

Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Diego Véras de Queiroz

Simulação Realista de Redes de Sensores sem Fio Industriais

João Pessoa, Paraíba, Brasil

©Diego Véras de Queiroz, 29 de Fevereiro de 2016

Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Simulação Realista de Redes de Sensores sem Fio Industriais

Diego Véras de Queiroz

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sinais, Sistemas Digitais e Gráficos

Iguatemi Eduardo da Fonseca
(Orientador)

João Pessoa, Paraíba, Brasil

©Diego Véras de Queiroz, 29 de Fevereiro de 2016

Q3s Queiroz, Diego Vêras de.
Simulação realista de redes de sensores sem fio
industriais/ Diego Vêras de Queiroz.- João Pessoa, 2016.
102f. : il.
Orientador: Iguatemi Eduardo da Fonseca
Dissertação (Mestrado) - UFPB/CI
1. Informática. 2. Redes de sensores sem fio. 3. Simulador.
4. Modelo de canal. 5. Interferência. 6. Multipercurso.

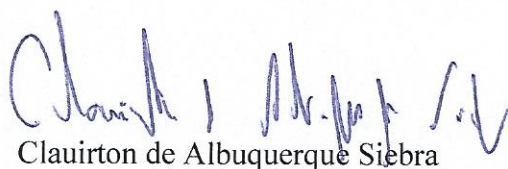
UFPB/BC

CDU: 004(043)

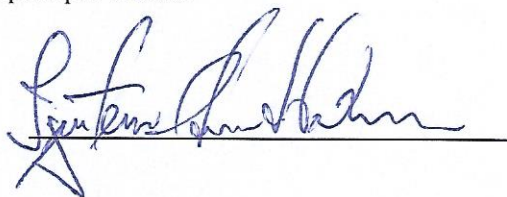
Ata da Sessão Pública de Defesa de Dissertação de
Mestrado de DIEGO VERAS DE QUEIROZ,
candidato ao título de Mestre em Informática na
Área de Sistemas de Computação, realizada em 29
de fevereiro de 2016.

Ao vigésimo nono dia do mês de fevereiro, do ano de dois mil e dezesseis, às dez horas, no Centro de Informática - Universidade Federal da Paraíba (unidade Mangabeira), reuniram-se os membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Diego Veras de Queiroz, vinculado a esta Universidade sob a matrícula 2014108618, candidato ao grau de Mestre em Informática, na área de "*Sistemas de Computação*", na linha de pesquisa "*Sinais, Sistemas Digitais e Gráficos*", do Programa de Pós-Graduação em Informática, da Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores: Iguatemy Eduardo de Fonseca, Orientador e Presidente da Banca, Fernando Menezes Matos (PPGI-UFPB), Examinador Interno, e Waslon Terlizzie Araújo Lopes, Examinador Externo ao Programa. Dando início aos trabalhos, o professor Iguatemy Eduardo de Fonseca, cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse, oralmente, a exposição do trabalho de dissertação intitulado "*Simulação Realista de Redes de Sensores sem Fio Industriais*". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "*aprovado*". Assim sendo, eu, Claurton de Albuquerque Siebra, Coordenador do Programa de Pós-Graduação em Informática - PPGI, lavrei a presente ata que vai assinada por mim e pelos membros da Banca Examinadora.

João Pessoa, 29 de fevereiro de 2016.


Claurton de Albuquerque Siebra

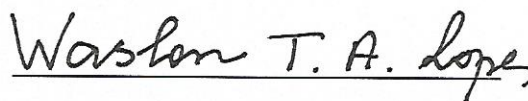
Prof Dr Iguatemy Eduardo de Fonseca
Orientador (PPGI-UFPB)



Prof Dr Fernando Menezes Matos
Examinador Interno (PPGI-UFPB)



Prof Dr Waslon Terlizzie Araújo Lopes
Examinador Externo ao Programa(UFPB)



Resumo

Uma Rede de Sensores sem Fio (RSSF) pode ser caracterizada pela utilização de nós sensores em variados ambientes para as mais diversas aplicações, por exemplo, para monitoramento e controle em ambiente industrial. O uso de RSSFs para implementar aplicações industriais precisa lidar com problemas típicos das redes sem fio, como alta atenuação e obstruções. Nesses ambientes, diferenças nas características dos canais podem ser observadas dado o seu perfil de multipercurso. Mudanças na topologia também podem provocar mudanças nas características dos canais ao longo do tempo, podendo causar diferença no valor médio da potência recebida. Uma alternativa para lidar com isso e com a variação da qualidade dos canais é o desenvolvimento de protocolos que utilizam múltiplos canais. Seu uso permite, por exemplo, alcançar uma melhor utilização do canal, aumentando a capacidade de transmissão da rede. No entanto, existe uma dificuldade em avaliar e comparar as diferentes abordagens devido à carência de simuladores que apresentem modelos mais precisos. Para avaliar os reais benefícios desses protocolos deve-se levar em consideração um modelo de propagação mais próximo da realidade. Esse trabalho apresenta um modelo de simulação que captura os efeitos de desvanecimento em larga escala, atenuação em pequena escala, sombreamento log-normal, as características não-estacionárias do canal em longos períodos de tempo, e as características distintas dos diferentes canais do padrão IEEE 802.15.4. O modelo foi então integrado ao simulador de eventos discretos de código aberto Castalia e testado, comparando-o com o modelo estacionário padrão do próprio simulador. A partir da integração, o protocolo *Tree-based Multi-Channel Protocol* (TMCP) foi implementado e avaliado utilizando o modelo proposto. Com os experimentos foi possível identificar uma ótima precisão do modelo e do protocolo TMCP.

Palavras-chave: Redes de Sensores sem Fio, Simulador, Modelo de Canal, Indústria, Interferência, Multipercurso

Abstract

A Wireless Sensor Network (WSN) can be characterized by the use of sensor nodes in varied environments for many different applications, *e.g.*, monitoring and control systems applications in industrial environments. The use of WSNs in industrial applications must deal with typical problems of wireless networks, such as high attenuation and obstructions. In these environments, different characteristics of the channels can be observed due to its multipath profile. Changes in the topology may also cause changes in the channel characteristics as the time goes on, and it may cause differences in the average value of the received power. An alternative to deal with it and with the variation of channel quality is the development of protocols that use multiple channels. Its use allows, *e.g.*, to achieve better use of the channel, increasing the network transmission capacity. However, there is a difficulty in evaluating and comparing different approaches due to the lack of simulators which have more accurate models. To assess the real benefits of these protocols, one should take into account a more realistic propagation model. This work presents a realistic simulation model that captures the effects of fading on a large and small scale, log-normal shadowing, the non-stationary characteristics of the channel for a long time, and the different characteristics of the different IEEE 802.15.4 standard channels. The model was then built into the open-source discrete event simulator Castalia and tested, comparing it with the default stationary model of the simulator. After integration, the protocol *Tree-based Multi-Channel Protocol* (TMCP) was evaluated using the proposed model. In the experiments it was possible to identify good accuracy in both the model and the TMCP protocol.

Keywords: Wireless Sensor Networks, Simulator, Channel model, Industry, Interference, Multipath

Agradecimentos

Agradecimentos a Deus pelo dom da vida e pelas oportunidades;

Aos meus pais, Queiroz e Creusa, pelos ensinamentos de vida que nenhum estudo poderia me dar;

À minha esposa, Elane, pela paciência e amor dedicados a mim durante todo o tempo em que estamos juntos;

A meu sogros, irmãos e cunhados pela compreensão e ajuda;

Ao professor Dr. Iguatemi E. Fonseca, pela orientação e incentivos na construção desse trabalho e pelo direcionamento na continuação dos estudos;

Ao professor doutorando Ruan Delgado Gomes (IFPB e UFCG) pelo trabalho em conjunto dentro do laboratório e em campo, pela amizade, co-orientação e incentivos, e por ter sempre me mantido informado sobre as oportunidades de doutorado;

Aos professores Dr. Alisson Brito e Dr. Eudisley Anjos pela disponibilidade em também me ajudarem na busca do meu futuro;

Ao professor Dr. Marcelo Sampaio de Alencar (UFCG) pela participação e orientação em alguns artigos submetidos e publicados, inclusive na minha apresentação no simpósio, e por ter me informado, mesmo que indiretamente através do prof. Ruan D. Gomes, sobre a vaga de doutorado para a qual me candidatei e que, felizmente, fui aceito;

Ao STI e à Gerência de Segurança por ter me permitido, através da adaptação de horários, trabalhar e ao mesmo tempo cursar o mestrado;

Aos colegas e amigos do STI pelo incentivo.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos e Contribuições da Dissertação	3
1.2.1	Objetivo Geral	3
1.2.2	Objetivos Específicos	4
1.2.3	Contribuições da Dissertação	4
1.3	Metodologia	4
1.4	Estrutura da Dissertação	5
2	Fundamentação Teórica	6
2.1	Redes de Sensores sem Fio	6
2.1.1	Interferência nas Comunicações	9
2.1.2	RSSFs Industriais	10
2.2	Padrão IEEE 802.15.4	14
2.3	Protocolos Multicanais	20
2.4	Simuladores de Rede	22
2.4.1	Castalia	26
2.4.2	Comparação dos Simuladores	32
2.4.3	Trabalhos Utilizando o Castalia	33
3	Metodologia de Simulação com o Castalia	35
3.1	Modelo de Simulação para o Desvanecimento em Longos Períodos de Tempo	36
3.1.1	Integração do modelo com o Castalia	39
3.2	Protocolo de múltiplos canais TMCP	43

4	Resultados	50
4.1	Perfil não-estacionário do canal	50
4.2	Simulação com o protocolo de múltiplos canais TMCP	52
4.2.1	Definição do experimento de simulação	53
4.3	Simulações numéricas	54
4.3.1	Plano de Simulação	54
4.3.2	Execução	56
4.3.3	Análise das Simulações	56
5	Conclusões e Trabalhos Futuros	65
5.0.1	Publicações e Submissões	67
	Referências Bibliográficas	75
A	Código em C++ do protocolo de múltiplos canais TMCP	76
B	Tabelas de localização dos nós nos ambientes, incluindo identificação dos nós folhas e respectivos pais	83

Lista de Símbolos

BAN : *Body Area Network (Rede de Área Corporal)*

CAD : *Conversor Analógico-Digital*

CAP : *Contention Access Period (Período de Contenção de Acesso)*

CCA : *Clear Channel Assessment (Avaliação de Canal Livre)*

CFP : *Contention Free Period (Período Livre de Contenção)*

CSMA/CA : *Carrier Sense Multiple Access with Collision Avoidance (Acesso Múltiplo com Verificação de Portadora com Prevenção de Colisão)*

DSSS : *Direct Sequence Spread Spectrum (Espalhamento Espectral por Frequência Direta)*

FFD : *Full Function Devices (Dispositivos com Função Completa)*

GTS : *Guaranteed Time Slot (Garantia de Tempo por Exposição)*

IDE : *Integrated Development Environment (Ambiente de Desenvolvimento Integrado)*

IEEE : *Instituto de Engenheiros Eletricistas e Eletrônicos*

LLC : *Logical Link Control (Controle do Enlace Lógico)*

LQI : *Link Quality Indication (Indicador de Qualidade do Enlace)*

LR-WPAN : *Low-Rate Wireless Personal Area Network (Rede Pessoal sem Fio com Baixa Taxa de Transmissão)*

MAC : *Media Access Control (Controle de Acesso ao Meio)*

NED : *Network Description (Protocolo de Descrição de Rede)*

NS3 : *Network Simulador 3*

OMNeT++ : *Objective Modular Network Testbed in C++*

OSI : *Open Systems Interconnection (Interconexão de Sistemas Abertos)*

PHY : *Physical layer (Camada Física)*

RF : *Radio Frequency (Frequência de Rádio)*

RFD : *Reduced Function Devices (Dispositivos com Função Reduzida)*

RSSF : *Rede de Sensores sem Fio*

RSSI : *Received Signal Strength Indication (Indicador de Força do Sinal Recebido)*

SSCS : *Service Specific Convergence Sublayer (Sub-camada de Convergência)*

TMCP : *Tree-based Multi-Channel Protocol (Protocolo de Múltiplos Canais Baseado em Árvores)*

TOSSIM : *TinyOS Simulator*

TPP : *Taxa de Perda de Pacotes*

TRP : *Taxa de Recepção de Pacotes*

WPAN : *Wireless Personal Area Network (Rede Sem Fio Pessoal)*

Lista de Figuras

2.1	Sensores capturando qualidade do ar.	7
2.2	Componentes básicos do nó.	8
2.3	Desvanecimento em larga escala e atenuação em pequena escala.	11
2.4	Potência de recepção em experimento realizado por 20 horas.	12
2.5	Padrões IEEE 802.15.4 e 802.11 compartilham a mesma banda.	15
2.6	Exemplos de Topologias.	16
2.7	Arquitetura do padrão IEEE 802.15.4.	17
2.8	Frequências de operação e bandas no IEEE 802.15.4 na camada física. . . .	18
2.9	Camada MAC do IEEE 802.15.4.	18
2.10	Estrutura do Superframe com períodos ativo e inativo.	19
2.11	Lista de módulos implementados pelo simulador NS3.	24
2.12	Estrutura do Omnet++.	25
2.13	Os módulos e conexões no Castalia.	26
2.14	O módulo composto do nó (estrutura).	27
2.15	Lista de módulos implementados pelo simulador Castalia.	27
2.16	Modelos de Canal Rice e Rayleigh.	31
3.1	Processo de Markov de dois estados para modelar canal não-estacionário. .	37
3.2	RSSI simulado com o modelo proposto utilizando a biblioteca IT++. . . .	39
3.3	Potência de Recepção Média simulada com o Castalia.	40
3.4	Taxa de Recepção de Pacotes.	41
3.5	Dados de entrada para o algoritmo TMCP e resultados.	46
4.1	Disposição dos nós dentro da indústria.	51

4.2	Comportamento não-estacionário do canal em experimento realizado na indústria.	51
4.3	Experimento que mostra canais descorrelacionados.	52
4.4	Experimento de simulação em área de 200x200 metros com 8 canais.	55
4.5	Potência de recepção <i>versus</i> Pacotes recebidos (MARKOV) 200 m.	58
4.6	Potência de recepção <i>versus</i> Pacotes recebidos (PADRÃO) 200 m.	59
4.7	Potência de recepção <i>versus</i> Pacotes recebidos (MARKOV) 400 m.	60
4.8	Potência de recepção <i>versus</i> Pacotes recebidos (PADRÃO) 400 m.	61
4.9	Valores médios da Taxa de Recepção de Pacotes para cada sorvedouro em todos os cenários.	64

Lista de Tabelas

2.1	Comparação de dois dos principais simuladores.	32
3.1	Parâmetros utilizados na simulação.	40
4.1	Ambientes simulados utilizando modelo não-estacionário e o protocolo TMCP	54
B.1	Localização dos nós no ambiente com área de 200x200 m com múltiplos canais.	84
B.2	Continuação da Tab. B.1.	85
B.3	Nós folhas em cada ambiente (200x200 m) e seus respectivos pais (roteadores).	85
B.4	Nós folhas em cada ambiente (400x400 m) e seus respectivos pais (roteadores).	86
B.5	Localização dos nós no ambiente com área de 400x400 m com múltiplos canais.	87
B.6	Continuação da Tab. B.5.	88
B.7	Continuação da Tab. B.5.	89
B.8	Continuação da Tab. B.5.	90

Lista de Códigos Fonte

2.1	Abstração dos detalhes de roteamento.	28
2.2	Instanciação dos parâmetros do Radio.	29
2.3	Valores padrão do modelo do sombreamento log-normal no Castalia.	30
3.1	Cálculo do sombreamento log-normal no módulo <i>WirelessChannel.cc</i>	41
3.2	Função que atualiza as características do canal com probabilidade p de mudança.	42
3.3	Chamada da função <i>updateChannel</i>	43
3.4	Resultado da execução do algoritmo TMCP com dados compilados.	47
4.1	Exemplo de execução da simulação pelo Castalia.	56
A.1	Código do protocolo multicanal TMCP	76

Capítulo 1

Introdução

1.1 Motivação

Uma Rede de Sensores sem Fio (RSSF) pode ser caracterizada pela utilização de uma grande quantidade de nós sensores capazes de se comunicarem entre si em variados ambientes para as mais diversas aplicações. Quando uma RSSF é estruturada, os nós são implantados de uma forma planejada, com localização pré-definida. Quando a rede é não-estruturada, a comunicação entre os nós é feita de modo *ad-hoc* e exige auto-organização, com um nó transmitindo os valores do sensoramento ao mais próximo, até que a informação chegue ao(s) nó(s) principal(is) (nós sorvedouros - *sink nodes*).

O uso de RSSF para implementar sistemas de monitoramento e controle em ambientes industriais possui algumas vantagens em relação ao uso de redes cabeadas. Em sistemas baseados nas redes cabeadas, o processo de instalação de cabos e sensores possui geralmente um custo muito mais alto do que os próprios sensores [Bin Lu e Gungor 2009]. Além disso, o cabeamento possui pouca flexibilidade, o que dificulta a instalação e manutenção da rede. Uma alternativa para construção de sistemas de monitoramento industrial de custo reduzido é a utilização de redes sem fio, que apresentam muitas vantagens, incluindo maior flexibilidade, baixo custo e facilidade de instalação e manutenção [Gungor e Hancke 2009]. Mais especificamente, as RSSFs apresentam ainda como vantagens a capacidade de auto-organização, o processamento local e surgem como uma plataforma promissora para implementar sistemas *online* e remotos de monitoramento e controle em ambientes industriais.

Apesar dessas vantagens, é preciso lidar com problemas típicos das redes sem fio, como

alta atenuação devido à presença de muitos objetos metálicos e obstruções [Tanghe et al. 2008]. Os sistemas de monitoramento industrial usualmente requerem alta confiabilidade, de modo a monitorar parâmetros críticos como vibração, temperatura, pressão e eficiência [Delgado Gomes et al. 2013], mas o meio de comunicação entre os nós não é confiável por causa da interferência e do ruído no canal utilizado.

Devido às limitações das RSSFs, principalmente com relação à baixa taxa de transmissão e à falta de confiabilidade, a implementação dessa estrutura se torna ainda mais desafiadora, especialmente em ambientes industriais. Em tais ambientes, diferenças nas características dos canais podem ser observadas devido ao perfil de multipercurso, uma vez que os canais são descorrelacionados em frequência e o impacto do multipercurso é diferente em canais diferentes [Gomes et al. 2014]. Além disso, mudanças na topologia do ambiente (exemplo, a movimentação de uma grande estrutura metálica) podem provocar mudanças nas características dos canais ao longo do tempo, o que pode causar diferença no valor médio da potência recebida, mesmo o transmissor e receptor permanecendo estáticos. Dessa forma, o canal sem fio no ambiente industrial pode permanecer durante várias horas apresentando as mesmas características e após esse período pode ocorrer uma mudança brusca nas características do canal [Agrawal et al. 2014].

Uma alternativa para lidar com os problemas da comunicação sem fio e com a variação da qualidade dos canais no decorrer do tempo em ambientes industriais, é o desenvolvimento de protocolos que utilizam múltiplos canais. O uso de protocolos multicanais permite alcançar uma melhor utilização do canal, aumentando a capacidade de transmissão da rede por meio da transmissão em simultâneo usando vários canais diferentes [Incel 2011].

Algumas soluções utilizam múltiplos canais simultaneamente com múltiplos transceptores nos nós ou com a transmissão por diferentes canais em diversos locais da rede [Huang et al. 2013]. Outras soluções utilizam salto em frequência, como os protocolos MAC dos padrões WirelessHART e ISA100 [Silva et al. 2012]. Esses mecanismos são tipicamente implementados na camada de acesso ao meio. Alguns trabalhos focam no uso de mecanismos de alocação dinâmica de canal [Gomes et al. 2014], em que a RSSF troca de canal se são identificados problemas causados pela presença de fontes de interferência ou por problemas devido à atenuação por multipercurso.

Vários protocolos multicanais foram propostos na literatura [Incel 2011], e em sua mai-

oria visam diminuir a quantidade de colisões por meio da associação de canais diferentes para nós que podem interferir entre si ou por meio do escalonamento conjunto no tempo e na frequência. No entanto, existe uma dificuldade em avaliar e comparar as diferentes abordagens devido à falta de um *testbed* (laboratório de testes) comum e à carência de simuladores que apresentem modelos acurados para simular ou emular características dos múltiplos canais simultaneamente. Muitos protocolos multicanais são implementados apenas em simuladores com suposições simplistas [Incel 2011].

Para avaliar os reais benefícios de protocolos multicanais deve-se levar em consideração um modelo realista de propagação e também o atraso relativo à troca de canais [Soua e Minet 2015]. O uso de simuladores com modelos mais acurados permite avaliar o desempenho de novos protocolos para RSSFs, com múltiplos nós e em diferentes cenários, sem a necessidade de implementação física da rede. É possível prever eventuais problemas na sua estrutura sem precisar vivenciá-los, o que leva a uma redução dos seus custos e uma otimização da rede.

A caracterização de ambientes industriais, verificando as fontes de interferência e as características de propagação, é um passo importante para o desenvolvimento de novas aplicações que fazem uso de redes sem fio e para o aprimoramento das atuais tecnologias [Coll 2012]. Dessa forma, estudos experimentais para identificação dessas fontes e a análise de desempenho dos rádios operando no ambiente industrial podem gerar informações valiosas para o desenvolvimento de novas técnicas e protocolos para redes de sensores sem fio industriais.

1.2 Objetivos e Contribuições da Dissertação

1.2.1 Objetivo Geral

O objetivo geral desse trabalho é implementar um modelo mais adequado de simulação que capture os efeitos de desvanecimento em larga escala, atenuação em pequena escala, sombreamento log-normal, as características não estacionárias do canal em longos períodos de tempo, e as características distintas dos diferentes canais do padrão IEEE 802.15.4.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

- Utilizar o simulador de eventos discretos Castalia para estudar os efeitos causados pelo ambiente industrial na comunicação entre nós dentro de uma RSSF. A partir desse estudo, é possível implementar aplicações que podem monitorar importantes parâmetros e controlar os processos industriais;
- Integrar o modelo de simulação proposto com o Castalia e verificar a ocorrência de alterações no perfil multipercurso do ambiente simulado;
- Avaliar o protocolo multicanal *Tree-based Multi-Channel Protocol* (TMCP) [Wu et al. 2008], ou Protocolo de Múltiplos Canais Baseado em Árvores, utilizado para aplicações de coleta de dados, que aloca canais para árvores disjuntas e explora transmissões paralelas entre elas.

