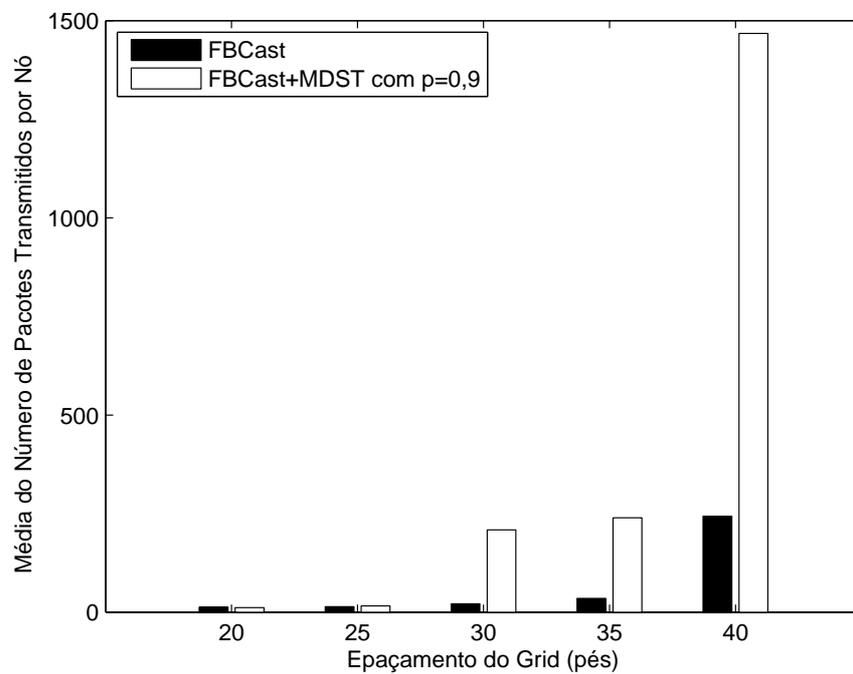


Figura 5.7: Métodos FBCast e FBCast+MDST com $p = 0.9$

res resultados em relação ao FBCast+MDST. A redução dos nós retransmissores estabelecida pela utilização da MDST aliada a redução natural imposta pela probabilidade p dos nós no FBCast limitam em excesso o número de nós retransmissores comprometendo o desempenho da estratégia FBCast+MDST.

O gráfico representado pela Figura 5.9 demonstra os valores não normalizados do número de mensagens transmitidas pelos nós. É visível que o número de mensagens enviadas para a disseminação eleva-se muito nos casos em que existe limitação dos nós que operam como repetidores. Ambos os métodos com MDST não confirmaram bom desempenho no cenário de baixa densidade, 35 e 40 pés.

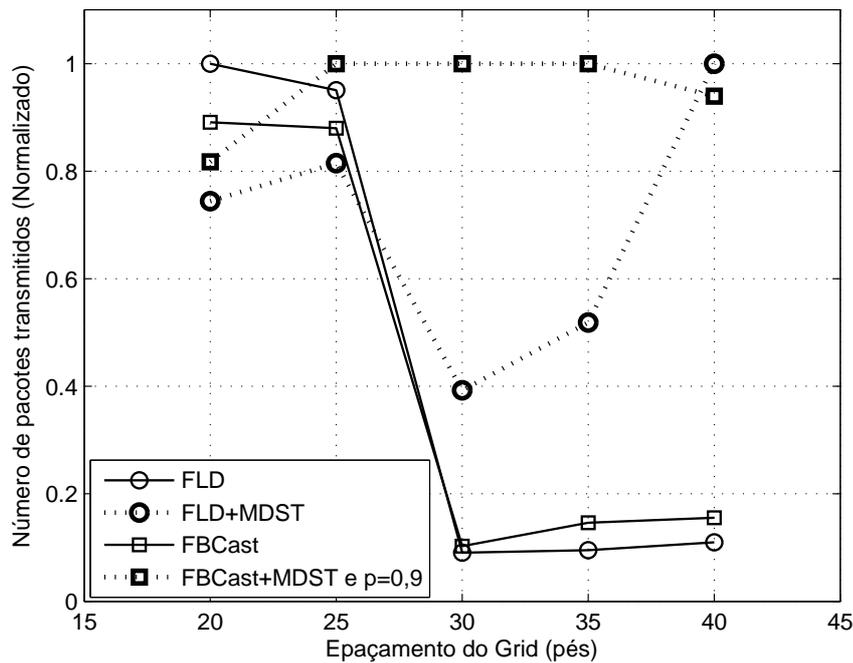


Figura 5.8: Métodos FLD, FLD+MDST, FBCast e FBCast+MDST com $p = 0.9$

A partir da comparação e avaliação dos métodos de disseminação FLD e FBCast, foi criado um novo método para o processo de disseminação denominado RR (*Relay Replication*) em que associamos conceitos de códigos fontanais, nós estratégicos *relays* e, posteriormente, também combinamos com a utilização da MDST. O objetivo deste método foi reduzir ainda mais o número de mensagens trocadas pelos nós da rede para uma completa disseminação de dados. Os cenários de rede envolvidos no estudo do método RR e RR+MDST permaneceram iguais aos cenários aplicados aos demais métodos de disseminação anteriormente apresentados.

Ao contrário dos demais métodos (FLD, FLD+MDST, FBCast e FBCast+MDST) o RR não mantém apenas a fonte inicial operando o envio dos pacotes aos demais nós. No método RR os nós estratégicos *relays* identificados a partir da aplicação do MDST é que assumem o controle de ser fontes ao longo da operação da rede, ou seja, as fontes são replicadas estrategicamente na rede.

É possível visualizar os resultados do desempenho dos métodos RR e RR+MDST nos gráficos representados pelas Figuras 5.10 e 5.11. Assim como nos demais resultados apresenta-

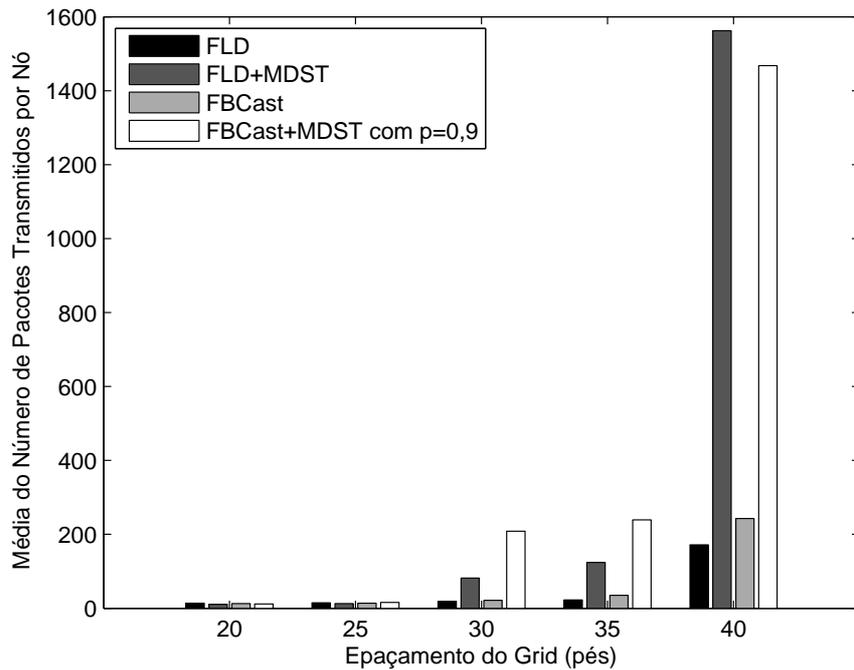


Figura 5.9: Métodos FLD, FLD+MDST, FBCast e FBCast+MDST com $p = 0.9$

dos em que a MDST está presente na estratégia de disseminação, o RR+MDST obteve melhor desempenho em relação a metodologia em sua forma original. Da mesma forma que na Figura 5.6, no gráfico da Figura 5.10, a variação dos valores coletados do método RR estão ocultos devido a normalização aplicada. Em complemento, demonstramos no gráfico da Figura 5.11 os valores não normalizados do número de mensagens trocadas na rede e observa-se neste que em todas as densidades simuladas existe um número maior de mensagens trocadas pelos nós aonde a MDST não está presente.