1.2.3 Contribuições da Dissertação

Entre as contribuições desse trabalho estão o desenvolvimento de um modelo de canal que simula os efeitos causados pelo ambiente industrial na comunicação sem fio, a utilização da biblioteca IT++ para gerar valores aleatórios com a distribuição de Rice, os estudos experimentais do protocolo de múltiplos canais TMCP, a comparação do modelo proposto com o modelo estacionário padrão do Castalia, e a investigação da relação entre a taxa de recepção de pacotes e a potência de recepção pelo nó sorvedouro.

1.3 Metodologia

Com o objetivo de implementar um modelo mais realista de simulação que pudesse capturar os efeitos e problemas da comunicação sem fio no ambiente industrial, inicialmente foram realizados experimentos na indústria com três nós comunicando-se entre si, inseridos em posições estratégicas, um com visada direta e outro sem visada direta ao nó sorvedouro (*sink node*), transmitindo dados em vários canais diferentes. Com esse experimento foi possível visualizar os efeitos causados pelo ambiente industrial na comunicação em cada canal.

Em seguida aos experimentos, foi realizada uma revisão bibliográfica sobre os principais trabalhos publicados nos meios de divulgação de maior relevância da área, focando a pesquisa na simulação de RSSFs para ambientes industriais, e nas características desse tipo de ambiente.

Foram analisadas as vantagens e desvantagens de dois dos principais simuladores de redes existentes, o Castalia e o NS3, e, baseando-se no nível de precisão específico para simulação de RSSFs que cada um oferecia, escolheu-se o simulador Castalia. A partir daí, foi feito um estudo sobre sua estrutura, implementação, seus módulos e relacionamentos.

Utilizando esses conhecimentos e, para que os objetivos gerais e específicos fossem alcançados, um módulo do Castalia foi adaptado para que permitisse a captura dos efeitos de desvanecimento do canal por um longo período de tempo, no qual é identificada sua característica não-estacionária.

Concluída essa etapa, foi criada uma arquitetura para simulação de protocolos multicanais utilizando o protocolo TMCP no simulador, para permitir que cada um dos 16 canais do padrão IEEE 802.15.4 trabalhasse com um perfil não-estacionário diferente. O modelo proposto e o protocolo multicanal foram analisados e constatados possuírem boa precisão.

1.4 Estrutura da Dissertação

A dissertação está organizada inicialmente com a introdução ao problema e à proposta da dissertação no Capítulo 1. Em seguida são feitas a fundamentação teórica, revisão da literatura, incluindo a descrição do estado da arte no Capítulo 2. A apresentação da metodologia de simulação utilizando o Castalia é feita no Capítulo 3. Os resultados e análises obtidos no estudo de caso prático são mostrados no Capítulo 4. Nesse capítulo é ressaltada a contribuição da metodologia para o estado da arte ao apresentar benefícios que o projeto de RSSFs industriais pode obter ao utilizar o modelo desenvolvido para o simulador. As conclusões e trabalhos futuros são apresentados no Capítulo 5.

Como anexos ao trabalho estão código do protocolo multicanal TMCP no Apêndice A, e as tabelas de localização dos nós nos experimentos de simulação com o modelo proposto, utilizando o protocolo TMCP no Apêndice B.

Capítulo 2

Fundamentação Teórica

Nesse capítulo são apresentados conceitos básicos sobre RSSF e sua utilização na indústria, juntamente com as características do padrão IEEE 802.15.4. Também são apresentados os protocolos multicanais existentes e aquele que será utilizado nesse trabalho. Por fim são apresentados os conceitos de simuladores de redes, dando ênfase ao simulador de eventos discretos Castalia.

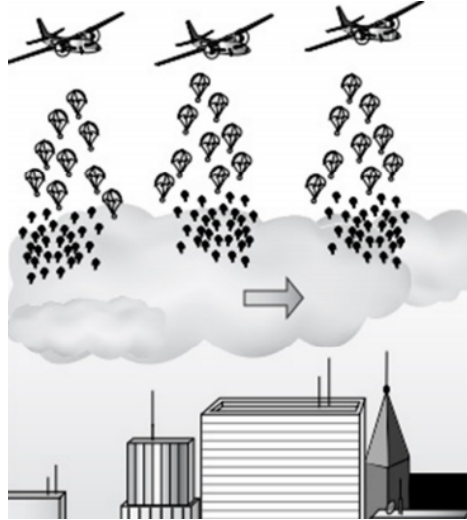
2.1 Redes de Sensores sem Fio

As RSSFs são formadas por nós equipados com sensores ou atuadores com capacidade de comunicação por radiofrequência. Alguns nós têm a capacidade de processamento, possibilitando uma melhor utilização do canal de comunicação [Lima-Filho et al. 2012]. Esses nós sensores podem ser inseridos no ambiente a ser analisado ou em suas proximidades de acordo com alguma estratégia de implantação, ou de forma aleatória como, por exemplo, quando um avião distribui os nós sensores em áreas de difícil acesso, conforme ilustrado na Fig. 2.1, adaptada de [Zhao e Guibas 2004]. Seu principal objetivo é monitorar o ambiente ou localizar objetos.

Os nós sensores consomem pouca energia e possuem capacidade de processamento das informações e de comunicação a curtas distâncias. Um nó possui quatro componentes básicos de hardware [Karl e Willig 2007]:

- Microcontrolador: processa os dados importantes para a rede, ou seja, coleta os dados dos sensores, armazena ou transmite as informações (para outros nós ou para o nó

Figura 2.1: Sensores capturando qualidade do ar.



sorvedouro);

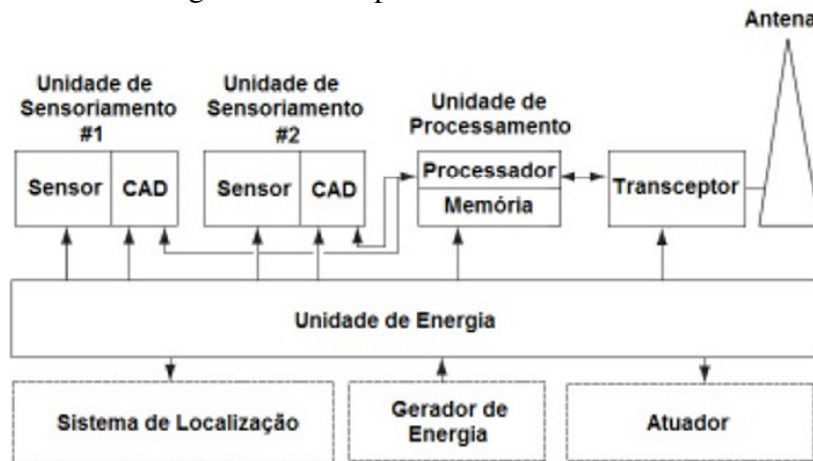
- Memória: armazena programas e dados coletados. Podem-se utilizar memórias RAM, ROM, EEPROM ou *flash* [Carvalho et al. 2012];
- Sensores e atuadores: realizam o sensoriamento e o controle físico dos parâmetros, respectivamente;
- Fonte de energia: representada, em geral, por uma bateria (mais recentemente, tem-se utilizado células fotovoltaicas que auxiliam na redução do seu consumo). Roteadores ou coordenadores às vezes dispõem de fonte de alimentação contínua devido à necessidade de funcionamento contínuo na rede.

Os componentes básicos de um nó sensor estão ilustrados na Fig. 2.2, adaptada de [Sousa e Lopes 2011]. É possível notar que um nó pode conter um ou mais sensores que monitoram diferentes variáveis, e para cada sensor há um conversor analógico-digital (CAD) que transforma os dados capturados (analógicos) em digitais.

Em algumas implementações é possível ainda adicionar um sexto componente que gerencia um sistema de geoposicionamento. Ele é útil nos casos em que seja necessário identificar a localização física do evento em lugares abertos.

Na RSSF os dados capturados do ambiente são transmitidos através de múltiplos saltos, retransmitidos por nós intermediários ao nó sorvedouro, que se conecta a outras redes através de um *gateway*.

Figura 2.2: Componentes básicos do nó.



Desafios das RSSFs

As restrições de um projeto de RSSF dependem geralmente da aplicação e se baseiam no ambiente que será monitorado. Os principais desafios ao se estabelecer uma RSSF são [Sousa e Lopes 2011] [Akyildiz et al. 2002]:

- Tolerância a falhas: alguns sensores podem falhar ou ser bloqueados por falta de energia, sofrer danos físicos ou interferências ambientais. Essas falhas não podem interferir no funcionamento geral da rede. Tolerância a falhas é a capacidade de manter a rede em funcionamento caso alguns desses problemas aconteçam;
- Escalabilidade: o número de sensores utilizados irá depender da aplicação, da situação a ser medida, que pode variar de centenas, milhares ou até milhões de sensores. A rede deve ser capaz de lidar com a inclusão de novos sensores ou se alguns pararem de funcionar;
- Custo: o custo de um único nó sensor é um fator limitante para que o projeto de uma RSSF seja viável, especialmente em grandes redes;
- Limitações de hardware: um nó sensor é composto basicamente pelos cinco componentes citados. Ao inserir esses componentes, é necessário atender a algumas exigências como elevada confiabilidade, baixo consumo de energia e custo e tamanho reduzidos;

- Consumo de energia: o sensor, sendo um dispositivo microeletrônico, deve ser equipado com uma fonte de energia limitada (menos que 0,5 Ah e 1,2 V). Em uma rede de sensores de múltiplos saltos, cada nó executa duas tarefas, uma para capturar a informação e a outra para rotear aquelas informações, podendo gerar consumo de energia além da capacidade da bateria;
- Ambiente: os sensores devem ser capazes de trabalhar nos mais variados ambientes, como no fundo dos oceanos, em equipamentos da indústria, em plantações, oceanos, em áreas altamente insalubres, entre outros;
- A sincronização e localização: há vários estudos nesse campo [Oliveira 2008] [Tabish Niaz, Ahmad Khan e Shafi 2011] e o uso de dispositivos GPS integrados aos nós sensores tem se tornado comum em aplicações de localização e rastreamento;
- Meio de transmissão: a maioria dos sensores atuais utiliza radiofrequência como meio de transmissão. Outras alternativas de comunicação são as comunicações ópticas e utilizando ultrassom [Karl e Willig 2007].

Os principais requisitos de uma RSSF são baixa latência, otimização para baixo consumo de energia, possibilidade de implementação de redes com elevado número de dispositivos e baixa complexidade dos nós de rede. O padrão *ZigBee* tem sido considerado como um padrão que está de acordo com as necessidades da comunicação sem fio entre dispositivos de controle e sensores.

2.1.1 Interferência nas Comunicações

Devido ao aumento no número de aplicações que utilizam redes sem fio, o espectro disponível para comunicação tende a ficar congestionado, o que aumenta a interferência e reduz a qualidade nas transmissões.

Em [Delgado Gomes et al. 2012] foi realizado um estudo experimental para verificar a correlação entre a taxa de perda de pacotes (TPP) e a potência média no canal de rádios IEEE 802.15.4 sujeitos à interferência de uma rede IEEE 802.11 e de um forno microondas. Os resultados mostraram que essas fontes de interferência aumentam significativamente o nível de energia nos canais, influenciando diretamente o desempenho de comunicação dos

rádios. Em alguns cenários, a TPP dos rádios IEEE 802.15.4 chegou a 90% quando sujeitos à interferência da rede IEEE 802.11. Quando sujeitos à interferência do forno microondas, a TPP no pior caso foi em torno de 50% operando nos canais 21 e 23. No trabalho de [Delgado Gomes et al. 2012] foi observada alguma correlação entre o nível de potência médio no canal e a TPP. Porém, para cada tipo de fonte de interferência, a relação entre potência média e a TPP é diferente.

Em [Lima-Filho et al. 2012] experimentos foram realizados de modo a analisar o impacto de uma rede IEEE 802.11 e um forno microondas no desempenho de uma RSSF para monitoramento de motores em ambiente industrial. Resultados semelhantes aos descritos em [Delgado Gomes et al. 2012] foram observados, com a TPP chegando a aproximadamente 90% quando a RSSF estava sujeita à interferência de uma rede IEEE 802.11. Resultados semelhantes também foram descritos em [Sikora e Groza 2005].

Em [Guo, Healy e Zhou 2012] são descritos experimentos para verificar o impacto da distância entre transmissor e receptor e entre os nós sensores e as fontes de interferência (rádios *Bluetooth*, IEEE 802.11 e forno microondas) em ambiente predial. Observou-se uma TPP de até 25% devido às interferências da rede IEEE 802.11 e forno microondas para os cenários estudados. A interferência por parte dos rádios *Bluetooth* foi muito pequena.

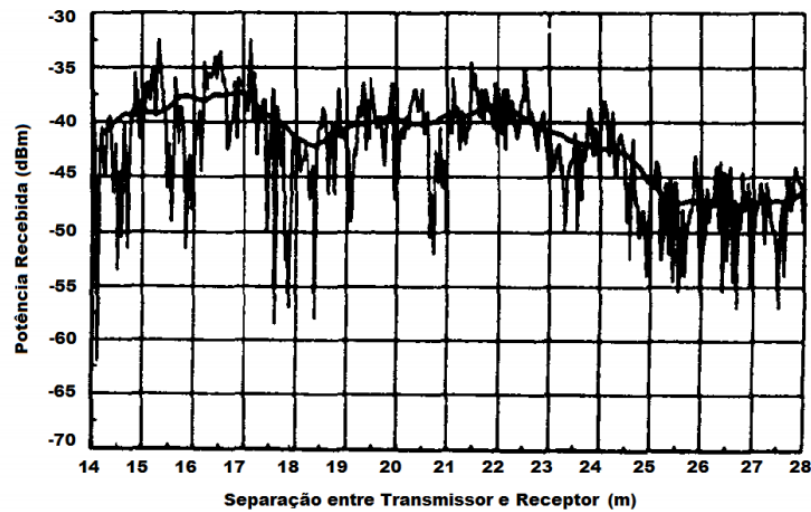
Em [Guo, Healy e Zhou 2012] a interferência do *Bluetooth* foi muito pequena devido ao mecanismo de salto de frequência definido pelo padrão IEEE 802.15.1, que utiliza 79 canais de 1 MHz ao longo da banda de 2,4 GHz.

2.1.2 RSSFs Industriais

Embora apresente vantagens em comparação com as redes cabeadas, como a facilidade de instalação e manutenção e o custo menor, a implantação de RSSF em ambientes industriais deve lidar com problemas típicos de redes sem fio, como interferência eletromagnética causada por outras redes ou por equipamentos industriais e com alto nível de atenuação de pequena e larga escala, dada a presença de muitos objetos metálicos nesses ambientes [Tanghe et al. 2008]. Além disso, os nós sensores usualmente possuem restrições de processamento e largura de banda disponível para comunicação. Todas essas restrições devem ser levadas em consideração no projeto de uma RSSF para operação em ambiente industrial.

Na Fig. 2.3, adaptada de [Rappaport 2009], é possível distinguir dois comportamentos

Figura 2.3: Desvanecimento em larga escala e atenuação em pequena escala.



distintos. As flutuações rápidas do sinal são denominadas atenuação em pequena escala, associadas ao multipercurso. À variação mais suave e lenta dá-se a denominação de desvanecimento em larga escala, associada ao sombreamento por obstáculos.

Em uma RSSF nesse tipo de ambiente, os sensores são implantados em equipamentos industriais para monitorar parâmetros críticos como vibração, temperatura, pressão e eficiência [Delgado Gomes et al. 2013] e para otimizar o processo de produção. As medidas obtidas pelos sensores são transmitidas sem fio até um nó sorvedouro, que provê as informações para análise em uma central de monitoramento. Baseando-se nas informações transmitidas pelos sensores, é possível reparar ou substituir equipamentos defeituosos antes que maiores prejuízos ocorram [Gungor e Hancke 2009].

Nesses ambientes, diferentes fontes de interferência para RSSF podem ser encontradas, como equipamentos de solda, fornos microondas, e outros equipamentos de comunicação sem fio, como redes locais e *Bluetooth*. Os nós em uma rede podem sofrer interferência a partir da coexistência com os outros nós da mesma rede, de outras redes e a partir de outros dispositivos que causam interferência no espectro de comunicação. Como o canal é compartilhado, é possível ainda que existam ataques à rede, gerando informações distorcidas e que degradam a qualidade de transmissão.

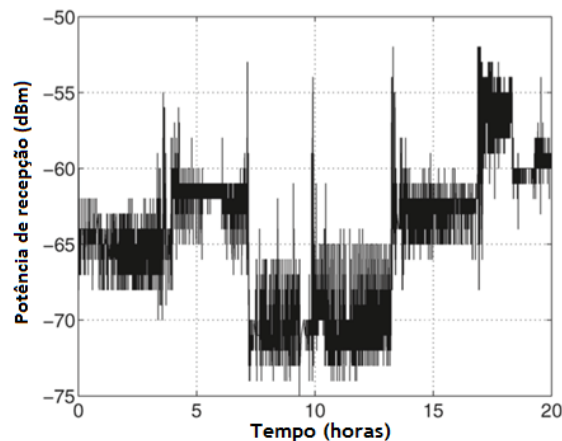
Muitos sistemas de monitoramento industrial necessitam processar sinais heterogêneos que mudam rapidamente e devem ser obtidos utilizando uma alta taxa de amostragem [Hou e Bergmann 2012] [Delgado Gomes et al. 2013]. Algumas aplicações, como o monitora-

mento de eficiência e a detecção de falhas de motores de indução, se encaixam nesse tipo de aplicação [Bin Lu e Gungor 2009] [Lima-Filho et al. 2012] [Delgado Gomes et al. 2013] [Esfahani, Wang e Sundararajan 2014] [Hou e Bergmann 2012].

Características não-estacionárias dos canais de comunicação

Em ambientes industriais, diferenças nas características dos canais podem ser observadas devido ao perfil de multipercurso do ambiente, uma vez que os canais são descorrelacionados em frequência e o impacto do multipercurso é diferente em canais diferentes [Gomes et al. 2014]. Dessa forma, o canal sem fio no ambiente industrial pode permanecer durante várias horas apresentando as mesmas características e após esse período uma mudança brusca nas características do canal pode ocorrer [Agrawal et al. 2014]. Por exemplo, o movimento de objetos, metálicos ou não, pode causar modificações aleatórias no canal no decorrer do tempo.

Figura 2.4: Potência de recepção em experimento realizado por 20 horas.



Em [Agrawal et al. 2014], uma caracterização do canal em um ambiente industrial foi realizada para avaliar as características do canal em um longo período de tempo (20 horas), ilustrado na Fig. 2.4. O RSSI (*Received Signal Strength Indication*) é um indicador de intensidade do sinal recebido, que mede a potência desse sinal. O movimento de uma grande estrutura metálica pode modificar o perfil de multipercurso relacionado ao conjunto de raios que permanecem invariantes por um longo período de tempo, o que modifica o valor médio da potência recebida, embora o transmissor e o receptor permaneçam estáticos.

Em [Remley et al. 2009] foram descritas medições de propagação em ambientes in-

dustriais, visando avaliar fatores ambientais que influenciam na qualidade do canal, como a geometria do ambiente e as características dos objetos existentes no ambiente.

Em [Tang et al. 2007] foi realizada uma caracterização do canal de comunicação em um ambiente industrial, utilizando rádios IEEE 802.15.4. Nesse trabalho, foi estudado o efeito no desempenho de comunicação de rádios IEEE 802.15.4 devido à topologia do ambiente.

Durante os experimentos realizados por [Tang et al. 2007], todas as máquinas e equipamentos de rede foram desligados, de modo a evitar interferência externa. Foram verificadas variações temporais e espaciais no desempenho. Para verificar a variação espacial, um galpão com tamanho de 15 m x 11 m foi dividido em uma grade, com células de 1 m². O nó receptor foi posicionado no centro do ambiente e o nó transmissor foi posicionado em cada um dos outros locais na grade. Para cada local foram medidos a TPP, o nível de energia no canal (RSSI) e o indicador de qualidade do canal (LQI). Para verificar variações temporais, os três parâmetros (TPP, RSSI e LQI) foram monitorados em uma posição específica durante 24 horas. Nesses experimentos foi utilizado o canal 26 nos rádios com potência de transmissão de -5 dBm e pacotes de 94 bytes. Na análise temporal, observou-se mínima variação, uma vez que não existiam objetos móveis nem fontes de interferência no ambiente.

A identificação da relação entre a qualidade dos canais com a topologia do ambiente pode permitir o desenvolvimento de uma metodologia para predizer a qualidade de um canal, dado um determinado leiaute do ambiente [Tang et al. 2007]. Mais especificamente, pode-se simular o perfil de multipercurso no ambiente. Por meio das informações obtidas nessa simulação o planejamento de implantação da rede pode ser feito visando mitigar os problemas de atenuação por multipercurso.

Uma limitação do trabalho de [Tang et al. 2007] é que ele realizou experimentos em apenas um canal. Experimentos realizados em outros canais poderiam fornecer as variações do perfil de multipercurso em diferentes frequências. Outra limitação é que o ambiente considerado foi muito pequeno (distância entre transmissor e receptor menor que 10 metros). Dessa forma, observou-se boa qualidade de comunicação em todos os pontos do ambiente (TPP menor que 6%). Mesmo com essa limitação foi possível observar variações de qualidade entre diferentes pontos. Mais especificamente, os locais que possuíam uma maior quantidade de materiais metálicos ao redor e sem visada direta apresentaram pior qualidade de canal. Outro fenômeno que pode ser observado em ambientes industriais é a assimetria de canal.

Como o ambiente influencia de maneira diferente dois nós que estão comunicando entre si, a qualidade do canal pode ser melhor em uma direção do que em outra, como foi observado nos resultados descritos em [Tang et al. 2007].

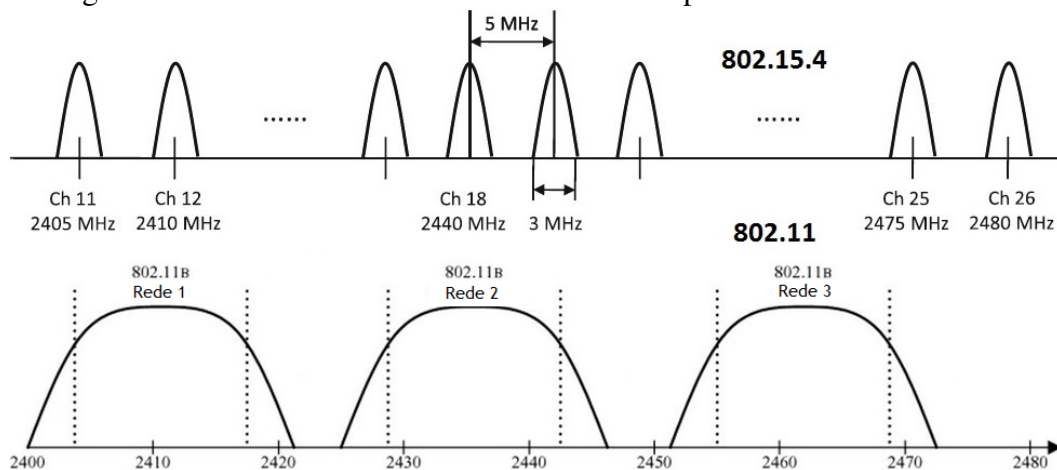
Em [Tanghe et al. 2008] foram realizados estudos sobre a variação temporal no nível de energia dos canais. Essa variação temporal está ligada principalmente à movimentação de obstáculos ao redor do transmissor e do receptor e ocorre devido à modificação do perfil de multipercurso no tempo. Os experimentos demonstrados em [Tanghe et al. 2008] validaram o uso da distribuição de Rice como modelo para a atenuação temporal.

2.2 Padrão IEEE 802.15.4

O IEEE 802.15.4, também chamado de LR-WPAN (*Low-Rate Wireless Personal Area Networks*), ou Redes Pessoais sem Fio com Baixa Taxa de Transmissão, é um padrão designado para aplicações de RSSF e define as camadas física e de acesso ao meio, provendo comunicação sem fio com baixo consumo de energia e baixo custo para aplicações que não requerem alta taxa de transferência de dados. Alguns protocolos implementam as camadas superiores sobre esse padrão, sendo o *ZigBee* um dos mais populares.

O padrão opera em bandas não licenciadas em que são definidos um total de 27 canais para comunicação. Como as bandas utilizadas pelo padrão não são licenciadas, os rádios que seguem esse padrão compartilham o meio de comunicação com outros dispositivos que implementam outras tecnologias, como rádios IEEE 802.11 e IEEE 802.15.1. Como compartilham a banda, existe uma sobreposição entre os canais definidos por esses padrões. A Fig. 2.5, adaptada de [Yang, Xu e Gidlund 2011], mostra os canais definidos pelos padrões IEEE 802.11 e IEEE 802.15.4 na banda de 2,4 GHz. Na figura apenas três canais (1, 6 e 11) do padrão IEEE 802.11 são mostrados, mas o padrão define 14 canais (apenas 11 permitidos no Brasil) que são dispostos no espectro, de modo que existe sobreposição entre canais vizinhos. No padrão IEEE 802.15.4 todos os canais são separados por um intervalo de 5 MHz. Entretanto, devido à modulação utilizada na camada física, uma pequena fração do sinal é espalhada para fora do intervalo de 5 MHz [Lo Bello e Toscano 2009]. Dessa forma, redes IEEE 802.15.4 que estão funcionando em um mesmo ambiente e em canais adjacentes podem interferir uma na outra.

Figura 2.5: Padrões IEEE 802.15.4 e 802.11 compartilham a mesma banda.



Embora esses padrões compartilhem a mesma banda, é possível que múltiplas redes co-existam no mesmo ambiente sem causar interferência entre si, se os canais forem alocados de maneira adequada.

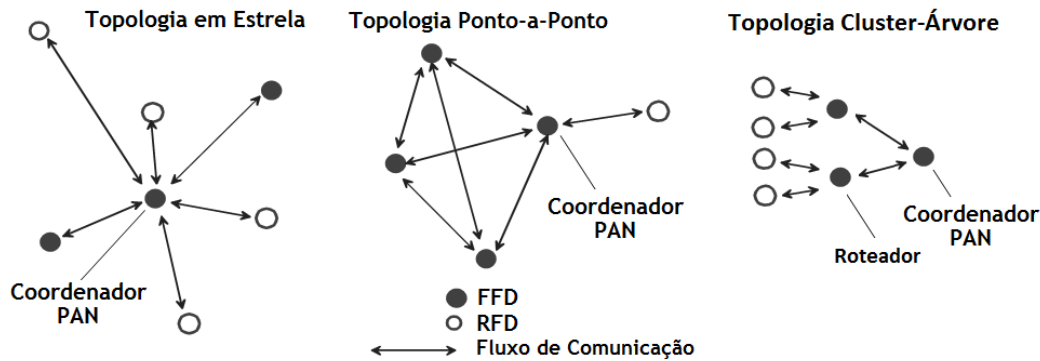
Estudos experimentais descritos em [Jin-Shyan Lee 2006] mostraram que a taxa de bits máxima real do padrão é em torno de 153 *kbit/s*. Para algumas aplicações industriais essa taxa de transferência pode ser suficiente, mas outras aplicações requerem a aquisição de uma grande quantidade de dados e necessitam transmitir muitas informações pela rede. Utilizando processamento local nos nós sensores, é possível reduzir a quantidade de dados transmitidos pela rede.

Além dessa limitação, é importante notar que a taxa de transferência de bits para um determinado nó sensor pode ser muito menor que 153 *kbit/s* ao considerar o compartilhamento do meio de comunicação com múltiplos nós na mesma RSSF e o baixo nível de qualidade do meio de comunicação, dada a presença de fontes de interferência e o desvanecimento em pequena e larga escala no ambiente industrial.

Dependendo dos requisitos da aplicação, uma rede IEEE 802.15.4 pode operar em três topologias [Society 2006]: estrela, ponto-a-ponto ou árvore, ilustradas na Fig. 2.6.

Na primeira, a comunicação é feita entre os dispositivos e um coordenador. Na segunda há também a presença do coordenador, mas qualquer dispositivo pode se comunicar com qualquer outro, desde que estejam no alcance do sinal. A terceira é formada por uma hierarquia na qual os nós finais se comunicam com o roteador, que repassa as informações para o coordenador da PAN.

Figura 2.6: Exemplos de Topologias.



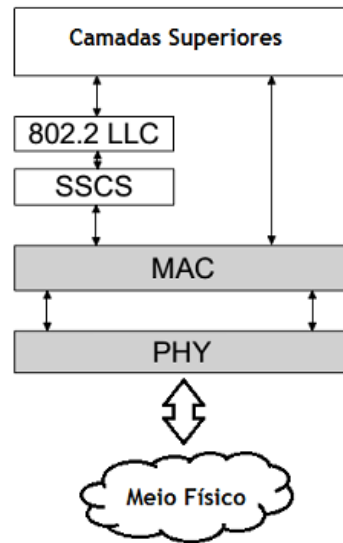
Como ilustrado na Fig. 2.6, os dispositivos IEEE 802.15.4/ZigBee podem ser classificados de acordo com suas funcionalidades [Tennina et al. 2013]: *Full Function Devices* (FFD), que implementam a pilha completa do protocolo, e *Reduced Function Devices* (RFD), que implementam um subconjunto da pilha do protocolo. Em relação ao seu papel na rede, o ZigBee define três tipos de dispositivos:

- **Coordenador Zigbee (ZC) / Gateway (GW) / Coordenador PAN:** é configurado um para cada rede. Age como um coordenador da WPAN ou como um roteador Zigbee (ZR) uma vez que a rede esteja formada. Ele é FFD e, se a rede estiver operando no modo *beacon*, o coordenador envia periodicamente *beacon frames* que sincronizam os nós na rede; o *beacon* é um frame de sincronismo;
- **Roteador Zigbee (ZR):** participa do roteamento de múltiplos saltos de mensagens;
- **Nó Sensor (SN):** implementa um subconjunto reduzido da pilha do protocolo. Ele é um RFD e não permite os dispositivos se associarem a ele, além de não participar do roteamento.

Embora sejam suportadas as três topologias, o padrão não define a implementação da camada de rede, por exemplo, com algoritmos de roteamento. Essa é uma tarefa definida para os protocolos que implementam as camadas superiores do padrão. Em relação à arquitetura do IEEE 802.15.4 [Society 2006], ela é dividida em camadas, como ilustrado na Fig. 2.7.

Por camadas superiores entende-se a camada de rede e a de aplicação. Abaixo delas encontra-se a camada de enlace, subdividida em LLC (*Logical Link Control*), ou Controle do Enlace Lógico, e MAC (*Media Access Control*), ou Controle de Acesso ao Meio. O SS

Figura 2.7: Arquitetura do padrão IEEE 802.15.4.



(*Service Specific Convergence Sublayer*), ou Sub-camada de Convergência, é uma subcamada de convergência. Um LLC proprietário pode acessar a camada MAC diretamente sem passar pelo SSCS, que assegura compatibilidade entre diferentes subcamadas LLC [Cordeiro e Agrawal 2011].

A camada física (PHY) do padrão é responsável pela transmissão e recepção de dados utilizando um canal e de acordo com uma modulação específica e técnica de espalhamento [Tennina et al. 2013].

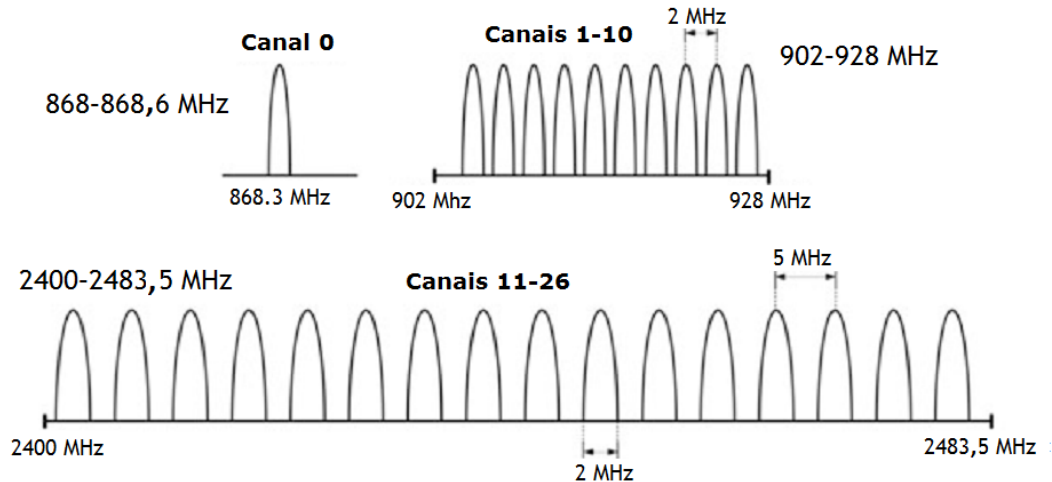
O rádio da camada física opera em uma ou mais das seguintes bandas não licenciadas, ilustradas na Fig. 2.8:

- 868–868,6 MHz (exemplo, Europa) - (20 kbit/s);
- 902–928 MHz (exemplo, América do Norte) - 10 canais - (40 kbit/s);
- 2400–2483,5 MHz (Mundial) - 16 canais - (250 kbit/s).

A camada física possui as seguintes tarefas [Mahalik 2007]:

- Modificação de estado do transceptor: é possível operar no modo de transmissão, de recepção ou no modo de descanso (*sleeping*);
- Detecção de energia: estima a potência do sinal recebido dentro da largura de banda de um canal;

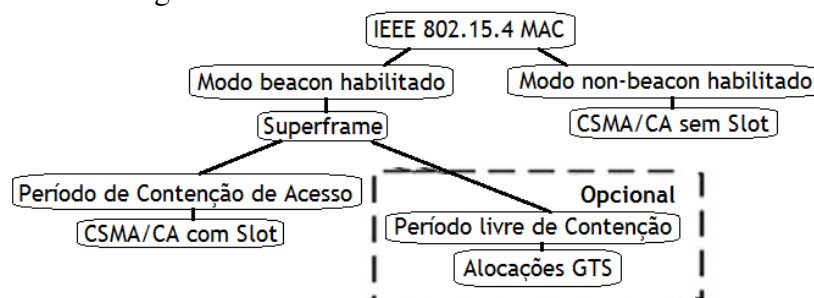
Figura 2.8: Frequências de operação e bandas no IEEE 802.15.4 na camada física.



- LQI: mede a qualidade da transmissão de um pacote recebido;
- CCA: avalia o estado de atividade do meio (disponível ou ocupado);
- Seleção da frequência do canal: quando requisitado por uma camada superior, é possível alterar o canal em uso pelo transceptor.

A camada MAC suporta dois modos operacionais, que estão ilustrados na Fig. 2.9, adaptada de [Tennina et al. 2013]:

Figura 2.9: Camada MAC do IEEE 802.15.4.

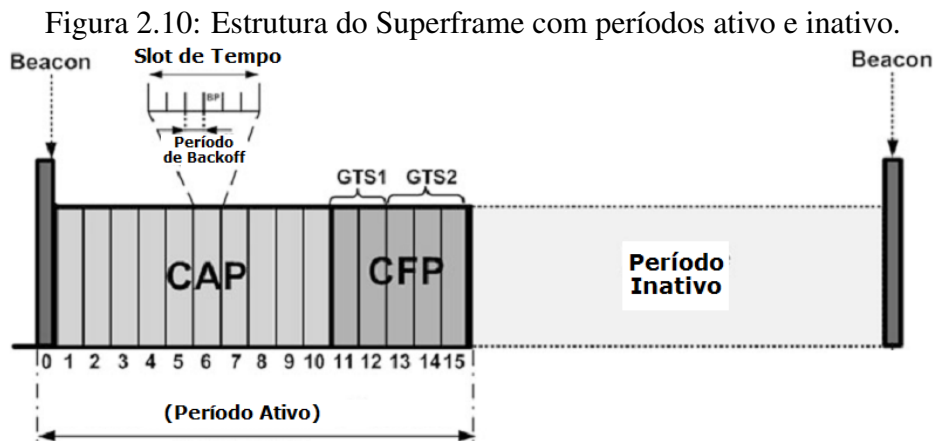


- Modo *beacon*: nesse modo, *beacons* são enviados periodicamente pelo coordenador ou roteador para sincronizar os nós associados a ele e para identificar a WPAN. Um *beacon frame* delimita o início de um superframe definindo um intervalo de tempo durante o qual frames são trocados entre diferentes nós na PAN. O acesso ao meio é coordenado pelo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), ou Acesso Múltiplo com Verificação de Portadora com Prevenção de Colisão,

com slots. Esse modo também habilita a alocação de slots de contenção de tempo livre, chamado GTS, para nós que requisitam largura de banda garantida;

- Modo *non-beacon*: quando um coordenador seleciona o modo *non-beacon*, não há *beacons* nem superframes; o acesso ao meio é coordenado pelo mecanismo CSMA/CA sem slot.

O superframe é definido entre dois *beacon frames* e possui os períodos ativo e inativo. Como está ilustrada na Fig. 2.10, adaptada de [Tennina et al. 2013], a porção ativa do superframe é composta de três partes, o *Beacon*, o Período de Acesso por Contenção (CAP) e o Período Livre de Contenção (CFP):



- **Beacon**: o *beacon frame* é transmitido no começo do slot 0. Ele contém informações de endereçamento, especificação do superframe, GTS e outros dados;
- **CAP**: começa imediatamente após o *beacon frame* e finaliza antes do começo do CFP, se esse existir. Se não existir, o CAP é concluído no final da parte ativa do superframe. Todas as transmissões nesse período são feitas utilizando o mecanismo do CSMA/CA com slot (*slotted CSMA/CA*);
- **CFP**: começa imediatamente depois do final do CAP e finaliza antes do próximo *beacon frame* ou do final do superframe. As transmissões são realizadas sem uso do CSMA/CA.

O mecanismo de GTS permite que os dispositivos acessem o meio sem contenção, no CFP. Os GTSs são alocados pelo coordenador e utilizados somente na comunicação entre coordenador e um dispositivo. Cada GTS pode conter um ou mais slots de tempo. O coordenador pode alocar até 7 GTSs no mesmo superframe, e eles possuem apenas uma direção: dispositivo $\rightarrow$ coordenador (transmissão), ou coordenador $\rightarrow$ dispositivo (recepção) [Tenina et al. 2013].

2.3 Protocolos Multicanais

Entre os fatores que podem interferir na qualidade de comunicação entre os nós estão a alta atenuação temporal na indústria devido à presença de muitos objetos metálicos e obstruções, e redes que trabalham no mesmo ambiente e que usam a mesma faixa de frequência e em canais adjacente. Para diminuir esses problemas, uma alternativa é o desenvolvimento de protocolos que utilizam múltiplos canais.

O uso de protocolos multicanais permite alcançar uma melhor utilização do canal, aumentando a capacidade de transmissão da rede por meio da transmissão em simultâneo utilizando vários canais diferentes. Também é possível atender às variações espaciais de qualidade dos canais [Incel 2011].

Um grande número de protocolos multicanais foram propostos para as camadas MAC, de roteamento, e de transporte. Um exemplo de protocolo da camada MAC é o CMAC (Multi-channel energy efficient MAC for wireless sensor networks), ou Protocolo MAC Eficiente em Energia com múltiplos canais para redes de sensores sem fio, que utiliza dois rádios, um half-duplex para transmissão dos dados e outro para sinalização de controle. Resultados demonstraram que esse protocolo pode diminuir em 200% o consumo de bateria e de 5 a 150% a latência, se comparada com o protocolo SMAC [Namboothiri e Sivalingam 2013].

Foram também implementados os protocolos MMSN, TMMAC e MCMAC, desenvolvidos especificamente para RSSF [Wu et al. 2008]. Os três tentam atribuir canais aos nós a dois saltos para evitar potenciais interferências. Esse tipo de protocolo é baseado nos nós, e estudos demonstraram que eles melhoram o desempenho das RSSFs se comparados com a utilização de protocolos de canal único. A desvantagem deles é que são necessárias várias mensagens de controle ao informar a mudança de canal, o que pode causar perda de pacotes.

tes, perda de sincronismo dos nós e, conseqüentemente, aumento no consumo da bateria em ambientes sem fonte de alimentação constante.

Experimentos descritos em [Amzucu, Li e Fledderus 2014] mostraram que trocar o canal de comunicação pode levar a uma variação de até 30 dB na potência recebida, com os nós posicionados em locais fixos e em um ambiente de escritório. Essa diferença pode ser ainda maior em ambientes mais refletivos. Segundo o estudo realizado em [Wu et al. 2008], os protocolos multicanais baseados nos nós não são adequados para RSSFs, pois neles, em grandes redes, é necessário um grande número de canais, além de que também é necessário ter uma boa precisão na sincronização entre os nós.

Como alternativa a essa abordagem, em [Pal e Nasipuri 2013] é proposto o protocolo DRCS, que executa seleção de canal e roteamento para melhorar o tempo de vida da bateria. Sua vantagem é que pode ser aplicado sem a necessidade de sincronização de tempo e requer um único transceptor por nó.

Em uma rede com múltiplos nós os pacotes direcionados a outros nós também são escutados por aqueles não envolvidos na comunicação, o que acaba também gerando consumo de bateria desnecessário. Com um único canal de comunicação, esse problema é inevitável.

No protocolo DRCS, os canais disponíveis são distribuídos entre os nós para que cada nó escute no seu canal selecionado por padrão. Para transmissão e repasse, cada nó temporariamente comuta para o canal de seu pai e em seguida retorna ao canal anterior quando a transmissão é completada. Já que os nós só escutam no canal do receptor, a escuta de pacotes externos à comunicação diminui, o que economiza no tempo de vida da bateria.

Outras abordagens utilizam a atribuição de canal conjunta. Essa estratégia é descrita em [Wu et al. 2008]. Nele, os autores propõem um protocolo de múltiplos canais baseado em árvore chamado *Tree-based multichannel protocol* (TMCP), ou Protocolo de Múltiplos Canais Baseado em Árvores). Nesse protocolo, toda a rede é dividida estaticamente em sub-árvores mutualmente exclusivas para reduzir a interferência. No TMCP são alocados diferentes canais para cada sub-árvore, e então cada nó repassa cada fluxo de dados apenas pela sua sub-árvore correspondente. Essa abordagem funciona bem com um número pequeno de canais e possui um esquema simples de transmissão, sem a necessidade de sincronização dos nós, o que o torna adequado para RSSFs.

De modo a avaliar os reais benefícios desses protocolos multicanais e desenvolver novos

e mais adequados a RSSFs, torna-se necessária a utilização de simuladores de rede.

2.4 Simuladores de Rede

Existem três técnicas para realizar a avaliação de desempenho de redes e sistemas [Wehrle, Gunes e Gross 2010]: análise matemática, medições e simulação computacional. As medições requerem a implementação de um ambiente real, que em geral possui custos como aquisição de equipamentos, montagem da estrutura física, risco de modificar uma rede ativa, alocação de espaço, enquanto que na análise matemática corre-se o risco de o modelo matemático se tornar intratável, além de oferecer uma visão limitada do sistema. Por isso, a simulação computacional é usualmente escolhida porque ela permite comparar alternativas de projeto com boa precisão e baixo custo.

Um simulador pode ser usado para comparar modelos teóricos e computacionais que, se fossem vistos apenas do ponto de vista conceitual, seriam um pouco difíceis de compreender. Por essa razão, os simuladores possuem um papel crucial na indústria e nas pesquisas científicas, uma vez que a simulação permite visualizar possíveis problemas antes da implantação da estrutura na indústria.

A seguir são listadas algumas técnicas utilizadas para simular um sistema [Wehrle, Gunes e Gross 2010] [Guizani et al. 2010] [Obaidat e Papadimitriou 2003]:

- Emulação: processo de projeto e construção de hardware e software que imita a funcionalidade de um sistema real;
- Método Monte Carlo: por ser um método estatístico, baseia-se na repetição da simulação um elevado número de vezes para capturar o comportamento do sistema em análise;
- Simulação orientada a indícios: utiliza uma lista ordenada de eventos do mundo real como entrada;
- Simulação contínua: variáveis de estado se modificam continuamente. Se alcançarem um determinado limiar, um evento ocorre;

- Simulação de eventos discretos: baseia-se na duração da simulação ou em quantidade finita de eventos.