A partir da análise do gráfico da Figura 5.12 podemos afirmar que o método RR apresenta o melhor desempenho entre todos os demais métodos estudados neste trabalho para os cenários de alta densidade (20 e 25 pés). Além deste resultado, ambos os métodos RR e RR+MDST apresentam o melhor desempenho para cenários de baixa densidade, pontualmente 40 pés. No gráfico da Figura 5.12 estão apenas demonstrados conjuntamente os melhores resultados obtidos nas simulações dos métodos estudados.

5.5 Análise Complementar dos Métodos FLD e RR

As simulações realizadas anteriormente neste trabalho demonstraram o comportamento dos métodos FLD e RR e seus respectivos desempenhos nos cenários propostos entre alta e baixa densidades. A partir da análise dos resultados, resolvemos então fazer uma investigação analítica em que pudéssemos encontrar em que condições de canal é que o método RR passava a ter melhor desempenho em relação ao FLD.

Para tal investigação adotamos um cenário de rede simples, de topologia sequencial e constituída de N nós ($N - 1$ enlaces) representada pela Figura 5.13. A partir deste modelo

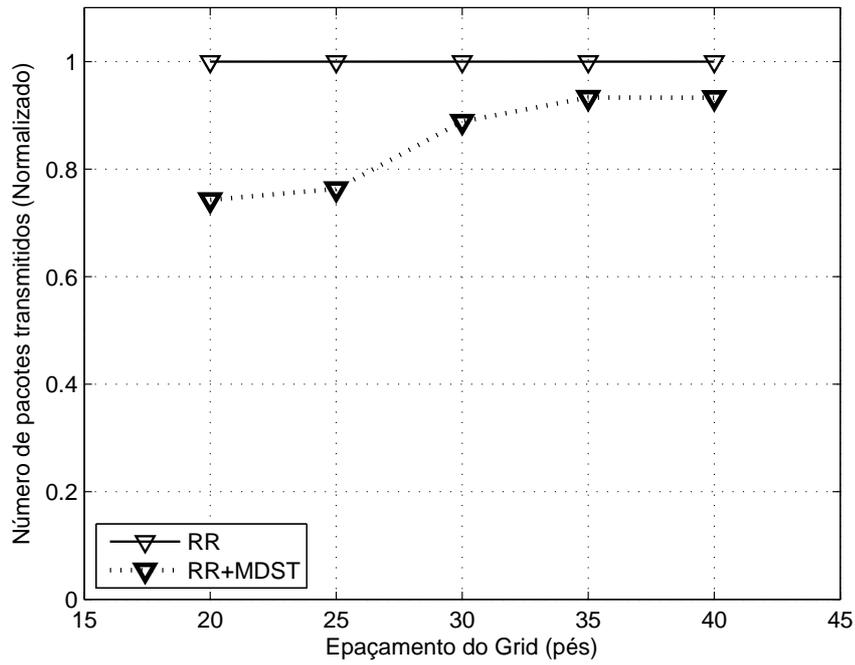


Figura 5.10: Métodos RR e RR+MDST