Há muitos simuladores de eventos discretos cujo foco é a modelagem da rede e da pilha de protocolos e avaliação de desempenho. Baseando-se na complexidade, esses simuladores podem ser classificados em três categorias [Shu et al. 2011]:

- Simuladores em nível de pacotes: implementam ambas as camadas de enlace e física do modelo OSI. O NS3¹ (*Network Simulator 3*) é um dos mais utilizados, seguido pelo Omnet++² (*Objective Modular Network Testbed in C++*). O primeiro foi desenvolvido para redes IP e pode também ser usado para RSSF. O segundo possui suporte a redes de sensores, redes sem fio *ad hoc* e protocolos de Internet;
- Simuladores em nível de instrução: alguns simuladores modelam a execução de protocolos em nível de instrução e são considerados emuladores. O TOSSIM é um exemplo desses simuladores. Ele simula protocolos que executam sobre o TinyOS em nível de bit. O TinyOS é um sistema operacional embarcado baseado em componentes e escrito em nesC, dialeto da linguagem C otimizado para redes de sensores com memória limitada. Sua desvantagem é que simula apenas códigos para o TinyOS;
- Simuladores em nível de algoritmo: foca na lógica, na estrutura de dados, e na apresentação dos algoritmos. O Shawn³ é um simulador de eventos discretos para redes de sensores e possui habilidade de alta customização e pode ser ajustado a diferentes níveis de precisão.

Os modelos de simulação podem ser implementados em diversas linguagens, incluindo linguagens de programação de propósito geral (não específicas para simulação), tais como FORTRAN, C, C++ e Java, por exemplo [Obaidat e Papadimitriou 2003], em linguagens específicas como GPSS, GASP, SIMAN, SLAM II, SIMULA, SIMSCRIPT 11.5, MODSIM III, CSIM, e em pacotes de simulação, tais como Arena, Automod, OPNET, NS3 e Omnet++.

Apesar da existência de muitos simuladores para os mais diversos ambientes, esse trabalho foca no estudo do simulador de rede que implementa RSSFs no padrão IEEE 802.15.4, e

¹<http://www.nsnam.org> - Acessado em 02/02/2016.

²<http://www.omnetpp.org> - Acessado em 02/02/2016.

³<https://github.com/itm/shawn/> - Acessado em 02/02/2016.

que trabalha na banda de 2,4 GHz. Entre os principais simuladores com essas características estão o NS3 e o Castalia, sendo esse último baseado no Omnet++.

Entre os simuladores de rede mais utilizados estão o NS3 (e a versão anterior, NS2), desenvolvido em C++ e Python, e o Omnet++, que é um *framework* de simulação baseado em componentes.

O NS3 é um simulador incompatível com suas versões anteriores. Uma comparação entre o NS2 e NS3 é encontrada em [Font et al. 2011]. Ele permite, por meio de suas subclasses, receber pacotes de uma interface real e enviar pacotes simulados através da mesma rede. Sua mais recente versão (3.24.1) executa em sistemas Linux, FreeBSD e OS X. A Fig. 2.11⁴ lista todos os módulos suportados pelo NS3, incluindo os modelos para LR-WPAN.

Figura 2.11: Lista de módulos implementados pelo simulador NS3.

6LoWPAN	Core	MPI Distributed Simulation	Propagation Models
AODV Routing	DSDV Routing	Mesh Device	RIPng
Applications	DSR Routing	Mobility	Spectrum Models
BRITE Topology Generator	Emu Network Device	Network	Statistics
Bridge Network Device	Energy Models	Network Animation	Tap Bridge Network Device
CSMA Layout Helpers	File Descriptor Network Device	Nix-Vector Routing	Topology Input Readers
CSMA Network Device	Flow Monitor	OLSR Routing	UAN Models
Click Routing	Internet	OpenFlow Switch Device	Utils
Configuration Store/Load	LR-WPAN models	Point-To-Point Network Device	WiMAX Models
Constants	LTE Models	Point-to-Point Layout Helpers	Wifi Models
	Visualizer	Virtual Device	

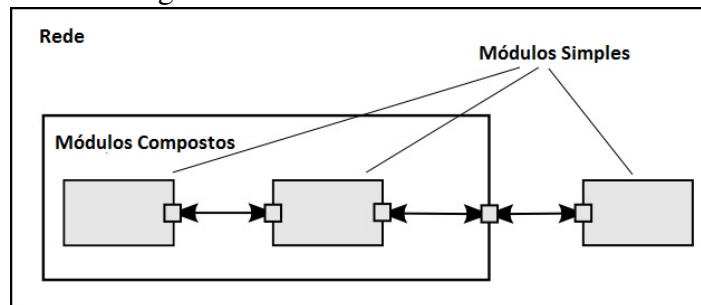
O Omnet++ é um simulador com extensiva documentação e possui IDE baseada no Eclipse. Livre para modificar, utilizar e distribuir sob a *Academic Public Licence*, ele é orientado a objetos, modular, e pode ser executado no Linux, Mac OS/X e Windows. Ele é composto de módulos que se comunicam entre si por meio de mensagens [Khan, Pathan e Alrajeh 2012]. Os módulos são escritos em C++ e então montados em componentes maiores e modelos utilizando a linguagem de alto nível NED (Network Description) [R. et al. 2013]. Cada módulo tem um ou mais parâmetros que afetam seu comportamento. Um ar-

⁴<https://www.nsnam.org/doxygen/modules.html> - Acessado em 02/02/2016.

quivo *.ned* define a estrutura básica de um módulo, inserindo suas portas de entrada e saída e seus parâmetros [Castalia User's Manual].

No Omnet++ um dispositivo ou objeto pode ser definido por um módulo simples ou composto e é ilustrado na Fig. 2.12, adaptada de [Omnet++ Homepage]. O primeiro caso é o elemento mais básico na estrutura do Omnet++, possuindo interfaces de entrada e saída e seus parâmetros escritos na linguagem NED e seu comportamento na linguagem C++. No segundo caso, os módulos são formados pela junção de módulos simples, e as portas são utilizadas para comunicação entre os módulos.

Figura 2.12: Estrutura do Omnet++.



Embora não seja exatamente um simulador de rede [Omnet++ Homepage], mas um *framework* que permite a construção de simuladores de rede [Chamberlain 2013], o Omnet++ ganhou popularidade como uma plataforma de simulação na comunidade científica e provê infraestrutura e ferramentas para escrever simulações.

O Omnet++ possui os seguintes *frameworks*: Castalia, MiXiM, INET, Mobility e NesCT. O MiXiM, a partir da versão 2.0, passou a suportar a integração com o INET, permitindo construir simulações com ambos os *frameworks* ao mesmo tempo⁵. Com o exemplo do "Mixnet", pode-se utilizar as interfaces de rede do MiXiM com as altas camadas dos protocolos (TCP, UDP, IP) do INET. O *framework* Mobility foi desenvolvido pela Universidade Técnica de Berlim e provê apenas suporte básico a redes sem fio e móveis, e o NesCT não é um *framework* real, mas um tradutor da linguagem de programação NesC para classes C++ para o Omnet++ [Ali 2012].

O Castalia é um *framework* que permite aos usuários testar e validar algoritmos, protocolos e aplicações para RSSFs [Tennina et al. 2013]. De acordo com os desenvolvedores, esse

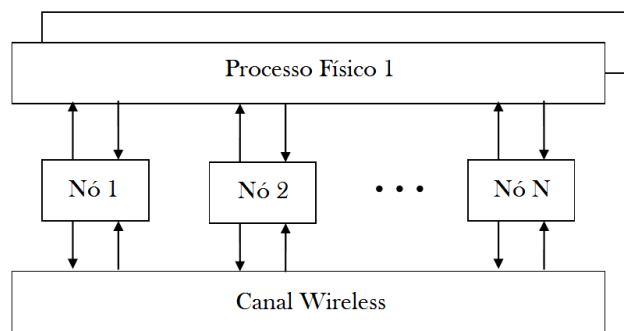
⁵<http://www.omnetpp.org/component/content/article/9-software/>

é o mais realista para RSSF e BAN (*Body Area Network*). Seu módulo de rádio é baseado em rádios reais para dispositivos embarcados de baixa potência, tais como nós sensores sem fio [Stehlík 2011], incluindo suporte aos transceptores CC2420 [Tennina et al. 2013] e CC1000, e ao modelo de sombreamento log-normal (*Log-normal Shadowing*), que provê estimativas precisas para a perda de percurso média.

2.4.1 Castalia

O Castalia é um simulador de eventos discretos cuja modularidade e confiabilidade é parcialmente habilitada pelo Omnet++ [Castalia User's Manual]. Sua estrutura é baseada em módulos, como ilustrado na Fig. 2.13, adaptada de [Tennina et al. 2013].

Figura 2.13: Os módulos e conexões no Castalia.

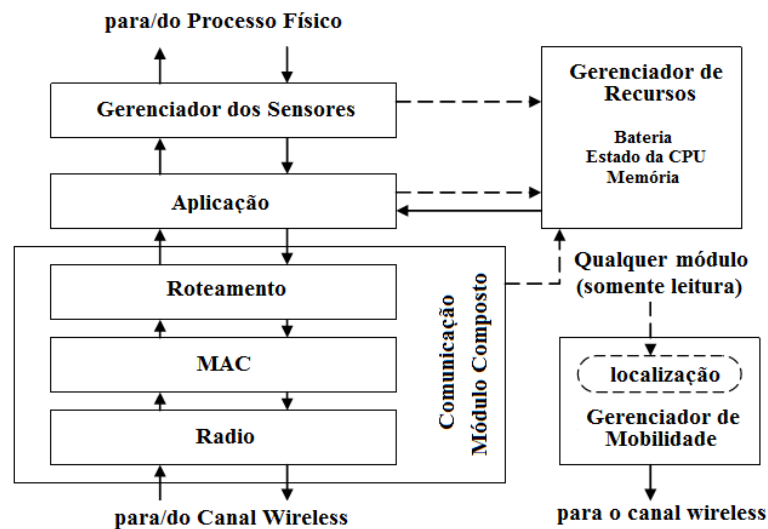


Na estrutura, os nós não se conectam entre si diretamente, mas através do mesmo módulo do Canal *Wireless*; por padrão, todos os nós trabalham no mesmo canal. Quando possui um pacote a ser enviado, o nó passa primeiro pelo canal *wireless*, que então decide qual nó receberá o pacote. Eles também são ligados através dos processos físicos os quais monitoram.

Seguindo para uma análise mais detalhada do nó, seu módulo com a estrutura interna é ilustrado na Fig. 2.14, adaptada de [Castalia User's Manual]. As setas identificam as mensagens sendo trocadas entre os módulos e as setas pontilhadas indicam uma chamada simples de função. Todos os módulos, simples e compostos, chamam uma função de gerenciamento de recursos que sinaliza o quanto de energia foi consumido. Apesar da existência do módulo do Gerenciamento de Mobilidade, ele não será utilizado nesse trabalho, pois os nós deverão estar estáticos.

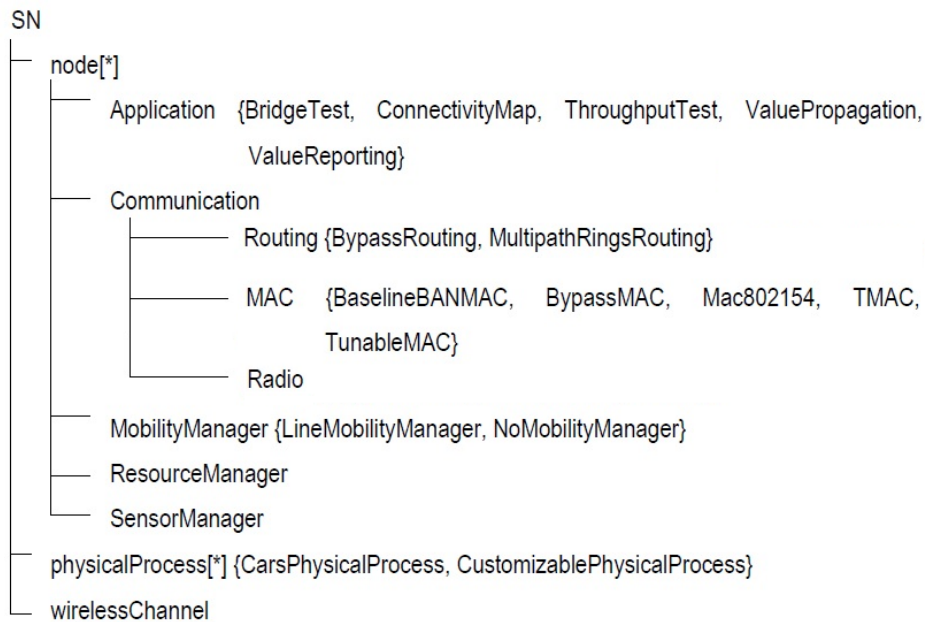
Com relação aos módulos e seus relacionamentos hierárquicos, eles estão listados na Fig. 2.15. Os módulos *Application*, *MAC*, *Communication*, *Routing* e o *Mobility Manager*

Figura 2.14: O módulo composto do nó (estrutura).



são os quais o usuário mais irá manipular para criar outros módulos e implementar novos algoritmos.

Figura 2.15: Lista de módulos implementados pelo simulador Castalia.



Iniciando o estudo pelo módulo *Application*, há cinco tipos de aplicações que podem ser definidas. Para permitir o uso de parâmetros como taxa de transmissão de pacotes e para que todos os nós direcionem o tráfego ao nó sorvedouro, o *ThroughputTest* é o módulo mais indicado. O nó sorvedouro, por padrão, é o *node[0]*.

Após a definição desse módulo, é possível configurar o de roteamento cujas configura-

ções podem ser abstraídas ao desenvolvedor, como segue:

Código Fonte 2.1: Abstração dos detalhes de roteamento.

```
1 SN.node[*].Communication.RoutingProtocolName = "BypassRouting"
```

A abstração foi utilizada pois o foco do trabalho está nas camadas mais baixas do modelo.

Na camada MAC há diversos parâmetros a serem ajustados para adaptar a simulação aos dispositivos sensores reais e à rede a ser construída, como por exemplo, se um nó é o coordenador, qual o tamanho do *buffer*, etc. Para o propósito desse trabalho, os módulos *Mac802154* (Zigbee MAC) e *TunableMAC* foram estudados e escolhidos.

No módulo *Mac802154* do Castalia, por padrão, são implementadas as seguintes funcionalidades:

- CSMA-CA (com ou sem *slot*);
- WPAN com modo *beacon* habilitado;
- GTS (*Guaranteed Time Slots*).

As seguintes funcionalidades não são implementadas [Castalia User's Manual]:

- WPAN no modo *non-beacon*;
- Topologias de WPAN de múltiplos saltos.

Os detalhes dessas funcionalidades podem ser encontrados na Seção 2.2 (*IEEE 802.15.4*) desse trabalho.

Em relação ao módulo *TunableMAC*, ele simula o comportamento de um protocolo CSMA/CA simples. Nele é empregado o mecanismo CSMA e transmite um conjunto de beacons antes de cada transmissão para "acordar" os nós que irão receber os pacotes. Os valores padrão desse módulo implementam um protocolo CSMA-CA não-persistente sem qualquer ciclo de trabalho (*duty cycle*) no rádio. O *duty cycle* é utilizado para descrever a fração de tempo em que um sistema está em um estado "ativo", escutando um canal. Com valor *1.0*, o nó entra no estado de sono. Esse é provavelmente o parâmetro mais importante que afeta o consumo de energia ao diminuir drasticamente o tempo de escuta [Castalia User's Manual].

Se a implementação CSMA for persistente, o nó monitora continuamente o meio de transmissão até que ele fique livre. Assim que for liberado, ele inicia a transmissão. Já no CSMA não-persistente, se o canal está ocupado, ele não fica monitorando continuamente, mas espera um intervalo de tempo aleatório e tenta novamente.

Em uma comparação entre o módulo *Mac802154* e o *TunableMAC*, esse último utiliza uma versão mais simples do CSMA-CA, com um foco maior para comunicação em *broadcast*, que não suporta pacotes de confirmação. Como nos experimentos na indústria utilizamos nós se comunicando sem a utilização de pacotes de confirmação, no modo non-beacon, e o Castalia não implementa esse modo, escolhemos para os testes de simulação o módulo *TunableMAC*, apesar de os nós reais utilizarem o protocolo IEEE 802.15.4 na camada MAC.

Com relação à camada de rádio, seu comportamento pode ser simulado com o módulo *Radio*. Esse módulo permite definir, por exemplo, a potência de transmissão e a frequência do rádio. Há configurações padrão para os rádios CC2420 e CC1000, que podem ser referenciados da seguinte forma:

Código Fonte 2.2: Instanciação dos parâmetros do Radio.

```
1 SN.node[*].Communication.Radio.RadioParametersFile = "../Parameters/Radio/CC2420.txt"
```

No módulo *WirelessChannel* alguns parâmetros dependem do modelo de canal a ser utilizado. Parâmetros como o expoente de perda de percurso e o sombreamento σ são utilizados pelo modelo de sombreamento log-normal.

Modelos de Canal Wireless

Em ambientes industriais existem muitos objetos metálicos, que influenciam no desvanecimento em larga escala e na atenuação em pequena escala. As perdas em larga escala indicam o valor médio da potência do sinal recebido e em pequena escala indicam o valor instantâneo.

Em relação à potência de recepção do sinal no receptor, ela depende dos seguintes fatores: potência de transmissão, ganho das antenas, distância entre transmissor e receptor e dos efeitos causados pelo ambiente [Gomes et al. 2013].

Como modelos que calculam as perdas em larga escala estão:

- *Free Space* (Espaço Livre): utilizado somente quando o meio está livre de obstáculos. Esse modelo calcula a perda de intensidade do sinal por meio de espaço livre

(habitualmente, o ar), sem obstáculos por perto para causar reflexão ou difração;

- *Two-ray Ground Reflection* (Reflexão no Solo em Dois Raios): leva em consideração não só a propagação direta do sinal, mas também o sinal refletido pelo chão. Esse modelo não possui boa precisão a curtas distâncias;
- *Log-Distance Path Loss* (Perda de Percurso Log-Distância): considera que a potência média recebida diminui logaritmicamente na medida em que a distância aumenta entre transmissor e receptor. Nesse modelo a perda é definida pela Eq. 2.1:

$$L(d) = L(d_0) + 10n \log \left(\frac{d}{d_0} \right), \quad (2.1)$$

em que n representa o expoente de perda de percurso (no Castalia, é identificado pela variável *pathLossExponent*), que pode variar dependendo do ambiente. Em linha reta e sem obstáculos, ele possui valor 2. A perda de percurso L na distância d depende de n e da perda de percurso na distância de referência d_0 . O valor d é o comprimento do percurso e d_0 é uma distância de referência, geralmente 1 km (ou 1 milha). $L(d_0)$ pode ser determinado pela Eq. $L(d_0) = 20 \log(4\pi d_0/\lambda)$, em que λ é o comprimento de onda da portadora;

- *Log-normal Shadowing*: similar ao modelo *Log-Distance Path Loss*, é adicionada uma variável aleatória X_σ de distribuição Gaussiana em dB com média zero e desvio padrão σ . Como n , o σ é também depende do ambiente. Em [Tanghe et al. 2008] experimentos foram conduzidos para determinar ambos n e sombreamento log-normal X_σ em ambientes industriais nas frequências 900, 2400 e 5200 Mhz. Nos experimentos, o n variou entre 0,68 e 4,47, dependendo do ambiente, e o X_σ variou entre 3,16 e 8,63. O modelo é calculado pela Eq. 2.2.

$$L(d) = L(d_0) + 10n \log \left(\frac{d}{d_0} \right) + X_\sigma. \quad (2.2)$$

Observando o arquivo *src/wirelessChannel/WirelessChannel.ned* no Castalia é possível listar os parâmetros que o simulador utiliza como padrão para executar o modelo de canal nas simulações:

Código Fonte 2.3: Valores padrão do modelo do sombreamento log-normal no Castalia.

```

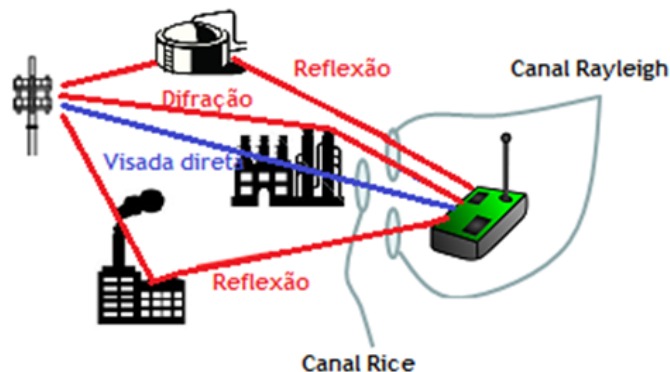
1 double pathLossExponent = default (2.4);
2 double PLd0 = default (55);
3 double d0 = default (1.0);
4 double sigma = default (4.0);

```

Além do desvanecimento em larga escala, há também a atenuação em pequena escala, que trata das mudanças bruscas no perfil de multipercurso do ambiente causado pelo movimento de objetos ao redor do receptor e do transmissor [Gomes et al. 2015].

Há duas importantes distribuições que consideram a atenuação em pequena escala: *Rayleigh* e *Rice*. As duas distribuições estão ilustradas⁶ na Fig. 2.16. A primeira pode ser utilizada quando não há linha de visada entre transmissor e receptor e quando a estrutura do ambiente é aleatória [Garg 2007]. Com relação à distribuição Rice, experimentos demonstraram que em um ambiente industrial a atenuação temporal segue essa distribuição. Ela descreve a atenuação em pequena escala quando existe um sinal estacionário dominante, ilustrado de azul, e componentes aleatórios, de vermelho, que se sobrepõem ao componente principal [Gomes et al. 2015] [Rappaport 2009]. Os resultados mostraram que a distribuição de Rice apenas modela a potência recebida em pequenos períodos de tempo, em que o valor médio da potência recebida permanece constante.

Figura 2.16: Modelos de Canal Rice e Rayleigh.



O fator K de Rice na Eq. 2.3, representa a relação entre a potência do caminho direto e a potência relativa aos outros percursos. Em ambientes industriais, o fator K de Rice possui um valor alto, entre 4 dB e 19 dB, enquanto que em escritórios o fator ficou entre -12 dB e -6 dB [Tanghe et al. 2008]. Isso pode ser explicado pela natureza aberta dos prédios

⁶Adaptada de http://www.sharetechnote.com/image/ChannelModel_Rayleigh_04.PNG - Acessado em 02/02/2016.

Tabela 2.1: Comparação de dois dos principais simuladores.

ASPECTOS	NS3	CASTALIA
Linguagem	C++, Python	C++
Modulação	BPSK, QPSK, DBPSK, DQPSK, QAM	SINR, PSK, FSK
Modelo de Consumo de Energia	SIM	SIM
Licença de Software	GNU GPL	Academic Public License
Sistemas Operacionais	Linux, FreeBSD, Mac OS X, Windows (Cygwin)	Linux/Unix (recomendado), Windows (Cygwin)
Canal Wireless	802.11, YANS, Modelos Free space/ two-ray/shadowing/smallscale path loss	Modelo de Sombreamento Log-normal, path loss map, packet reception rates map, variação temporal
Camada PHY	LTE, LAN, Módulo Mobility	CC1100, CC2420
Camada MAC	802.11, CSMA/CA(CD), QoS, mesh, bridge	TMAC, SMAC, Tunable MAC, IEEE 802.15.4 MAC, Baseline BAN MAC
Camada de Rede	AODV, dsdv, olsr, internet	Simple tree, multipath rings
Camada de Transporte	TCP, UDP	NÃO

industriais e pela ocorrência de muitos materiais refletivos. Dessa forma, existem vários raios invariantes no tempo e apenas uma pequena parte do perfil de multipercurso é afetada por objetos móveis [Tanghe et al. 2008]. Na ausência de um componente dominante, $A \rightarrow 0$, $K \rightarrow -\infty$, e a atenuação de Rice se torna atenuação de Rayleigh [Guimaraes e Adionel 2009]. O A^2 representa a parte que não varia ao longo do tempo, relacionado à potência do componente estacionário, e $2\sigma^2$ representa a parte variável devido ao movimento que modifica os padrões dos componentes de multipercurso.

$$K = 10 \log\left(\frac{A^2}{2\sigma^2}\right) \quad (2.3)$$

2.4.2 Comparação dos Simuladores

Existem diversos trabalhos desenvolvidos focando na comparação entre os principais pacotes de simulação [Wehrle, Gunes e Gross 2010] [Guizani et al. 2010] [Obaidat e Papadimitriou 2003] [Dwivedi e Vyas 2011] [Monika e S. 2014] [Weingartner, Lehn e Wehrle 2009] [Stehlík 2011] [R. et al. 2013] [Sundani et al. 2011]. As comparações consideram características

como modelos suportados, linguagens de programação utilizadas, licença de software, protocolos implementados, entre outras.

Para uma melhor compreensão das diferenças e similaridades entre dois dos principais simuladores utilizados pela comunidade acadêmica, a Tab. 2.1 descreve as características de cada um.

Embora o NS3 seja mais eficiente a fim de simular redes em geral, incluindo o suporte a grande quantidade de modelos de propagação (Fig. 2.11) e implementação da camada de transporte e do padrão IEEE 802.11, o Castalia possui um foco maior em RSSFs, incluindo a utilização de valores reais obtidos por meio de transceptores na camada PHY. Por ser um simulador baseado em rádios reais para dispositivos embarcados de baixa potência, o Castalia foi escolhido para realização desse trabalho.

Há muitos trabalhos que utilizaram o Castalia para propor, por exemplo, adaptação e avaliação de protocolos, e avaliação do comportamento de RSSFs considerando fontes de interferência.

2.4.3 Trabalhos Utilizando o Castalia

No trabalho de [Oliveira 2011] o autor utiliza o simulador Castalia para propor uma adaptação do protocolo MAC com foco em decisão de espectro para RSSFs Cognitivas, em que é feita a seleção do melhor canal disponível. Nesse trabalho, o método utilizado para escolher o melhor canal se baseou na técnica conhecida por *Analytical Hierarchical Process* (AHP). Esse método foi adicionado ao protocolo T-MAC disponível no Castalia, e no qual foram avaliados a taxa de entrega, latência, *overhead* de transmissão, números de trocas e consumo de energia. No trabalho de [Oliveira 2011] as simulações mostraram que as adaptações cognitivas do protocolo T-MAC apresentaram um desempenho de até 69% mais pacotes entregues do que a versão original do protocolo.

No trabalho de [Wykret 2014] foram consideradas fontes de interferência para avaliar o comportamento das RSSFs. Foi feito um protótipo de nó sensor multirrádio e multifrequência para mitigar os problemas de interferência. Esse nó possuía dois rádios (CC2420 e CC1000), o que lhe permitia escolher qual a melhor frequência de operação (2,4 Ghz ou 900 Mhz) e o melhor canal em determinado momento (16 canais do rádio CC2420 e 3 canais do rádio CC1000). Nele foram desenvolvidos dois protocolos de sensoriamento e decisão de

espectro, baseados no protocolo T-MAC e implementados no Castalia, chamados de DSC-MAC (*Dynamic Selection Channel MAC*) e DSCMAC-AHP (*Dynamic Selection Channel MAC-AHP*), que permitem a utilização oportunista do meio de transmissão.

No trabalho de [Wykret 2014] foi utilizada a versão adaptada do Castalia por [Oliveira 2011], o que permitiu realizar uma adequação do módulo de rádio, inserindo múltiplos modelos de rádio e um controlador responsável por selecionar qual estará ativo.

Em [Pal e Nasipuri 2013] o Castalia é utilizado para avaliação de performance do protocolo multicanais DRCS. Seu propósito é comparar o protocolo DRCS e o TMCP. Através de simulações e experimentos, foi demonstrado que o DRCS melhora o tempo de vida da rede sem sacrificar a taxa de entrega de pacotes.

Capítulo 3

Metodologia de Simulação com o Castalia

Como discutido nesse trabalho, o Castalia possui muitas vantagens se comparado com outros simuladores, mas também possui desvantagens. Uma desvantagem é a ausência de um modelo que implemente a comunicação através de múltiplos canais simultaneamente. O padrão IEEE 802.15.4 define 16 canais na banda 2,4 GHz, e protocolos multicanal podem ser utilizados para melhorar a qualidade do serviço nas RSSFs industriais. Dependendo do canal escolhido, o meio se comportaria de maneira diferente; o efeito de desvanecimento e a presença de interferência, dependendo do canal, causa essa diferença. Isso poderia ser útil para simulação de protocolos que utilizam múltiplos canais.

O transceptor CC2420 utilizado nesse trabalho pode ser programado para operar em 16 canais diferentes. Portanto, a capacidade de utilizar múltiplos canais pode ser utilizada pelos protocolos MAC para aumentar o rendimento da rede. Embora ele use a técnica *Direct Sequence Spectrum* (DSSS), ou Espalhamento Espectral por Frequência Direta, estudos mostraram que a maioria dos canais sofre de uma significativa interferência de outros transmissores de 2,4 GHz [Dezfouli et al. 2015].

Para simular protocolos multicanais, é preciso antes que o simulador possua um modelo de simulação realista, capaz de capturar os efeitos de desvanecimento em larga escala, a atenuação em pequena escala, o sombreamento log-normal e as características não-estacionárias do canal em longos períodos de tempo.

A implementação padrão do modelo de camada física do Castalia para a simulação de RSSFs considera o uso de apenas um canal. Além disso, não considera as características não-estacionárias do canal durante períodos acima de algumas horas de simulação. Dessa

forma, foi preciso desenvolver um modelo de simulação de múltiplos canais e depois fazer uma adaptação no Castalia para incorporar esse novo modelo. Essa é uma das contribuições da dissertação.

Após o desenvolvimento desse modelo, utilizou-se uma instância do mesmo para cada um dos 16 canais do padrão IEEE 802.15.4, de modo a simular protocolos que utilizam múltiplos canais, com cada canal possuindo característica diferente, considerando o comportamento não-estacionário. Para cada canal foi definida uma característica diferente utilizando uma função que implementamos para o Castalia: *generatePowerFading(srcAddr, (int)(signalMsg->getCarrierFreq() - 2405)/5)*. Ela atualiza o valor da potência recebida para cada conexão utilizando a distribuição de Rice.

Por fim, foi utilizado o protocolo de múltiplos canais TMCP baseado em árvores, que faz uso de transmissões paralelas entre as mesmas em vários canais simultaneamente, de modo a aumentar a vazão da rede e diminuir os problemas de interferência e sincronização da rede. Em um cenário conturbado, cheio de objetos metálicos e móveis, que influenciam tanto o desvanecimento em larga escala como a atenuação em pequena escala, realizar troca de canais pode melhorar a qualidade de serviço da rede, uma vez que a rede se torna capaz de se adaptar às variações de qualidade de canal.

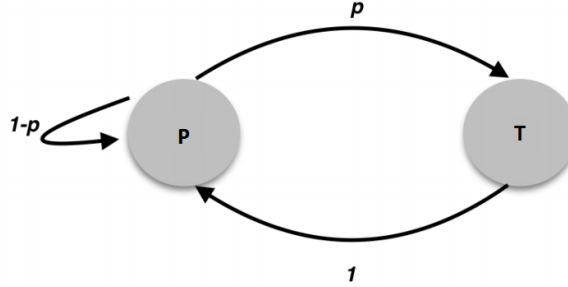
3.1 Modelo de Simulação para o Desvanecimento em Longos Períodos de Tempo

O canal sem fio pode ser modelado como estacionário no sentido amplo, por um período curto de tempo, apesar das movimentações ao redor do transmissor e do receptor. No entanto, as propriedades do canal podem se alterar significativamente em um período de poucas horas devido a mudanças ocorridas na topologia do ambiente. O movimento de objetos, metálicos ou não, pode causar modificações aleatórias no canal no decorrer do tempo, que não são levadas em consideração nas distribuições utilizadas para modelar o desvanecimento. Isso pode requerer novo cálculo dos parâmetros das distribuições, uma vez que esses parâmetros podem se tornar obsoletos no decorrer do tempo [Agrawal (2) et al. 2014].

Para modelar o efeito do desvanecimento no canal por um longo período de tempo utilizou-se uma abordagem baseada em uma cadeia de Markov de dois estados, ilustrada

na Fig. 3.1.

Figura 3.1: Processo de Markov de dois estados para modelar canal não-estacionário.



Enquanto a cadeia permanecer no estado definido como P, as características do canal permanecem inalteradas. A transição para o estado T ocorre com probabilidade p . Quando isso ocorre, os parâmetros dos modelos do canal são modificados de modo que haja uma mudança brusca nas características do canal. Ao realizar a troca dos parâmetros do canal, a cadeia retorna para o estado P com probabilidade igual a 1. O valor de p define a frequência com que mudanças nas características do canal ocorrem. Dessa forma, é possível simular ambientes que permanecem inalterados por muito tempo e ambientes que apresentam modificações na topologia com maior frequência.

Os valores de potência recebida por um nó são gerados a partir de alguma distribuição de probabilidade, como Rice. Considera-se também o desvanecimento em larga escala, obtido por meio da utilização do modelo log-distância e do sombreamento log-normal.

O primeiro passo para a redefinição das características do canal, quando a cadeia de Markov atinge o estado T, é a definição da potência de recepção média ($P_R(d, t)_{dBm}$) para uma distância d e no tempo t , de acordo com a Eq. 3.1.

$$P_R(d, t)_{dBm} = P_T - L(d). \quad (3.1)$$

P_T é a potência de transmissão e $L(d)$ possui uma parte determinística, relacionada apenas com a distância entre transmissor e receptor, e uma parte aleatória, referente ao sombreamento log-normal, de acordo com a Eq. 2.2.

Dessa forma, considerando uma mesma distância entre transmissor e receptor, o valor de $P_R(d, t)_{dBm}$ apresentará uma variação aleatória referente a $X\sigma$, pertencente à Eq. 2.2, que pode aumentar ou diminuir o valor médio de potência recebida.

Após a definição do valor resultante de potência média em dBm , o valor correspondente de potência é convertido em mW e utilizado como parâmetro para a distribuição de Rice, que é amostrada para definir o valor de potência recebida em cada instante.

Para modelar as variações no nível de atenuação em pequena escala, a variância da potência recebida também muda sempre que ocorrem mudanças nas características dos canais. Esse tipo de comportamento foi observado tanto em experimentos realizados nos nossos experimentos como em outros trabalhos [Agrawal et al. 2014]. O valor do desvio padrão $\sigma(t)$ da distribuição de Rice é, então, amostrado a partir de uma distribuição uniforme sempre que a cadeia de Markov atinge o estado T.

A Eq. 3.2 mostra a distribuição de probabilidade que modela o valor de potência recebida em cada instante, considerando o valor da potência média P_R e da variância $\sigma(t)$ no tempo t .

$$p_r(x) = \frac{x}{\sigma^2(t)} e^{-\frac{(x+P_R)}{2\sigma^2(t)}} I_0\left(\frac{P_R \cdot x}{\sigma^2(t)}\right), x \geq 0, \quad (3.2)$$

em que $P_R = 10^{\frac{P_R(d,t)_{dBm}}{10}}$ é a potência média em mW para uma distância d e no tempo t .

Para testar o modelo de simulação, inicialmente foram gerados valores aleatórios seguindo a distribuição de Rice utilizando a biblioteca IT++⁶.

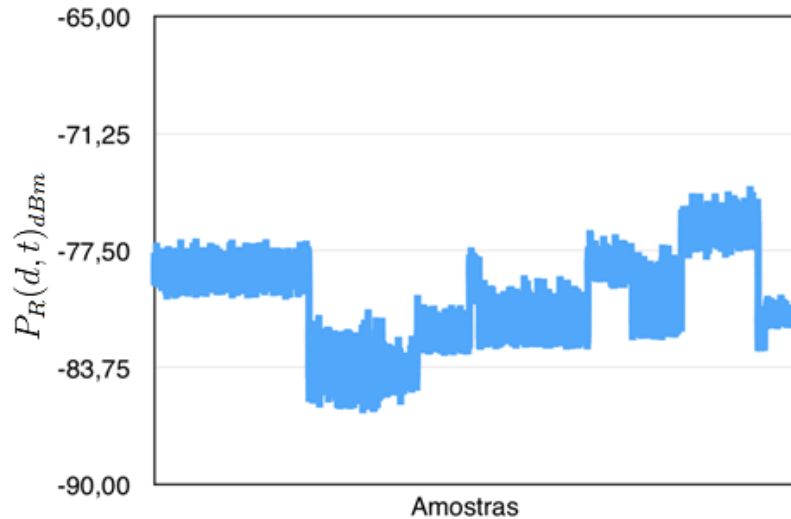
A IT++ é uma biblioteca de classes e funções matemáticas, de processamento de sinal e de comunicação, implementada em C++, *open-source*, cuja principal função é a simulação de sistemas de comunicação e a realização de pesquisas nessa área. Seu núcleo a torna similar ao MATLAB, GNU Octave e ao SciPy. A biblioteca simula valores de potência recebidos em um receptor, considerando uma distância d de 20 metros entre os nós, e os valores de $n = 1,69$, $d_0 = 15$ m, $L(d_0) = 80,48$ dB e $\sigma = 8,13$ dB, que são aplicados no cálculo do desvanecimento em larga escala e do sombreamento, utilizando o modelo de sombreamento log-normal (Eq. 2.2). Esses valores foram obtidos a partir de experimentos em ambiente industrial descritos em [Tanghe et al. 2008].

O resultado da simulação utilizando a biblioteca está ilustrado na Fig. 3.2. Na figura são mostradas 10.000 amostras. O valor de $\sigma(t)$ foi amostrado partir de uma distribuição uniforme entre 0 e 1,5. O valor de p foi definido em 0,1%, pois nas simulações verificamos que essa probabilidade era a mais adequada para simular os efeitos causados nos cenários

⁶<http://itpp.sourceforge.net/> - Acessado em 03/02/2016.

utilizados nesse trabalho. Por exemplo, ao considerar um intervalo de cinco segundos entre cada amostragem da cadeia de Markov, essa probabilidade representa uma modificação das características dos canais a cada 5000 segundos (1,4 horas), em média.

Figura 3.2: RSSI simulado com o modelo proposto utilizando a biblioteca IT++.



Pode-se observar que o modelo é capaz de capturar o comportamento do canal em longos períodos de tempo, considerando o comportamento não-estacionário do canal sem fio em ambientes industriais. Os resultados obtidos pelo modelo são compatíveis com resultados de experimentos realizados em ambiente industrial e descritos em artigos que fizeram análise do comportamento do canal em longos períodos de tempo [Agrawal et al. 2014] [Agrawal (2) et al. 2014] [Wang e Yao 2013], como ilustrado na Fig. 2.4.

3.1.1 Integração do modelo com o Castalia

Após a validação inicial do modelo utilizando a biblioteca IT++, realizou-se a integração com o simulador Castalia, que implementa a camada de acesso ao meio do padrão IEEE 802.15.4. Na camada física, são usados parâmetros de rádios IEEE 802.15.4 reais e um modelo de camada física baseado no modelo de sombreamento log-normal. O simulador também permite variação temporal na potência recebida pelos nós, a partir de amostras que simulam o comportamento de um canal sujeito a atenuação por multipercurso. No entanto, o modelo implementado pelo simulador considera que o canal é estacionário; ou seja, suas características permanecem inalteradas no decorrer do tempo.

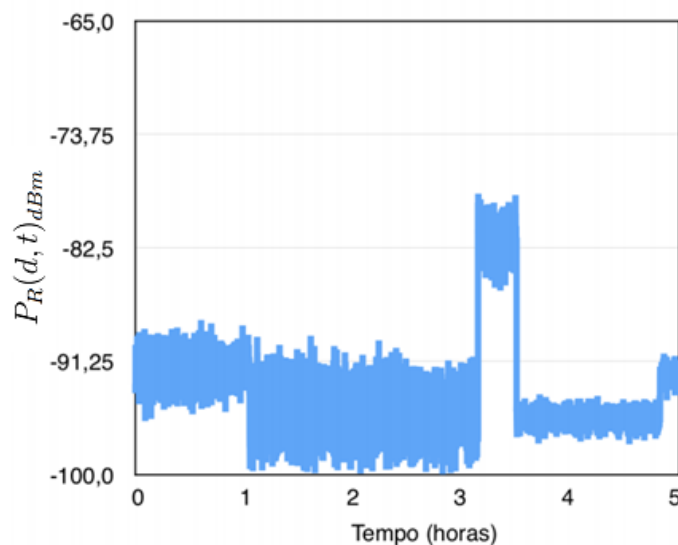
Tabela 3.1: Parâmetros utilizados na simulação.

Parâmetros de Simulação	
Distância entre os nós	70 metros
Camada física e MAC	IEEE 802.15.4
Tempo de Simulação	18000 s (5 horas)
Potência de transmissão	0 dBm
Taxa de transmissão de pacotes	0,2 pacotes/s
Probabilidade de transição (p)	0,1%

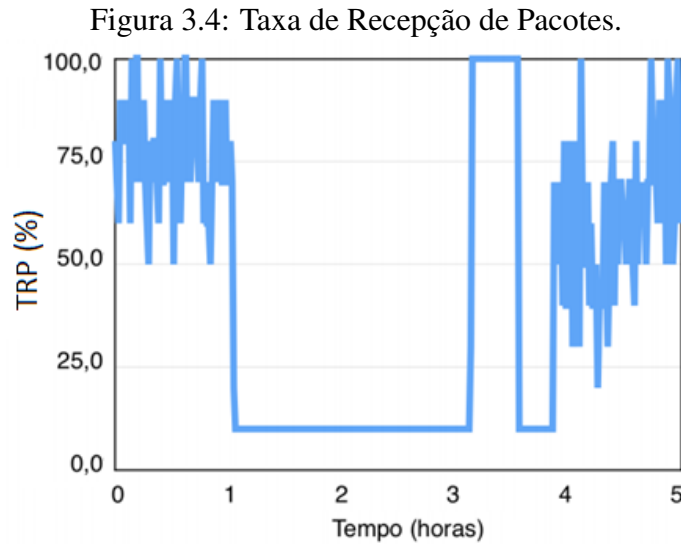
Para realizar a integração, foram realizadas modificações na classe *WirelessChannel*, para capturar as variações temporais nas características dos canais em um longo período de tempo. Após a integração do modelo, foi realizada uma simulação considerando um nó transmissor e um nó receptor. Os parâmetros da simulação são detalhados na Tab. 3.1. Os parâmetros para o modelo log-distância e sombreamento log-normal são os mesmos que foram descritos em [Tanghe et al. 2008].

A Fig. 3.3 mostra a potência recebida no nó durante as cinco horas de simulação. É possível observar os momentos em que ocorrem variações nas características dos canais, tanto com relação ao valor médio de potência, como com relação à severidade da atenuação por multipercurso.

Figura 3.3: Potência de Recepção Média simulada com o Castalia.



A Fig. 3.4 mostra a Taxa de Recepção de Pacotes (TRP) durante as cinco horas. É possível observar uma correlação entre a TRP com as características do canal em cada momento (Fig. 3.3). Esse tipo de comportamento é realista em ambientes dinâmicos, que provocam variações na qualidade dos canais no decorrer do tempo, como é o caso de ambientes industriais.



Como os canais em ambientes industriais são descorrelacionados em frequência, para simular protocolos que utilizam múltiplos canais pode-se utilizar uma instância do modelo para cada canal disponível para comunicação. Nesse cenário, realizar troca de canais pode melhorar a qualidade de serviço da rede, uma vez que a rede se torna capaz de se adaptar às variações de qualidade de canal. Para permitir a avaliação de protocolos multicanais com o Castalia se faz necessário realizar modificações na interface entre o módulo que implementa o modelo do canal sem fio com os módulos que implementam os protocolos de camadas superiores.

O Código 3.1 mostra o cálculo realizado na classe *WirelessChannel* que identifica o sombreamento log-normal no Castalia, definido na Eq. 2.2.

Código Fonte 3.1: Cálculo do sombreamento log-normal no módulo *WirelessChannel.cc*.

```

1  /* if the distance is very small (arbitrarily: smaller than one tenth
2  * of the reference distance) then make the path loss 0dB */
3  if (dist < d0/10.0) {
4      PLd = 0;
5      bidirectionalPathLossJitter = 0;
6  }
```

```

7  else {
8    PLd = PLd0 + 10.0 * pathLossExponent * log10(dist / d0) + shadowing;
9    trace() << "Atualiza pL: dist->" << dist << " de->" << src << " a->" << dst << ":" << PLd;
10   bidirectionalPathLossJitter = normal(0, bidirectionalSigma) / 2;
11 }

```

O Código 3.2 mostra a função *updateChannel* que criamos e que realiza a atualização das características do canal com probabilidade p de mudança. Essa alteração é também feita no módulo *WirelessChannel*.