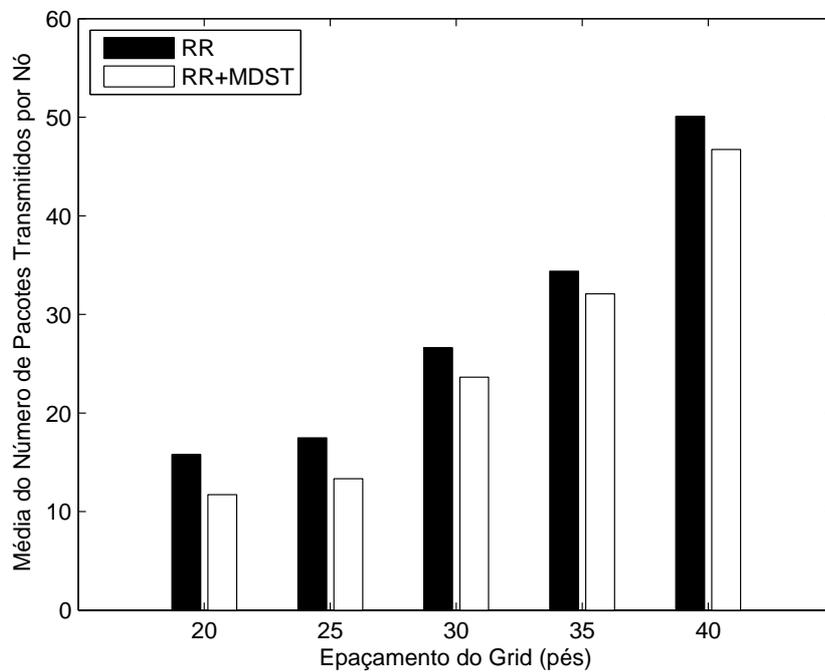


Figura 5.11: Métodos RR e RR+MDST

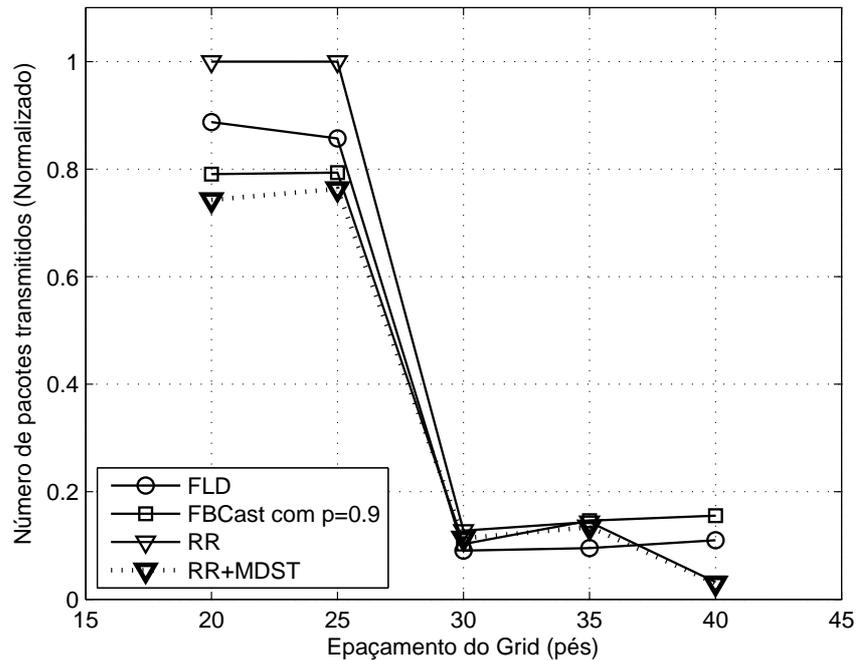


Figura 5.12: Métodos FLD, FBCast+MDST com $p = 0.9$, RR e RR+MDST



Figura 5.13: Topologia de Rede Sequencial

estabelecido, podemos então estimar o número de mensagens transmitidas N_p até que todos os nós recebam os dados distribuídos. Tal estimativa é determinada para os métodos FLD e RR e estão representados respectivamente nas equações 5.2 e 5.3 abaixo.

$$E_{FLD} = N_p \left(\prod_{l=1}^{N-1} \frac{1}{1-p_l} \right), \quad (5.2)$$

$$E_{RR} = N_p \left(\sum_{l=1}^{N-1} \frac{1}{1-p_l} \right). \quad (5.3)$$

O parâmetro p_l representa a probabilidade de erro de transmissão da conexão l . A Figura 5.14 demonstra o comportamento resultante da simulação de uma rede com $N = 4$ nós. Assumimos $p = p_1 = p_2 = p_3$ para determinar o ponto em que as estratégias invertem seus comportamentos. Pode-se observar de acordo com o resultado da simulação, representado na Figura 5.14, que para os valores de $p \geq 0.41$ o método RR realiza menos transmissões que o

método FLD e neste mesmo valor de p encontra-se o ponto de transição entre os métodos FLD e RR.