Código Fonte 3.2: Função que atualiza as características do canal com probabilidade p de mudança.

```

1  /* p: probabilidade de mudanca das caracteristicas do canal */
2  bool WirelessChannel::updateChannel(double p) {
3
4    bool changed = false;
5    int x;
6    double shad=0;
7    double shadowing=0;
8    static bool pri = false;
9
10   std::random_device rd;
11   std::normal_distribution<> gaussian(0,sigma);
12   std::mt19937 gen(rd());
13
14   x = rand() % ((int)(100.0/p));
15   trace() << "Valor sorteado: x=" << x;
16   if(!pri || x == 0 || simTime() == 0) {
17     pri = true;
18     trace() << "Muda parametros...";
19     // muda as caracteristicas dos canais
20     for(int c = 0; c < 16; c++) { //para cada canal
21       for (int i = 0; i < numOfSpaceCells; i++) {
22         for (int j = 0; j < numOfSpaceCells; j++) {
23           while(true) {
24             shad = std::round(gaussian(gen));
25             if(shad >= -40 && shad <= 40) break;
26           }
27           varrice[16*(j+i*numOfSpaceCells)+c] = rand()%15 + 1;
28           varrice[16*(j+i*numOfSpaceCells)+c] /= 10.0;
29           meanrice[16*(j+i*numOfSpaceCells)+c] = pow(10,shad/10);
30           trace() << "Sombreamento de->" << i << " para->" << j << " no
31             canal->" << c << ":" << shad;
32           changed = true;
33         }
34       }
35     }
36   }
37 }

```

```

33         }
34     }
35 }
36 return changed;
37 }

```

O Código 3.3 chama a função *updateChannel* que realiza a atualização das características do canal quando um nó envia um sinal de início de transmissão para outro nó no módulo *WirelessChannel*.

Código Fonte 3.3: Chamada da função *updateChannel*.

```

1 case WC_SIGNAL_START:{
2     WirelessChannelSignalBegin *signalMsg = check_and_cast <WirelessChannelSignalBegin*>(msg);
3     updateChannel(0.01); /* executa cadeia de markov para mudar parametros da distribuicao */
4     int srcAddr = signalMsg->getNodeID();
5     int receptioncount = 0;
6     /* atualiza valor de potencia recebida para cada conexao usando distribuicao de Rice */
7     generatePowerFading(srcAddr,(int)(signalMsg->getCarrierFreq()-2405)/5);
8     /* Find the cell that the transmitting node resides */
9     int cellTx = nodeLocation[srcAddr].cell;
10 ...

```

3.2 Protocolo de múltiplos canais TMCP

Proposto por [Wu et al. 2008] e com pseudo-código na Seção 3.2, o protocolo TMCP possui três componentes: Detecção do Canal (CD), Estabelecimento de Canal (CA), e Comunicação de Dados (DC). O módulo CD encontra canais ortogonais disponíveis que podem ser utilizados no ambiente. Para isso, dois nós são utilizados para amostrar a qualidade de cada canal transmitindo pacotes entre si, e então entre todos os canais com boa qualidade, sendo selecionados canais não-adjacentes.

Dados k canais ortogonais, o módulo CA particiona a rede em k sub-árvores e estabelece um canal para cada sub-árvore, explorando as transmissões paralelas entre elas. A desvantagem é que a interferência intra-árvore não pode ser evitada completamente, mas ainda assim, ao particionar a rede, a interferência entre os nós da rede como um todo diminui, melhorando a qualidade das transmissões.

Depois de atribuir os canais, o módulo DC gerencia a coleta de dados em cada sub-árvore. Quando um nó deseja enviar um pacote ao nó sorvedouro, ele utiliza a sub-árvore

à qual pertence. Nesse protocolo, assume-se que o nó sorvedouro possui uma interface de rede para cada canal.

Baseando-nos no pseudo-código, criamos um código que implementa o protocolo TMCP. Ele foi escrito de modo a criar as expressões exatas de localização (coordenadas) dos nós e a atribuição de canais utilizados pelo *script* de simulação do Castalia *omnetpp.ini*. Na sua execução, são pedidas informações como quantidade de nós e a localização X, Y e Z dos nós. O pseudo-código do protocolo está descrito a seguir:

Algoritmo 1: PSEUDO-CÓDIGO DO TMCP.

Entrada: k canais, um grafo $G = (V, E)$, uma raiz r e a interferencia atribuida de cada no

Saída: Para cada no u , c_u e p_u utilize o algoritmo Busca em Largura (BFS) para construir árvore com raiz em r

```

1  início
2      para cada canal  $i$  faça
3           $T_i = r$ ;
4      fim
5      para cada no  $u$  faça
6           $c_u = 0$ ;  $p_u = null$ ;
7      fim
8      nivel = 1; repita
9          lista_de_nos =  $\{u | altura(u) == nivel; c_u == 0\}$ ;
10         ordena lista_de_nos em ordem crescente do numero de pais do no;
11         para cada no  $u$  na lista_de_nos faça
12             encontre  $T_i$  que se mantem conectado e possui a menor interferencia apos
13             adicionar  $u$ ;
14              $T_i = T_i \cup \{u\}$ ;  $c_u = i$ ;  $p_u = v$ , que se conecta a  $u$  e tem a menor
15             interferencia entre todos os nos em  $T_i$ ;
16             atualize valor da interferencia de  $T_i$ ;
17         fim
18         nivel++
19 até nivel > altura maxima da arvore;
20 fim

```

No exemplo da execução do protocolo, ilustrado na Fig. 3.5, os dados inseridos pelo usuário estão destacados como "ENTRADA". As demais linhas são resultado da execução do protocolo. A primeira linha de entrada com valor 7 corresponde à quantidade de nós que serão utilizados. A segunda linha indica a localização X, Y e Z (0 0 0) do nó sorvedouro. As linhas seguintes (3 a 8) dos dados de entrada correspondem à localização dos outros seis

Figura 3.5: Dados de entrada para o algoritmo TMCP e resultados.

```

7
0 0 0
20 0 0
0 20 0
20 20 0
0 40 0
40 0 0
40 40 0

```

⇒ ENTRADA

```

Defining channel for node 2
Analyzing tree 0
Interference in the tree: 1
Analyzing tree 1
Interference in the tree: 1
Analyzing tree 2
Interference in the tree: 1
Analyzing tree 3
Interference in the tree: 1
Interference of 0: 0
Node 2 with parent: 0 and channel: 0
Defining channel for node 3
Analyzing tree 0
Interference in the tree: 2
Analyzing tree 1
Interference in the tree: 1
Analyzing tree 2
Interference in the tree: 1
Analyzing tree 3
Interference in the tree: 1
Interference of 0: 1
Node 3 with parent: 0 and channel: 1
Defining channel for node 1
Analyzing tree 0
Interference in the tree: 2
Analyzing tree 1
Interference in the tree: 2
Analyzing tree 2
Interference in the tree: 1
Analyzing tree 3
Interference of 2: 0
Node 4 with parent: 2 and channel: 0
0 0 3

Next hop: 0
Channel: -1
1 1 1
0
Next hop: 0
Channel: 2
2 1 1
0
Next hop: 0
Channel: 0
3 1 1
0
Next hop: 0
Channel: 1
4 2 0
2 3
Next hop: 2
Channel: 0
5 2 0
1 3
Next hop: 1
Channel: 2
6 2 0
3
Next hop: 3
Channel: 1
# Coordenadas dos nos
SN.node[0].xCoord = 0
SN.node[0].yCoord = 0
SN.node[0].zCoord = 0

SN.node[1].xCoord = 20
SN.node[1].yCoord = 0
SN.node[1].zCoord = 0

SN.node[2].xCoord = 20
SN.node[2].yCoord = 0
SN.node[2].zCoord = 0

SN.node[3].xCoord = 20
SN.node[3].yCoord = 20
SN.node[3].zCoord = 0

SN.node[4].xCoord = 0
SN.node[4].yCoord = 40
SN.node[4].zCoord = 0

SN.node[5].xCoord = 40
SN.node[5].yCoord = 0
SN.node[5].zCoord = 0

SN.node[6].xCoord = 40
SN.node[6].yCoord = 40
SN.node[6].zCoord = 0

# Frequencia (canal) dos nos
SN.node[0].Communication.Radio.carrierFreq = 2400
SN.node[1].Communication.Radio.carrierFreq = 2415
SN.node[2].Communication.Radio.carrierFreq = 2405
SN.node[3].Communication.Radio.carrierFreq = 2410
SN.node[4].Communication.Radio.carrierFreq = 2405
SN.node[5].Communication.Radio.carrierFreq = 2415
SN.node[6].Communication.Radio.carrierFreq = 2410

```

nós.

Os dados de entrada para o protocolo TMCP (oito primeiras linhas em destaque) são definidos de modo estático, antes da simulação. Porém, para validar o modelo de simulação nos experimentos descritos em 4.2.1, a posição dos nós foram definidas de modo aleatório, exceto para o nó sorvedouro, que sempre fica no centro do ambiente (é possível inseri-lo em qualquer posição). A posição dos demais nós é definida através de um programa simples que gera valores aleatórios com distância limite, de modo que todos os nós possam se comunicar, senão com o sorvedouro, pelo menos com o nó mais próximo que esteja no mesmo canal.

Após a execução do programa utilizando distâncias aleatórias, o algoritmo indica, para cada nó, o seu pai e o canal no qual cada um irá trabalhar, ilustrados nas demais linhas da Fig. 3.5. O resultado da execução permitiu definir, por exemplo, as seguintes características da rede utilizando a linguagem de simulação do Castalia:

Código Fonte 3.4: Resultado da execução do algoritmo TMCP com dados compilados.

```

1 SN.node[0].Application.nextRecipient = "0"
2 SN.node[1].Application.nextRecipient = "1"
3 ...
4 SN.node[7].Application.nextRecipient = "7"
5 SN.node[8].Application.nextRecipient = "2"
6 SN.node[9].Application.nextRecipient = "1"
7 SN.node[10].Application.nextRecipient = "2"
8
9 SN.node[12].Application.paiNext = "15"
10 SN.node[22].Application.paiNext = "18"
11 SN.node[24].Application.paiNext = "14"
12
13 SN.node[12].Application.fonte = 12
14 SN.node[22].Application.fonte = 22
15 SN.node[24].Application.fonte = 24
16
17 SN.node[0].Application.isSink = true
18 SN.node[1].Application.isSink = true
19 SN.node[2].Application.isSink = true
20 ...
21 # Frequencia (canal) dos sink nodes
22 SN.node[0].Communication.Radio.carrierFreq = 2405
23 SN.node[1].Communication.Radio.carrierFreq = 2410
24 ...
25 SN.node[7].Communication.Radio.carrierFreq = 2440
26 # Frequencia (canal) dos end nodes/routers
27 SN.node[8].Communication.Radio.carrierFreq = 2415

```

```

28 SN.node[9].Communication.Radio.carrierFreq = 2410
29 SN.node[10].Communication.Radio.carrierFreq = 2415
30 ...
31 # Coordenadas dos sink nodes
32 SN.node[0].xCoor = 100
33 SN.node[0].yCoor = 100
34 SN.node[0].zCoor = 1
35
36 SN.node[1].xCoor = 100
37 SN.node[1].yCoor = 100
38 SN.node[1].zCoor = 1
39 ...
40 # Coordenadas dos end nodes/routers
41 SN.node[8].xCoor = 92
42 SN.node[8].yCoor = 171
43 SN.node[8].zCoor = 1
44
45 SN.node[9].xCoor = 53
46 SN.node[9].yCoor = 186
47 SN.node[9].zCoor = 1
48 ...

```

No exemplo do Código Fonte 3.4 foram inseridos 25 nós (*node[0..24]*) com os oito primeiros sendo nós sorvedouros (*node[0..7]*). Da primeira à quarta linhas são definidos os pais dos nós (o pai de um sorvedouro é ele mesmo). Alguns nós possuem como pai seu sorvedouro (conexão direta entre *nó folha* -> *nó sorvedouro*, por exemplo, nas linhas 5-7), outros possuem como pai outros nós (Ex. linhas 9-11), os quais irão repassar (trabalham como roteadores - FFD - mas também enviam seus próprios pacotes) ao seu sorvedouro.

De modo a facilitar a manipulação do pacote pelo Castalia quando ele é originado de nós a dois saltos (nós folhas), foi definido o parâmetro "fonte", com o identificador do nó de origem (linhas 13-15). Também para facilitar o roteamento dos pacotes para os nós sorvedouros, foi necessário definir o parâmetro "isSink"(linhas 17-19) em cada sorvedouro.

As linhas 9-19 (*paiNext*, *fonte*, *isSink*) não foram criadas na execução do algoritmo TMCP, mas tão somente foram definidas para facilitar a simulação e roteamento correto dos pacotes, baseados nas informações geradas pelo algoritmo.

As linhas 22-48 do algoritmo geram a frequência para os sorvedouros e demais nós (na banda de 2,4 GHz), seguida das coordenadas de cada um (linhas 32 em diante). Como no algoritmo TMCP há um único nó sorvedouro com várias interfaces de rede, cada uma traba-

lhando em um canal diferente, e o Castalia não implementa nós com várias interfaces, senão apenas uma, foi necessário definir vários sorvedouros na mesma posição (100x100x1), dando a impressão de um único nó com várias interfaces. Como cada sorvedouro trabalha em um canal diferente, não há problemas de interferência entre eles, o que valida a implementação desse ambiente.

No exemplo, os sorvedouros se encontram no centro do ambiente com 200 m x 200 m de tamanho. No ambiente com 400 m x 400 m os sorvedouros iriam se manter nas coordenadas 200x200x1, sendo este último valor a altura do nó em relação ao solo, em metros.

Capítulo 4

Resultados

Nesse capítulo são apresentados os resultados obtidos através de experimentos realizados na indústria e das adaptações no modelo de canal do simulador Castalia para um modelo realista que simule os efeitos causados no canal de transmissão em ambientes industriais. Na Seção 4.1 é mostrado um estudo sobre o perfil não-estacionário dos canais durante longos períodos de tempo dentro da indústria. Na Seção 4.2 é mostrada a aplicação do protocolo TMCP na criação da estrutura da rede a ser simulada. Na Seção 4.3 são realizados experimentos simulando seis ambientes utilizando ambos os modelos com perfil estacionário e não-estacionário.

4.1 Perfil não-estacionário do canal

Em um experimento realizado no ambiente industrial localizado em Campina Grande/PB, foram utilizados dois rádios IEEE 802.15.4 operando na faixa de 2,4 GHz, comunicando-se entre si, com um nó enviando pacotes ao nó sorvedouro, como ilustrado na Fig. 4.1.

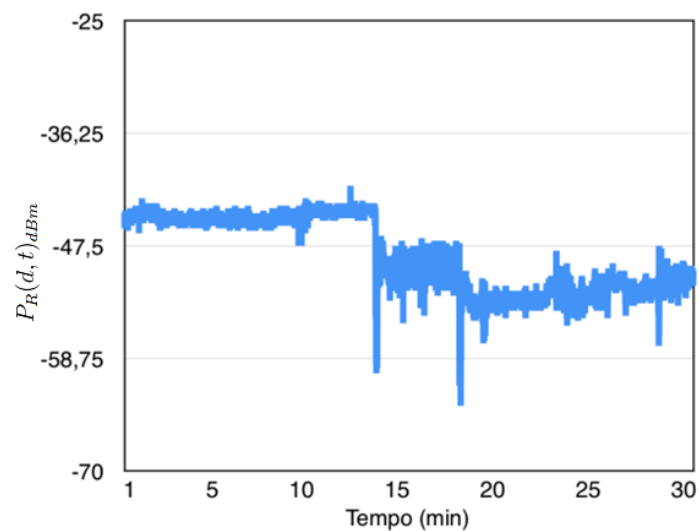
Durante 30 minutos, os dois nós permaneceram estáticos se comunicando no ambiente industrial. No gráfico da Fig. 4.2 são mostrados valores de RSSI obtidos durante esse período. Nele pode ser observada a potência de recepção dos pacotes obtidos pelo nó sorvedouro. Em determinado momento (no exemplo, após os primeiros 15 minutos, aproximadamente) é observada uma modificação brusca nas características do canal. Na primeira metade do experimento, o perfil do canal permaneceu praticamente o mesmo, mas na segunda metade a qualidade de transmissão diminuiu significativamente. Em comparação com o experimento

Figura 4.1: Disposição dos nós dentro da indústria.



realizado por [Agrawal et al. 2014], ilustrado na Fig. 2.4, essa mudança brusca indica que, para um grande intervalo de tempo, o canal se comporta de maneira não-estacionária.

Figura 4.2: Comportamento não-estacionário do canal em experimento realizado na indústria.

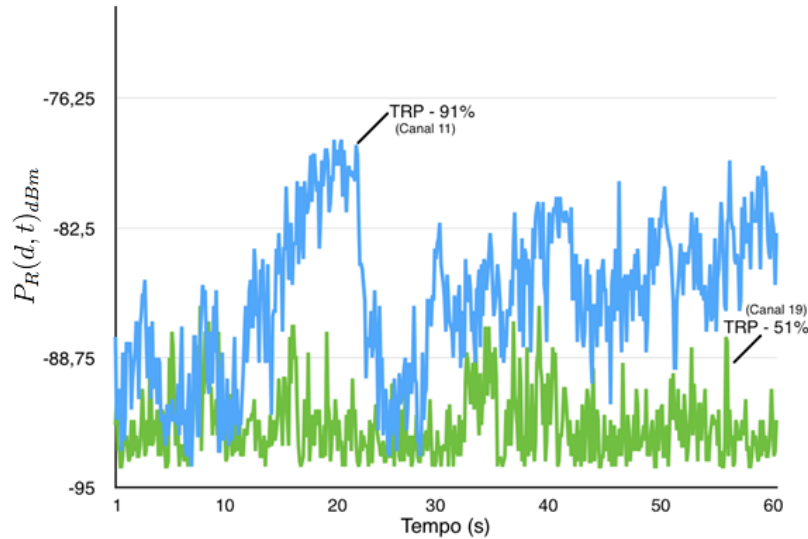


Em outros experimentos realizados no mesmo ambiente, os dois nós foram inseridos no canal 11 e se comunicaram por 60 segundos. Após esse período, modificamos o canal de transmissão para o 19 e nele permaneceram também durante 60 segundos.

O resultado, ilustrado na Fig. 4.3, mostra as diferenças notáveis nas características dos

canais. Essas diferenças podem ser observadas devido ao perfil de multipercurso do ambiente, uma vez que os canais são descorrelacionados em frequência e o impacto do multipercurso é diferente em canais diferentes.

Figura 4.3: Experimento que mostra canais descorrelacionados.



Na Fig. 4.3 a linha em azul identifica o canal 11, que teve TRP de 91%, e de verde o canal 19 é identificado, o qual obteve TRP de 51%. No mesmo ambiente, com a mesma topologia, na mesma distância, durante intervalos de tempo de mesma duração, os canais de comunicação demonstraram ter características diferentes.

Uma solução para amenizar os efeitos causados por esse problema é a alteração do canal de transmissão de modo a manter a TRP a um nível aceitável, dependendo da aplicação. Nós que se comunicam em um canal, ao perceberem redução na sua qualidade, podem alterá-lo. Para testar a funcionalidade de protocolos que implementam múltiplos canais, se faz necessário a utilização de simuladores de redes.

4.2 Simulação com o protocolo de múltiplos canais TMCP

Após a implementação do modelo de canal que simula seu perfil não-estacionário durante longos períodos, foram realizados experimentos de modo a testar com o Castalia o protocolo de múltiplos canais TMCP (descrito na Seção 3.2) em ambientes simulados com dois, quatro e oito canais, utilizando vários nós criados aleatoriamente.

Para isso foi desenvolvido um código para gerar aleatoriamente topologias de vários nós, com o TMCP definindo as rotas e os canais associados a cada nó. Considerando que o Castalia não provê nós com várias interfaces de rede, em que cada uma iria trabalhar em um canal diferente, foram criados e posicionados vários nós sorvedouros no mesmo local, cada um trabalhando em uma frequência diferente.

A fim de comparar e validar o modelo de canal com perfil não-estacionário, também foram realizados experimentos para os mesmos ambientes com o modelo de canal estacionário padrão do Castalia. Ao todo foram simulados 12 ambientes, sendo 6 utilizando o modelo proposto, e os outros 6 com o modelo padrão do Castalia. Dos 6 primeiros no modelo proposto, foram simulados ambientes com 2, 4 e 8 canais em áreas de 200x200 metros e de 400x400 metros. Para comparar o modelo proposto com o modelo padrão, foram utilizados os mesmos ambientes com o modelo padrão.

4.2.1 Definição do experimento de simulação

Para testar o protocolo de múltiplos canais TMCP, foram criados seis ambientes, os três primeiros com 200x200 metros de área e os demais com 400x400 metros. As Tabelas B.1 e B.2 ilustram as posições dos nós nos ambientes com 200x200 metros. As Tabelas B.5, B.6, B.7 e B.8 ilustram as posições dos nós nos ambientes com 400x400 metros de área. De modo a diminuir o tamanho das tabelas, considere-se que os nós estão a um metro de altura do solo, sendo a altura definida no Castalia da seguinte forma: $SN.node[ID].zCoor = 1$.

O primeiro ambiente com 200x200 metros de área foi definido com dois canais e 21 nós, sendo dois sorvedouros localizados no centro do ambiente, representando um único sorvedouro com duas interfaces de rede, cada uma em um canal. O mesmo acontece para os demais ambientes em relação ao nó sorvedouro. No segundo há 23 nós, sendo quatro sorvedouros e quatro canais. No terceiro ambiente há 27 nós com oito canais e oito sorvedouros.

Em relação aos ambientes com 400x400 metros de área, o primeiro possui 51 nós com dois canais, o segundo possui 53 nós com quatro canais e, por fim, o terceiro possui 57 nós com oito canais. Todos os cenários estudados estão listados na Tab. 4.1.

A Fig. 4.4 ilustra as posições reais dos nós geradas aleatoriamente, com as frequências e roteadores atribuídos pelo TMCP. Nesse exemplo, os nós 22, 12 e 24, por estarem muito distantes do sorvedouro, a eles foram atribuídos roteadores na mesma frequência de modo

Tabela 4.1: Ambientes simulados utilizando modelo não-estacionário e o protocolo TMCP

Área	Cenário	Quantidade Canais	Quantidade Nós	Quantidade Nós sorvedouros
200x200 m	A	2	21	2
	B	4	23	4
	C	8	27	8
400x400 m	D	2	51	2
	E	4	53	4
	F	8	57	8

que pudessem enviar os pacotes ao sorvedouro.

Nas Tabelas B.3 e B.4 estão ilustrados os nós folhas em cada ambiente com seus respectivos pais (roteadores) a dois saltos. Além da frequência, o TMCP também define as rotas de cada nó, decidindo assim qual é o pai de determinado nó.

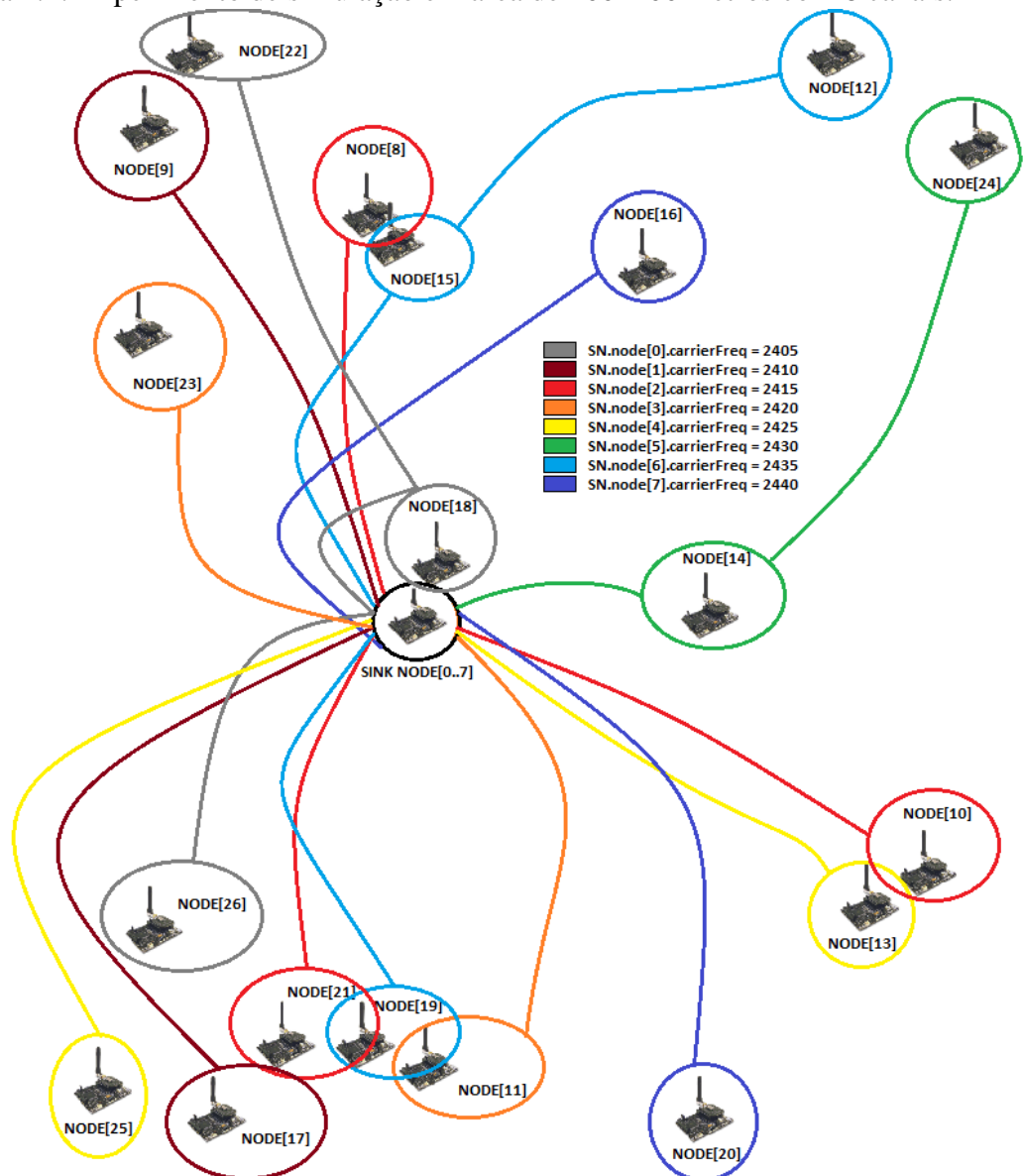
Para que o TMCP funcionasse corretamente, inicialmente foi necessário encontrar o alcance máximo entre dois nós de modo que conseguissem se comunicar. Após a análise, foi definido que a distância limite entre dois nós era 74 metros a 0 dBm de potência de transmissão. Aos 75 metros o nó sorvedouro deixou de receber os pacotes do nó folha. Mesmo que um nó estivesse muito distante do sorvedouro, se ele possuíse um nó pai que pudesse repassar seus pacotes, o nó folha e o sorvedouro poderiam se comunicar. Baseando-se nesse valor, como há muitos nós na rede e a comunicação entre eles também atrapalha na sua comunicação com os outros, foi estabelecido para o protocolo TMCP um alcance máximo de 30 metros entre os nós comunicantes.

4.3 Simulações numéricas

4.3.1 Plano de Simulação

O experimento consistiu da simulação dos 12 ambientes utilizando o modelo de canal que simula o perfil não-estacionário dos canais e dos mesmos ambientes com o perfil estacionário padrão. Como os ambientes simulados possuem vários nós, foram escolhidos alguns para

Figura 4.4: Experimento de simulação em área de 200x200 metros com 8 canais.



ilustrar os resultados da simulação.

Tendo como exemplo do Ambiente 3 (200x200 metros com 8 canais), ilustrado na Fig. 4.4 e nas Tab. B.1 para localização e Tab. B.3 para definição dos nós finais e roteadores, foram realizadas várias simulações nas quais foram obtidos resultados satisfatórios da aplicação do modelo e da utilização do protocolo TMCP nos ambientes da rede.

4.3.2 Execução

Para executar a simulação, é necessário estabelecer o ambiente a ser simulado, definido no arquivo *omnetpp.ini*. O *script* de execução da simulação pode ser compilado da seguinte maneira:

Código Fonte 4.1: Exemplo de execução da simulação pelo Castalia.

```
1 Castalia -c General,debugALL,camadaTunableMAC -i omnetpp.ini -o analisar.txt
```

O parâmetro *-c* indica que a simulação será compilada com os parâmetros *General,debugALL,camadaTunableMAC*. O *General* indica uma estrutura padrão da simulação (características gerais do ambiente, como tempo de simulação, tamanho da área, número de nós, coordenadas dos nós e algumas características do modelo de canal de sombreamento log-normal, além dos parâmetros gerados pelo TMCP). O parâmetro *debugALL* indica quais informações serão geradas pelo Castalia de modo a fazer uma auditoria mais detalhada sobre, por exemplo, a quantidade de pacotes que cada nó envia e recebe, a potência de recepção, quais informações estão sendo passadas no módulo de roteamento, entre outras. Com valor booleano, é possível controlar quais módulos serão monitorados. O parâmetro *camadaTunableMAC* indica o módulo da camada MAC que foi utilizado para simulação. O parâmetro *-i* indica o arquivo de configuração da simulação que será utilizado pelo Castalia e o *-o* é o arquivo criado e que gera informações como a quantidade de pacotes recebidos pelos sorvedouros, a quantidade de energia gasta pelos nós, entre outras informações [Castalia User's Manual].

4.3.3 Análise das Simulações

Um conjunto de simulações foi executado utilizando o simulador Castalia e aplicando os modelos de Rice e do sombreamento log-normal. Foram capturados a quantidade de pacotes

recebidos pelos nós sorvedouros e comparados os resultados entre os dois modelos de canal.

Iniciando a análise pelo exemplo da Fig. 4.4, é possível notar que os nós muito próximos entre si foram definidos com canais (cores) diferentes de modo que não haja interferência entre eles. Existe um cálculo no TMCP que tenta minimizar essa interferência (define a melhor árvore). Os canais foram definidos pelo protocolo TMCP após a atribuição aleatória da posição dos nós, realizada por um código escrito em C++.

Um detalhe para o *node[12]* é que ele é um nó folha e possui um pai, o *node[15]*. Mesmo estando muito distante do seu nó sorvedouro, o pacote enviado por ele consegue chegar ao destino passando pelo *node[15]*, embora com muitas perdas. O *node[15]*, por sua vez, mais próximo do sorvedouro, consegue uma TRP maior. O mesmo acontece na simulação com o modelo estacionário padrão.

De modo a comparar a potência de recepção dos pacotes no nó sorvedouro e a quantidade de pacotes recebidos (%) em uma amostra de 40 segundos e ambientes com 2, 4 e 8 canais, foram escolhidos alguns nós aleatoriamente para análise. Como nomenclatura das figuras, entende-se por "MARKOV" o modelo de canal proposto nesse trabalho, e "PADRÃO", como o modelo estacionário padrão do Castalia. Pelo termo "200 m" entende-se por a área de simulação de 200x200 metros, e "400 m" por área de 400x400 metros. A potência de recepção é calculada em valores de dBm.

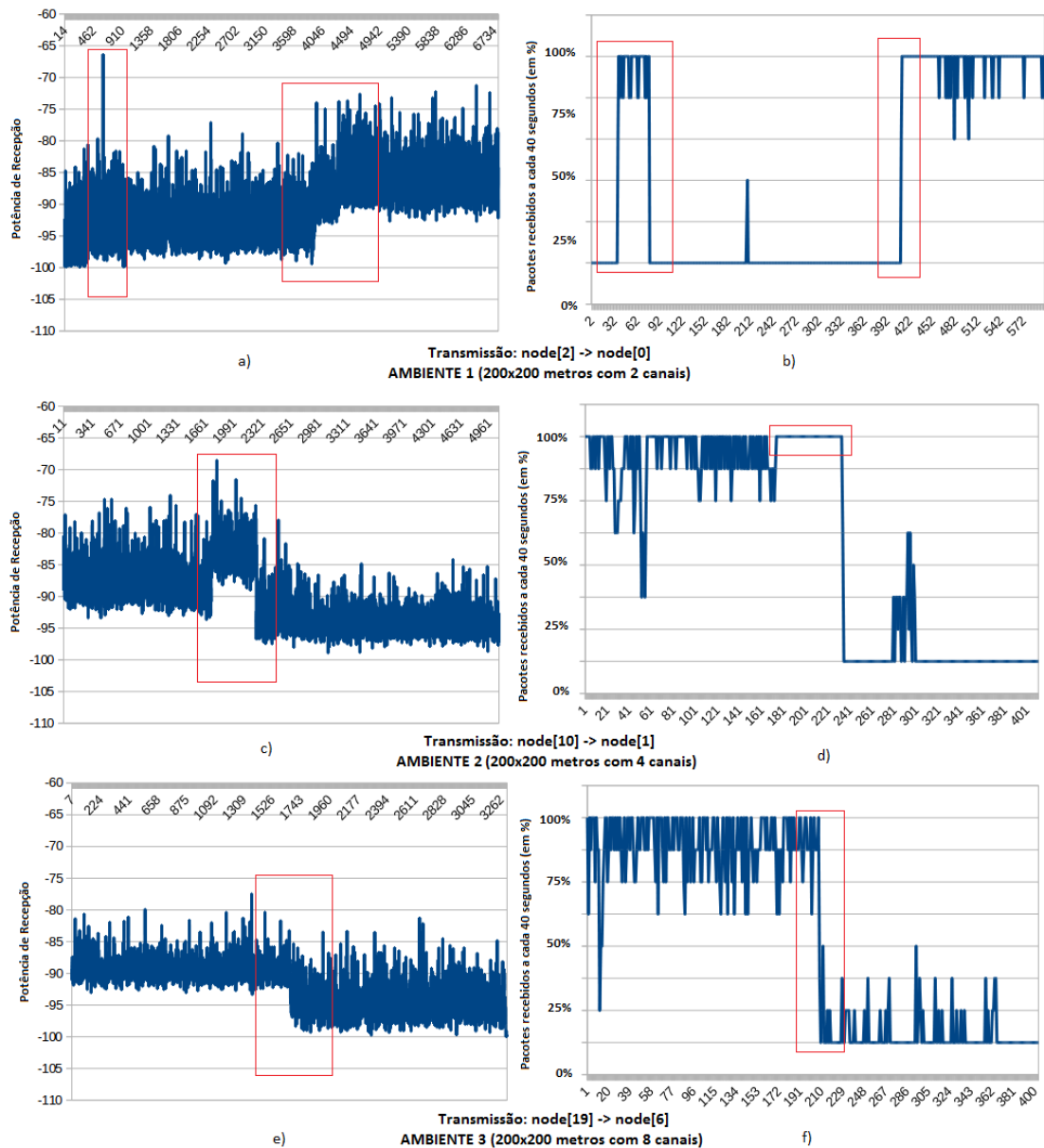
O primeiro gráfico da Fig. 4.5 a) ilustra a potência de recepção no sorvedouro *node[0]* enviada pelo *node[2]* durante cinco horas de simulação. É possível observar os momentos em que ocorrem variações nas características dos canais, tanto com relação ao valor médio de potência, como em relação à severidade da atenuação por multipercurso. Destacadas em vermelho estão identificadas as alterações nas características do canal. Paralelamente à potência de recepção, é ilustrada no gráfico b) a TRP no sorvedouro com origem do *node[2]*. É possível observar que existe uma correlação entre a TRP e as características do canal em cada momento.

No segundo gráfico da Fig. 4.5 nota-se que aproximadamente na primeira hora do experimento o canal permaneceu com boa qualidade, com quase todos os pacotes sendo recebidos. Logo após, destacado em vermelho, o canal sofre uma significativa melhora, o que permitiu a recepção de 100% dos pacotes. Aproximadamente na metade da simulação o canal sofre uma piora brusca na qualidade e a TRP cai para próximo de 10% e assim permanece até o

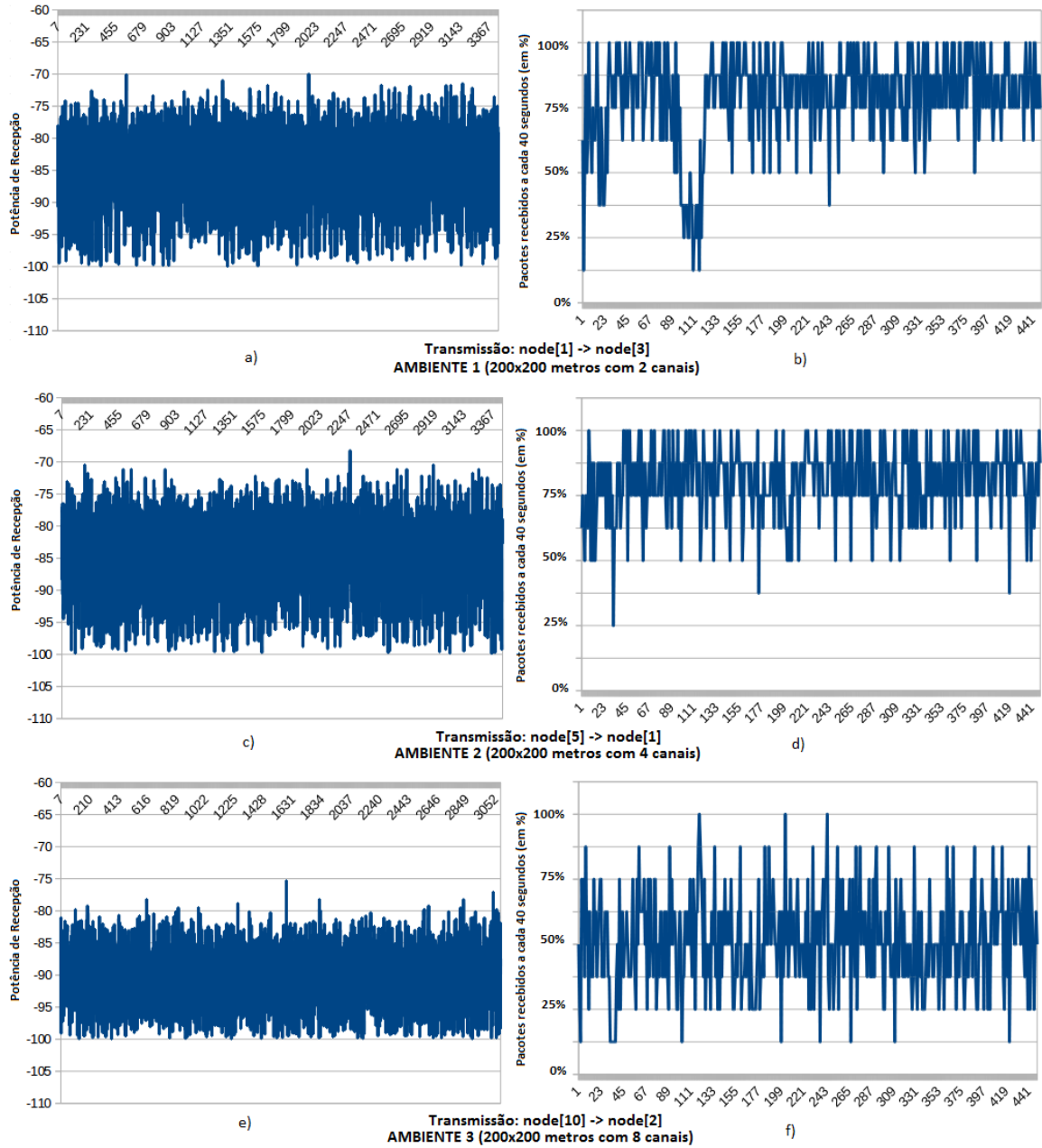
fim da simulação, com algumas poucas variações.

No terceiro gráfico da Fig. 4.5 percebe-se que na metade da simulação o canal de transmissão sofre uma piora brusca e o *node[6]* só consegue receber do *node[19]* em torno de 10% dos pacotes enviados.

Figura 4.5: Potência de recepção *versus* Pacotes recebidos (MARKOV) 200 m.

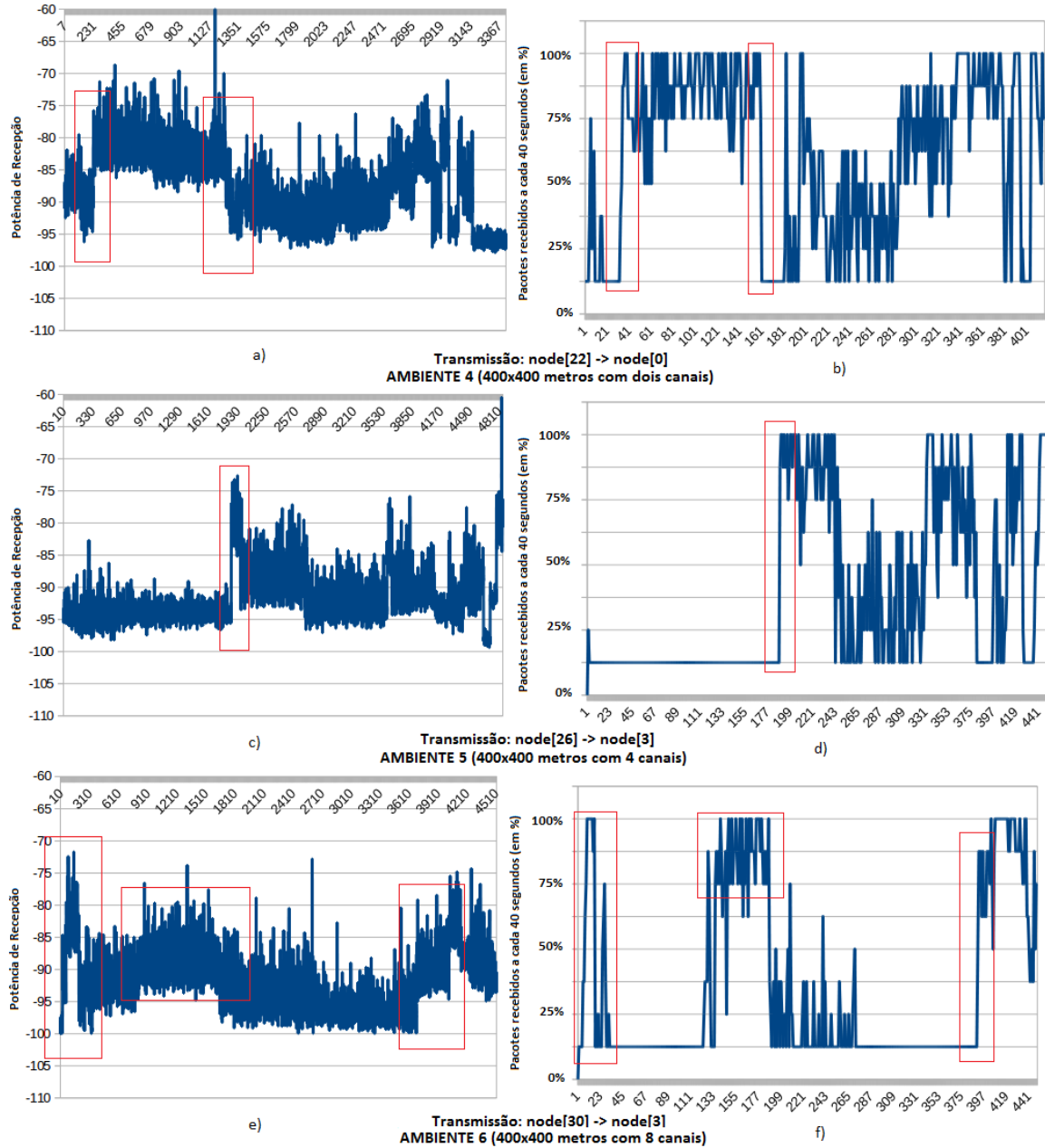


Diferentemente da Fig. 4.5, mas ainda utilizando os ambientes de 200x200 metros, a Fig. 4.6 ilustra a potência de recepção de pacotes e a TRP para os ambientes com 2, 4 e 8 canais no modelo padrão, respectivamente. Pode-se perceber que em todos os gráficos da potência de recepção dessa figura não houve qualquer alteração nas características dos canais utilizados.

Figura 4.6: Potência de recepção *versus* Pacotes recebidos (PADRÃO) 200 m.

O mesmo acontece com as TRPs, que permaneceram praticamente as mesmas durante todo o experimento.