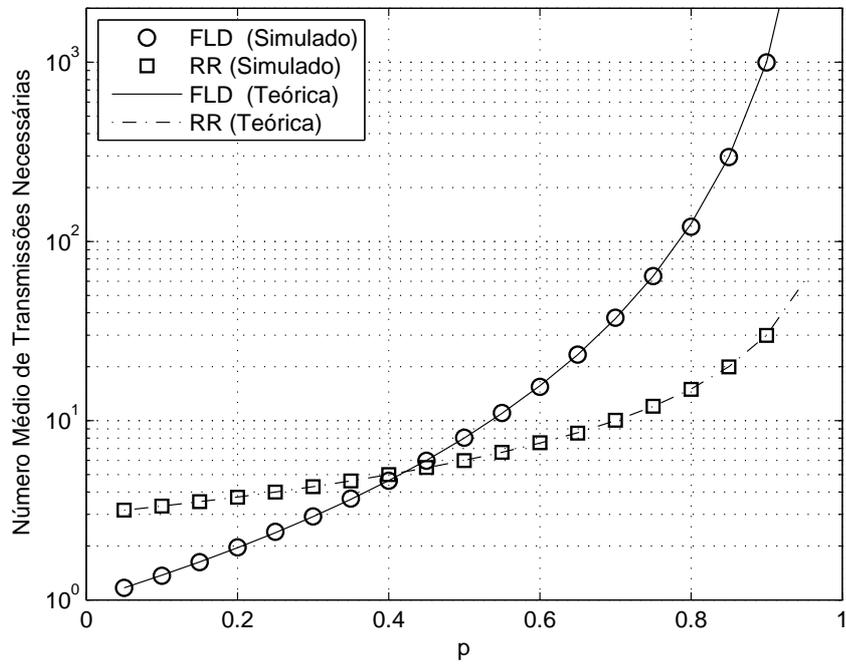


Figura 5.14: Comparação teórica entre os métodos de disseminação FLD e RR

Capítulo 6

Conclusão

As redes de sensores sem fio (RSSF) estão cada vez mais presentes no cotidiano das pessoas. A diversidade de aplicações que as RSSF viabilizam devem, em breve, torná-las rotineiras em residências, fábricas, escritórios, escolas dentre outros. A necessidade de novas estratégias e técnicas para otimizar este tipo de rede sem fio são iminentes em função deste crescimento de utilização e aplicações.

Um dos assuntos mais relevantes envolvendo as RSSF é a operação de disseminação de dados. A operação de distribuir a mesma informação para todos os nós de uma RSSF é denominado *disseminação de dados*. Tal processo é utilizado para diferentes fins neste tipo de rede e é de grande importância sob o ponto de vista operacional. Atualização dos equipamentos, manipulação de tabelas de roteamento ou, de modo geral, atualização de dados que devem estar presentes em todos os nós, são exemplos clássicos que justificam um processo de disseminação. Diversos trabalhos são propostos neste contexto para as RSSF e como resultado alguns protocolos para viabilizar uma disseminação são apresentados.

Realizamos neste trabalho primeiramente um estudo comparativo das estratégias de disseminação FLD (*flooding determinístico*) e FBCast incluindo em ambos os métodos conceitos de códigos fontanais. Em um segundo momento estendemos tais estratégias combinando-as com técnicas de nós estratégicos *relays*, a partir da utilização do algoritmo de Kruskal, que denominamos MDST (*minimum dissemination spanning tree*). Por fim, propomos um novo método para o processo de disseminação intitulado RR (*replicação de relay*). A escolha da utilização de métodos baseados em códigos fontanais se deu pelo fato de que tal técnica não exige a necessidade de um canal de retorno para confirmação das mensagens transmitidas entre os nós. As RSSF possuem uma alta taxa de perda de pacotes em função do meio de propagação das informações, portanto, os protocolos que operam com confirmação de pacotes recebidos são proibitivos nestes ambientes. A ausência de um canal de retorno, viabilizada pelos códigos fontanais, diminui o tráfego de pacotes na rede e influencia diretamente na economia de energia dos nós, requisito fundamental para operação de um RSSF.

A idealização do cenário em grade (*grid*) e o comportamento regular encontrado nesta topologia permitiu a avaliação isolada dos métodos de disseminação. Nossa métrica avaliada é o número de pacotes transmitidos pelos nós. Entendemos que diminuindo o número de pacotes transmitidos obtemos economia de energia. Ao proporcionar economia de energia, por consequência, estendemos a operação da rede priorizando assim o objetivo da mesma. A partir da utilização da MDST tivemos como objetivo limitar o número de nós repetidores e então diminuir o tráfego da rede sem comprometer o processo disseminatório. Contudo, percebemos que

tal técnica é indicada apenas para altas densidades. Pontualmente em nosso estudo os melhores desempenhos dos métodos associados a MDST ocorreram para cenários de 20 e 25 pés, exceto no caso do FBCast no cenário 25 pés.

Por outro lado, os métodos que não utilizavam as técnicas de MDST, sobressaíram-se em cenários de baixa densidade, ou seja, cenários de 35 e 40 pés. A justificativa dá-se por conta de que a distância entre os nós aumenta o número de transmissões falhas, logo, a redundância de conexões para um mesmo nó otimiza a entrega das informações. Observou-se que o método FLD teve o melhor desempenho em relação aos demais avaliados para os cenários simulados de 30 e 35 pés. Em relação ao desempenho do método RR, obtivemos um melhor desempenho quando combinamos o mesmo com a MDST para os cenários de alta densidade, 20 e 25 pés, e baixa densidade, pontualmente, no cenário de 40 pés. O RR associado a técnicas de *relays* (RR+MDST) ao mesmo tempo que replica a fonte na rede também controla o reenvio dos pacotes através da MDST.

Estabelecemos neste estudo a MDST de forma centralizada. Apesar do uso desta técnica ser incompatível para RSSF sob o ponto de vista prático, identificamos que o uso deste tipo de técnica é aceitável principalmente em cenários densos. Não o descartando para cenários mais esparsos em função do comportamento pontual apresentado no cenário de 40 pés. De qualquer forma, a idéia de combinar a utilização de técnicas de MDST à métodos de disseminação demonstrou ser uma boa alternativa. Recomenda-se novos estudos para a elaboração desta técnica em uma versão distribuída.

Um método de disseminação adaptativo, onde este possa modificar sua metodologia de disseminação em função do cenário de rede, merece uma pesquisa mais detalhada. Além disso, uma continuidade deste trabalho pode explorar questões envolvendo outras métricas, considerar cenários maiores, aleatórios e até mesmo aplicações práticas.

Referências Bibliográficas

- [Akyildiz and Vuran, 2010] Akyildiz, I. and Vuran, M. C. (2010). *Wireless Sensor Networks*. John Wiley & Sons, Inc., New York, NY, USA.
- [Akyildiz et al., 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):pg 393–422.
- [Birman et al., 1999] Birman, K. P., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., and Minsky, Y. (1999). Bimodal multicast. *ACM Trans. Comput. Syst.*, 17:41–88.
- [Bischoff et al., 2009] Bischoff, R., Meyer, J., and Feltrin, G. (2009). Wireless Sensor Network Platforms. *Encyclopaedia of structural health monitoring*. Chichester: John Wiley & Sons, pages 1229–1238.
- [Casari et al., 2008] Casari, P., Rossi, M., and Zorzi, M. (2008). Fountain codes and their application to broadcasting in underwater networks: performance modeling and relevant tradeoffs. In *Proceedings of the third ACM international workshop on Underwater Networks*, pages 11–18. ACM.
- [Community, 2010a] Community (2010a). *TinyOS - Open Source BSD*, Disponível em: <http://www.tinyos.net/>. <<http://www.tinyos.net/>>.
- [Community, 2010b] Community (2010b). *TinyOS Simulator TOSSIM*, Disponível em: <http://docs.tinyos.net/index.php/TOSSIM>. <<http://docs.tinyos.net/index.php/TOSSIM>>.
- [Cormen, 2001] Cormen, T. (2001). *Introduction to algorithms*. The MIT press.
- [Crepaldi et al., 2007] Crepaldi, R., Harris III, A., Rossi, M., Zanca, G., and Zorzi, M. (2007). Fountain reprogramming protocol (FRP): a reliable data dissemination scheme for wireless sensor networks using fountain codes. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, page 390. ACM.
- [Dahlman et al., 2009] Dahlman, E., Parkvall, S., Bovik, A., Beming, P., Chou, P., Correia, L., Skold, J., Da Silva, E., van der Schaar, M., Bensch, A., et al. (2009). *Communications engineering desk reference*. Academic Pr.
- [Ganesan et al., 2002] Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., and Wicker, S. (2002). An empirical study of epidemic algorithms in large scale multihop wireless networks. *Intel Research Berkeley*.
- [Gangali, 2002] Gangali, M. (2002). *Getting started with Bluetooth*. Premier Press.

- [Gislason, 2008] Gislason, D. (2008). *Zigbee wireless networking*. Newnes.
- [Gutierrez et al., 2003] Gutierrez, J., Callaway, E., and Barrett, R. (2003). *Low-rate wireless personal area networks: enabling wireless sensors with IEEE 802.15. 4*. Institute of Electrical & Electronics Engineers (IEEE).
- [Hongpeng Zhu, 2008] Hongpeng Zhu, guangxia Li, Z. X. (2008). Advanced LT Codes in Satellite Data Broadcasting System. *IEEE*.
- [Hui and Culler, 2004] Hui, J. and Culler, D. (2004). The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 81–94. ACM.
- [Hyytia et al., 2007a] Hyytia, E., Tirronen, T., and Virtamo, J. (2007a). Optimal degree distribution for LT codes with small message length. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2576–2580. IEEE.
- [Hyytia et al., 2007b] Hyytia, E., Tirronen, T., and Virtamo, J. (2007b). Optimal degree distribution for LT codes with small message length. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2576–2580. IEEE.
- [Ilyas and Mahgoub, 2006] Ilyas, M. and Mahgoub, I. (2006). *Smart dust: sensor network applications, architecture, and design*. CRC.
- [Jin et al., 2007] Jin, K., McEachen, J., and Singh, G. (2007). RF characteristics of Mica-Z wireless sensor network motes. In *Circuits and Systems, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on*, volume 2, pages 100–104. IEEE.
- [Kadi, 2009] Kadi, N. (2009). Optimized mpr-based flooding in wireless ad hoc network using network coding. In *Wireless Days, 2008. WD'08. 1st IFIP*, pages 1–5. IEEE.
- [Khisti,] Khisti, A. Tornado codes and luby transform codes.
- [Kumar et al., 2005] Kumar, R., Paul, A., and Ramachandran, U. (2005). Fountain broadcast for wireless networks. In *IEEE 2nd International Workshop on Network Sensing Systems (INSS). San Diego, CA*.
- [Kumar et al., 2006] Kumar, R., Paul, A., Ramachandran, U., and Kotz, D. (2006). On improving wireless broadcast reliability of sensor networks using erasure codes. *Mobile Ad-hoc and Sensor Networks*, pages 155–170.
- [Kurose and Ross, 2007] Kurose, J. and Ross, K. (2007). *Computer networking: A top-down approach*.
- [Leiner, 2005] Leiner, B. (2005). LDPC Codes—a brief Tutorial. *Stud. ID.: 53418L April*, 8.
- [Leon, 1999] Leon, S. (1999). *Algebra Linear com Aplicações, 4 edição*. LTC Editora, Rio de Janeiro.
- [Levis et al., 2009] Levis, P., Gay, D., and Books24x7, I. (2009). *TinyOS programming*. Cambridge University Press Cambridge.

- [Levis and Lee, 2003] Levis, P. and Lee, N. (2003). Tossim: A simulator for tinyos networks. *UC Berkeley, September*.
- [Levis et al., 2003] Levis, P., Lee, N., Welsh, M., and Culler, D. (2003). TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 126–137. ACM.
- [Levis et al., 2005] Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., et al. (2005). Tinyos: An operating system for sensor networks. *Ambient Intelligence*, pages 115–148.
- [Lim and Kim, 2001] Lim, H. and Kim, C. (2001). Flooding in wireless ad hoc networks. *Computer Communications*, 24(3-4):353–363.
- [Luby, 1998] Luby, M. (1998). Tornado codes: Practical erasure codes based on random irregular graphs. *Randomization and Approximation Techniques in Computer Science*, pages 171–171.
- [Luby, 2003] Luby, M. (2003). LT codes. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium*. IEEE.
- [MacKay, 2005] MacKay, D. (2005). Fountain codes. In *Communications, IEE Proceedings*, volume 152, pages 1062–1068. IET.
- [Mitzenmacher, 2005] Mitzenmacher, M. (2005). Digital fountains: A survey and look forward. In *Information Theory Workshop, 2004*. IEEE.
- [Otto et al., 2006] Otto, C., Milenkovic, A., Sanders, C., and Jovanov, E. (2006). System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1(4):307–326.
- [Rossi et al., 2008] Rossi, M., Zanca, G., Stabellini, L., Crepaldi, R., Harris, A., and Zorzi, M. (2008). SYNAPSE: A network reprogramming protocol for wireless sensor networks using fountain codes. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*, pages 188–196. IEEE.
- [Ruiz et al., 2003] Ruiz, L. B., Nogueira, J. M. S., and Loureiro, A. A. F. (2003). MANNA: a management architecture for wireless sensor networks. *IEEE Communications Magazine*, 41(2):116–125. ISSN 0163-6804.
- [Santamaria, 2001] Santamaria, A., editor (2001). *Wireless LAN Standards and Applications*. Artech House, Inc., Norwood, MA, USA, 1st edition.
- [Shokrollahi, 2006] Shokrollahi, A. (2006). Raptor codes. *Information Theory, IEEE Transactions on*, 52(6):2551–2567.
- [Tanenbaum, 2003] Tanenbaum, A. (2003). *Redes de computadores*. Rio de Janeiro: Editora Campus.
- [Telecomunicazioni, 2007] Telecomunicazioni, D. G. (2007). *SYNAPSEs TinyOS v2 Open Source Code Distribution, Disponível em: <http://telecom.dei.unipd.it/pages/read/59/>. <<http://telecom.dei.unipd.it/pages/read/59/>>*.

- [Toh, 2001] Toh, C. (2001). *Ad Hoc Wireless Networks: Protocols and Systems*. Prentice Hall PTR Upper Saddle River, NJ, USA.
- [Wicker, 1995] Wicker, S. (1995). *Error control systems for digital communication and storage*, volume 19. Prentice Hall New Jersey.
- [Williams and Camp, 2002] Williams, B. and Camp, T. (2002). Comparison of broadcasting techniques for mobile ad hoc networks. pages 194–205.
- [Zh and Sarikaya, 2007] Zh, X. and Sarikaya, B. (2007). Code Dissemination in Sensor Networks with MDeluge. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society*, volume 2, pages 661–666. IEEE.
- [Zheng and Sarikaya, 2009] Zheng, X. and Sarikaya, B. (2009). Task dissemination with multicast deluge in sensor networks. *Wireless Communications, IEEE Transactions on*, 8(5):2726–2734.

Apêndice A

Qualcomm e Digital Fountain Inc.

Os trabalhos relacionados com o tema códigos fontanais, na maioria dos casos, são pesquisas e simulações da viabilidade da técnica em diferentes aplicações. Os principais projetos práticos relacionados ao assunto são pertencentes a uma empresa e protegidos por patentes.

A.1 Projetos

A empresa Qualcomm, adquiriu os direitos da antiga Digital Fountain e propõe quatro diferentes tipos de produtos. O primeiro deles (IPTV) tem por objetivo fazer transmissões de conteúdo televisivo, com garantia de qualidade. A segunda solução (Mobile), basicamente da mesma forma, provê qualidade de distribuição de conteúdo para redes de comunicação celular. A terceira solução (DF Splash), não diferentemente em essência das anteriores, provê serviço de transmissão com qualidade para *streamming* de vídeos digitais. Por fim, o quarto produto é vinculado a defesa militar (Government and Defense). Neste caso, o atrativo do produto é prover serviço de qualidade na comunicação, justificado pelo fato de que em territórios hostis haja, provavelmente, escassez de comunicação de boa qualidade (interferência, ruído, banda limitada, latência), podendo comprometer a comunicação.

A empresa foi fundada pelos criadores diretos dos métodos de códigos fontanais Raptor Codes, Dr. Michael Luby e Dr. Amin Shokrollahi.

Como sub produtos, a empresa provê exatamente os mesmos serviços oferecidos em suas soluções, porém, para ser adicionado a clientes que, eventualmente, já trabalhem neste tipo de mercado.

Apêndice B

Wireless Sensor Networks Research Group

O portal Wireless Sensor Networks Research Group em <http://www.sensor-networks.org/> traz informações diversas sobre o assunto envolvendo redes de sensores sem fio.

B.1 Projetos

É possível encontrar diversos artigos, projetos e manual de produtos relacionados ao tema. Os membros são pesquisadores e desenvolvedores de projetos relacionados as redes de sensores sem fio.

As linhas de pesquisa dividem-se em: Embarcados, Gerência de energia, Redes GPS Mesh, Equipamentos, Gerenciador de Plataforma, Protocolos de roteamento, Integração de Sensores e Sistema e Comunicação.