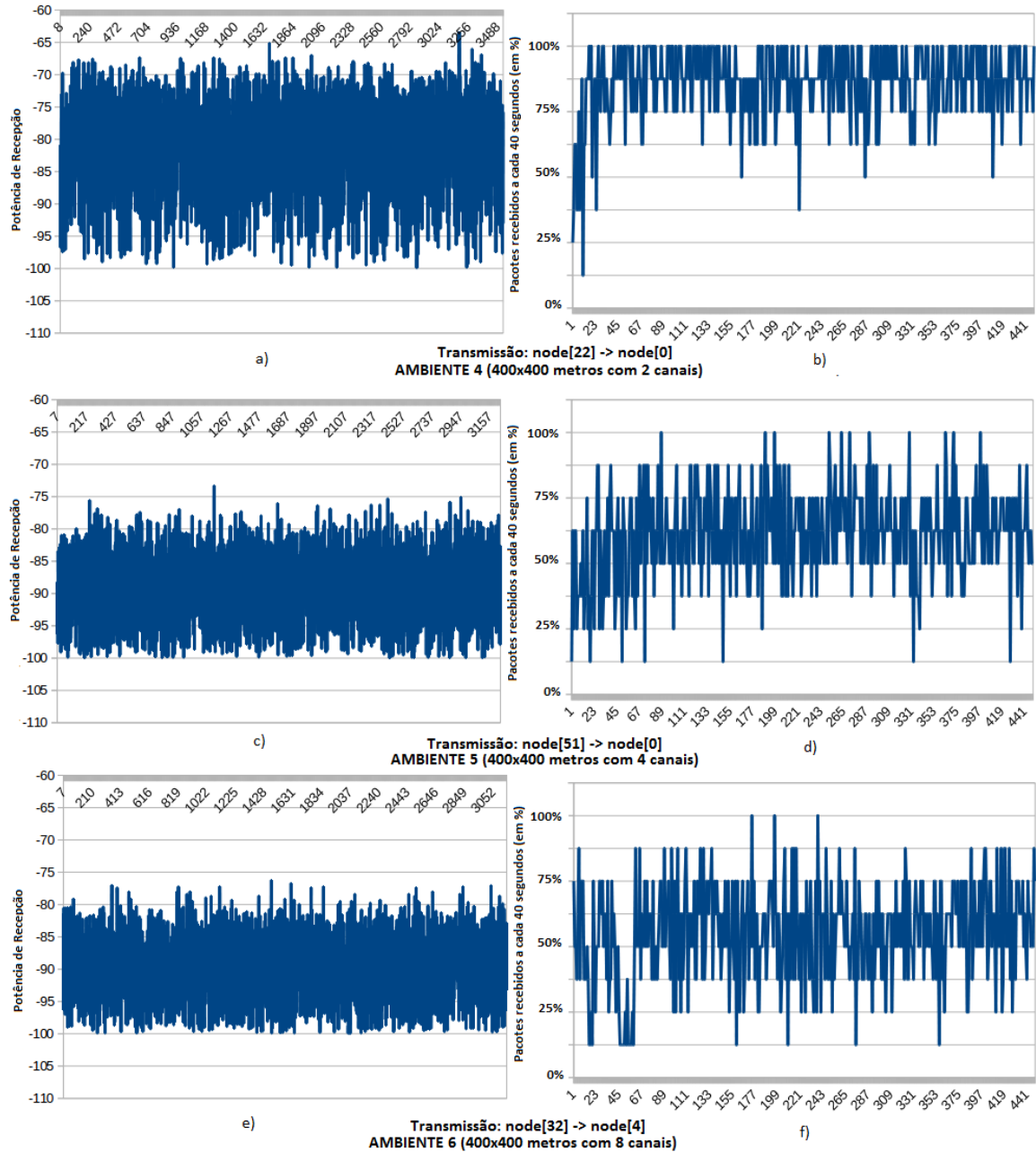
Figura 4.7: Potência de recepção *versus* Pacotes recebidos (MARKOV) 400 m.



Utilizando os ambientes com 400x400 metros de área, foram realizados novos experimentos para identificar a relação entre a potência de recepção e a TRP. Na Fig. 4.7 foi utilizado o modelo de canal não-estacionário. Como os ambientes desse tamanho possuem vários nós (chega a 57 nós, maior do que os 27 nós máximos dos ambientes com 200x200 metros), e muitos deles estão no mesmo canal, a variação na qualidade dos canais aumenta

bastante em relação aos ambientes menores, dada a interferência entre os nós de mesmo canal. Isso é perceptível tanto pela potência recebida, quanto pela TRP. Neles, a modificação na qualidade dos canais tem uma duração mais curta.

Figura 4.8: Potência de recepção *versus* Pacotes recebidos (PADRÃO) 400 m.



Ainda em relação aos ambientes de 400x400 metros, utilizando o modelo estacionário padrão do simulador, obtém-se resultados parecidos com os da Fig. 4.6.

Partindo para a análise da Fig. 4.9, ela mostra o valor médio da da TRP considerando todos os nós conectados com cada nó sorvedouro, para todos os cenários. Para calcular o

valor médio, o TRP média foi amostrada a cada 10 minutos de simulação, e considerando cinco replicações. O nível de confiança foi de 99%. Ao avaliar o desempenho da rede com o modelo estacionário, é possível perceber que, em média, o desempenho da rede aumenta quando mais canais são utilizados. A TRP global da rede quando se utiliza o modelo estacionário foi de 30,15%, 40,8% e 47,49%, para os cenários com dois, quatro e oito canais, respectivamente. Quando se utiliza o modelo não-estacionário, a TRP global da rede foi de 30,34%, 31,63% e 30,66%, para os cenários com dois, quatro, e oito canais, respectivamente. Por isso, o desempenho global permanece quase constante para todos os cenários quando se utiliza o modelo não-estacionário, embora para alguns nós o desempenho tenha aumentado. Mesmo para as sub-árvores que têm quase o mesmo número de nós terminais, as distâncias entre os nós finais e o sorvedouro podem variar em diferentes sub-árvores, e as características dos diferentes canais também são diferentes para o modelo não-estacionário.

A diferença mais importante entre os dois modelos pode ser vista nos valores de variância. Quando se utiliza o modelo estacionário, a TRP para todos os sorvedouros permanece quase a mesma durante as cinco horas de simulação, e mesmo considerando as cinco replicações, uma vez que os nós são colocados em posições fixas.

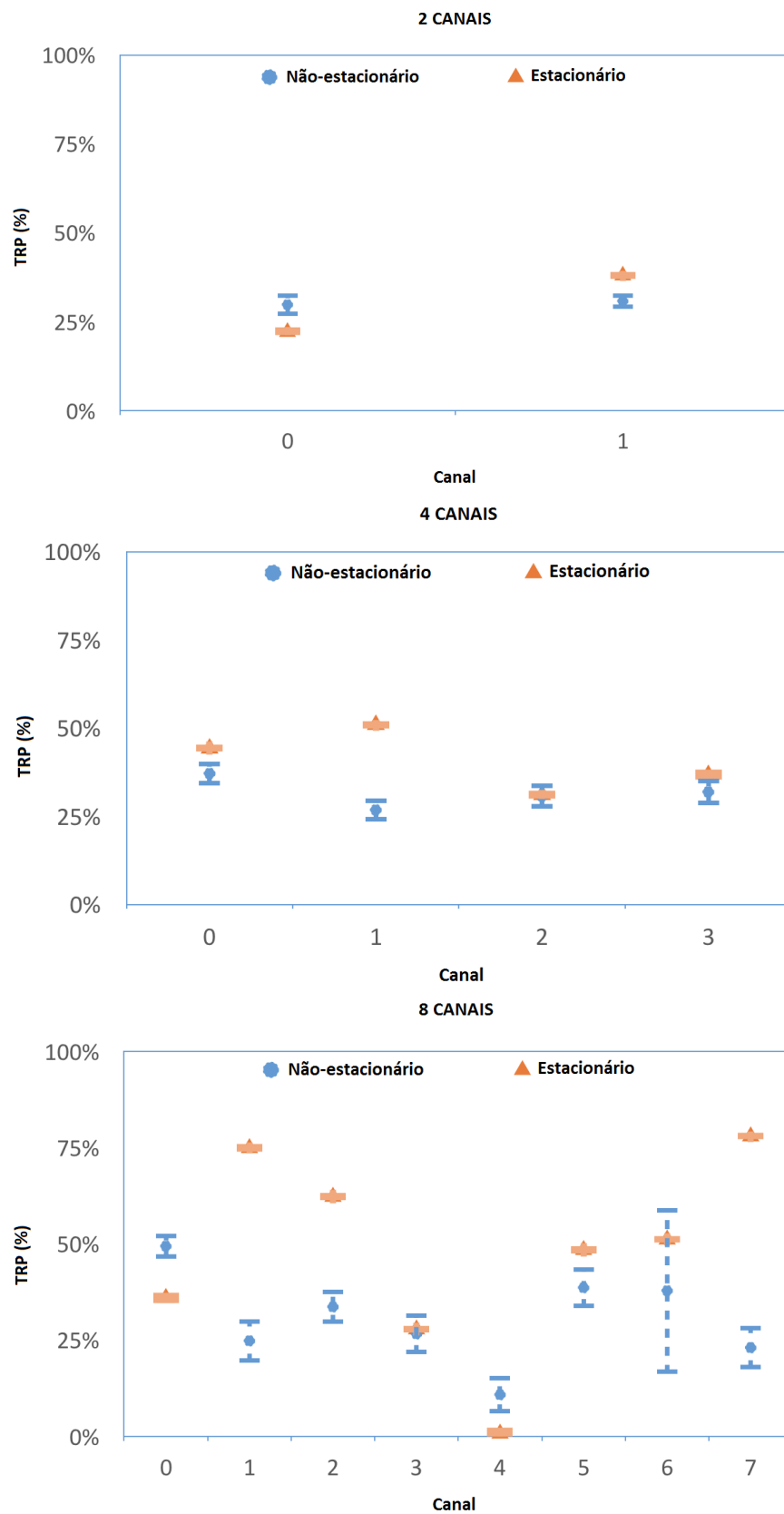
Ao usar o modelo não-estacionário, uma variação maior pode ser observada, devido à variação das características dos canais ao longo do tempo. A variação é maior quando se utiliza oito canais. Isso ocorre porque há menos nós por sub-árvore nesse cenário. Assim, existe uma maior heterogeneidade nas características das sub-árvores. No cenário utilizando dois canais, por exemplo, metade dos nós são alocados para um canal e a outra metade para o outro canal. Por isso, as variações de um determinado enlace podem ser canceladas por uma variação oposta em um outro enlace dentro da sub-árvore, que diminui a variação do valor médio da TRP considerando toda a sub-árvore.

No cenário com 8 canais, a TRP para a sub-árvore que utilizou o canal 4 era quase zero quando utilizou o modelo estacionário. Isso ocorreu porque o sombreamento calculado no início da simulação pelo modelo estacionário permaneceu o mesmo durante todo o tempo. Ao considerar o modelo não-estacionário, devido às variações na qualidade do canal, ao longo do tempo (que pode aumentar ou diminuir a qualidade do canal), a TRP média foi maior, e com uma variância maior.

Por todos os resultados apresentados, é possível perceber que o modelo de canal não-

estacionário de fato captura os efeitos de desvanecimento em larga escala, atenuação em pequena escala, sombreamento log-normal e as características não-estacionárias do canal em longos períodos de tempo. Com o modelo desenvolvido e integrado ao simulador, é possível avaliar protocolos multicanais em RSSFs industriais.

Figura 4.9: Valores médios da Taxa de Recepção de Pacotes para cada sorvedouro em todos os cenários.



Capítulo 5

Conclusões e Trabalhos Futuros

Nesse trabalho foi proposto um modelo de simulação mais adequado para Redes de Sensores Sem Fio Industriais que captura os efeitos de desvanecimento em larga escala, atenuação em pequena escala, sombreamento log-normal e as características não estacionárias do canal em longos períodos de tempo. Utilizando o modelo proposto para o ambiente industrial, foi avaliado o protocolo multicanal *Tree-based Multi-Channel Protocol* (TMCP) nas simulações, e que é baseado em árvore para aplicações de coleta de dados. Ele aloca canais para árvores disjuntas e explora transmissões paralelas entre elas. A partir desse estudo, é possível implementar aplicações que podem monitorar importantes parâmetros e controlar os processos industriais.

Para o modelo de canal não-estacionário foi feita uma abordagem utilizando a cadeia de Markov de dois estados, em que enquanto a rede permanecesse no primeiro estado, suas características permaneciam as mesmas. Dada uma probabilidade de modificação, a rede alteraria de forma significativa suas características, podendo assim causar uma piora ou melhora na qualidade de transmissão do canal. O efeito causado por essa mudança influenciaria na comunicação entre os nós que estivessem trabalhando no mesmo canal.

Para avaliar o desempenho da rede com esse modelo, foram utilizadas métricas como a taxa de recepção de pacotes e potência de recepção no nó sorvedouro. A taxa de recepção foi calculada com base nos pacotes enviados e recebidos. Com essa abordagem, foi possível identificar como o modelo proposto trabalha e comparar com os resultados obtidos dentro da indústria.

Experimentos foram realizados para validar a proposta, utilizando seis cenários de si-

mulação diferentes com posições de nós geradas aleatoriamente. Os experimentos também serviram para avaliar o protocolo multicanal TMCP. O mesmo é executado antes da simulação dos ambientes, de modo a definir para cada nó o seu pai (roteadores) e o canal de comunicação em que irão trabalhar. Antes da simulação são inseridos pelo usuário apenas a quantidade de nós e a posição de cada um. Antes de compilar o código, também é possível definir a quantidade máxima de nós, o alcance máximo entre os nós de modo que eles se comuniquem e a quantidade de canais. Com os experimentos foi possível identificar uma ótima precisão do modelo e do protocolo TMCP.

De modo a permitir a simulação dos ambientes utilizando o modelo proposto, foi necessário alterar o módulo de canal *wirelessChannel* e o de aplicação *ThroughputTest*. No módulo *wirelessChannel* foi inserida a abordagem da cadeia de Markov e no módulo *ThroughputTest* foi inserido um tratamento dos pacotes recebidos pelos sorvedouros e enviados pelos nós finais e roteadores. Nesse módulo é onde são calculados a quantidade de pacotes recebidos e enviados por cada nó.

Para comparar o modelo de canal não-estacionário proposto e o modelo de canal estacionário padrão do Castalia, foram realizados experimentos com os mesmos ambientes nos dois modelos. Foi possível perceber que todos os nós que conseguiram enviar pacotes ao nó sorvedouro no modelo padrão, também conseguiram utilizando o modelo não-estacionário, apenas diferenciados pela quantidade de pacotes recebidos em cada um, dado o perfil multipercurso criado pelo modelo proposto nesse trabalho. Em repetidas simulações o modelo padrão manteve a mesma quantidade de pacotes, o que não acontecia com o modelo não-estacionário. Nesse último, a quantidade de pacotes variava a cada simulação, mas manteve, na maioria das vezes, uma quantidade de pacotes recebidos próximo da média.

Como foi constatado nesse trabalho e em outros na literatura, a qualidade da transmissão no ambiente industrial passa por grandes variações. Uma das maneiras para mitigar o efeito causado por essa variação brusca na qualidade dos canais é o uso de técnicas de alocação dinâmica de espectro. Nelas, os rádios são capazes de escolher o melhor canal para operar de forma dinâmica, não somente no início da operação da rede, como aconteceu na utilização do TMCP nesse trabalho, mas durante toda a operação da rede. Os estudos experimentais realizados na dissertação proveram informações úteis sobre a relação entre o desempenho da comunicação e a qualidade do meio de transmissão. Essa informações podem ser utilizadas

para o desenvolvimento de novos protocolos de múltiplos canais mais adequados a RSSFs.

Como contribuições desse trabalho estão o desenvolvimento de um modelo de canal que simula os efeitos causados pelo ambiente industrial na comunicação sem fio, a utilização da biblioteca IT++ para geração de valores aleatórios com a distribuição de Rice, os estudos experimentais do protocolo de múltiplos canais TMCP, a comparação do modelo proposto com o modelo estacionário padrão do Castalia, e a investigação da relação entre a taxa de recepção de pacotes e a potência de recepção pelo nó servidor.

Como trabalhos futuros, estão listados:

- Implementar um protocolo de múltiplos canais;
- A adaptação do Castalia para que o usuário possa escolher entre uma simulação com ambientes gerais ou com ambientes industriais;
- A avaliação do modelo proposto em outros simuladores de rede, como o NS3;
- A avaliação do modelo utilizando a retransmissão de pacotes, considerando assimetria na conexão;
- A avaliação do impacto do tamanho dos pacotes e da taxa de transmissão no desempenho da rede.

5.0.1 Publicações e Submissões

Como resultado do trabalho realizado na dissertação foi publicado um artigo no SBrT 2015 (*XXXIII Simpósio Brasileiro de Telecomunicações*) [Gomes et al. 2015]. Um outro artigo foi submetido no periódico *IEEE Transactions on Industrial Informatics* cujo título é "An ANN-based Channel Quality Estimator for Industrial WSN Using Dedicated Nodes and Large Samples of RSSI". Submetemos outro artigo para o *INDIN 2016 IEEE International Conference on Industrial Informatics* como resultado do trabalho utilizando o protocolo TMCP, de título "Realistic Simulation of Multichannel Wireless Sensor Networks in Industrial Environments".

Bibliografia

- [Agrawal et al. 2014]AGRAWAL, P. et al. Characterization of Long Term Channel Variations in Industrial Wireless Sensor Networks. In: *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014. p. 1–6. ISBN 978-1-4799-2003-7. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6883285>>.
- [Agrawal (2) et al. 2014]AGRAWAL, P. et al. Long Term Channel Characterization for Energy Efficient Transmission in Industrial Environments. *IEEE Transactions on Communications*, v. 62, n. 8, p. 3004–3014, aug 2014. ISSN 0090-6778. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6842666>>.
- [Akyildiz et al. 2002]AKYILDIZ, I. et al. A Survey on Sensor Networks. *IEEE Communications Magazine*, v. 40, n. 8, p. 102–114, aug 2002. ISSN 0163-6804. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1024422>>.
- [Ali 2012]ALI, Q. I. *Simulation Framework of Wireless Sensor Network (WSN) Using MATLAB/SIMULINK Software*. [S.l.]: Intech, 2012. 324 p. ISBN 978-953-51-0751-4.
- [Amzucu, Li e Fledderus 2014]AMZUCU, D. M.; LI, H.; FLEDDERUS, E. Indoor Radio Propagation and Interference in 2.4 GHz Wireless Sensor Networks: Measurements and Analysis. *Wireless Personal Communications*, v. 76, n. 2, p. 245–269, may 2014. ISSN 0929-6212. Disponível em: <<http://link.springer.com/10.1007/s11277-014-1694-2>>.
- [Bin Lu e Gungor 2009]Bin Lu; GUNGOR, V. Online and Remote Motor Energy Monitoring and Fault Diagnostics Using Wireless Sensor Networks. *IEEE Transactions on Industrial Electronics*, v. 56, n. 11, p. 4651–4659, nov 2009. ISSN 0278-0046. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5191094>>.

- [Carvalho et al. 2012]CARVALHO, F. et al. Aplicações Ambientais de Redes de Sensores Sem Fio. *Revista de Tecnologia da Informação e Comunicação*, v. 2, n. 1, p. 14–19, oct 2012. ISSN 22375112. Disponível em: <<http://www.bibliotekevirtual.org/index.php/2013-02-07-03-02-35/2013-02-07-03-03-11/92-rtic/v02n01/128-v02n01a03.html>>.
- [Castalia User's Manual]Castalia User's Manual. *Manual do Castalia*. Data de acesso: 17/08/2015. Disponível em: <<https://forge.nicta.com.au/docman/view.php/301/592/Castalia++UserManual.pdf>>.
- [Chamberlain 2013]CHAMBERLAIN, T. *Learning OMNeT++*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2013. 102 p. ISBN 139781849697149.
- [Coll 2012]COLL, J. F. *RF Channel Characterization in Industrial, Hospital and Home Environments*. Tese (Doutorado) — KTH School of Information and Communication Technology, 2012.
- [Cordeiro e Agrawal 2011]CORDEIRO, C. de M.; AGRAWAL, D. P. *Ad Hoc and Sensor Networks: Theory and Applications*. [S.l.]: World Scientific, 2011. 664 p. ISBN 978-981-4338-88-2.
- [Delgado Gomes et al. 2012]Delgado Gomes, R. et al. Correlation between Spectral Occupancy and Packet Error Rate in IEEE 802.15.4-based Industrial Wireless Sensor Networks. *IEEE Latin America Transactions*, v. 10, n. 1, p. 1312–1318, jan 2012. ISSN 1548-0992. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6142478>>.
- [Delgado Gomes et al. 2013]Delgado Gomes, R. et al. On the Impact of Local Processing for Motor Monitoring Systems in Industrial Environments Using Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, v. 2013, p. 1–14, 2013. ISSN 1550-1329. Disponível em: <<http://www.hindawi.com/journals/ijdsn/2013/471917/>>.
- [Dezfouli et al. 2015]DEZFOULI, B. et al. Modeling Low-power Wireless Communications. *Journal of Network and Computer Applications*, v. 51, p. 102–126, may 2015. ISSN 10848045. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1084804514000472>>.

- [Dwivedi e Vyas 2011]DWIVEDI, A.; VYAS, O. An Exploratory Study of Experimental Tools for Wireless Sensor Networks. *Wireless Sensor Network*, v. 03, n. 07, p. 215–240, 2011. ISSN 1945-3078. Disponível em: <<http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/wsn.2011.37025>>.
- [Esfahani, Wang e Sundararajan 2014]ESFAHANI, E. T.; WANG, S.; SUNDARARAJAN, V. Multisensor Wireless System for Eccentricity and Bearing Fault Detection in Induction Motors. *IEEE/ASME Transactions on Mechatronics*, v. 19, n. 3, p. 818–826, jun 2014. ISSN 1083-4435. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6517212>>.
- [Font et al. 2011]FONT, J. L. et al. Analysis of Source Code Metrics from NS-2 and NS-3 Network Simulators. *Simulation Modelling Practice and Theory*, v. 19, n. 5, p. 1330–1346, may 2011. ISSN 1569190X. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1569190X11000190>>.
- [Garg 2007]GARG, V. *Wireless Communications and Networking*. [S.l.]: Elsevier, 2007. ISBN 978-0-12-373580-5.
- [Gomes et al. 2013]GOMES, R. et al. Desafios de Redes de Sensores sem Fio Industriais. *Revista de Tecnologia da Informação e Comunicação*, v. 4, n. 1, p. 16–27, dec 2013. ISSN 22375112. Disponível em: <<http://www.bibliotekevirtual.org/index.php/2013-02-07-03-02-35/2013-02-07-03-03-11/265-rtic/v04n01/2089-v04n01a03.html>>.
- [Gomes et al. 2015]GOMES, R. D. et al. Modelo de Simulação Realista de Redes de Sensores sem Fio Industriais. *XXXIII Simpósio Brasileiro de Telecomunicações (SBrT2015)*, 2015.
- [Gomes et al. 2014]GOMES, R. D. et al. Distributed Approach for Channel Quality Estimation Using Dedicated Nodes in Industrial WSN. In: *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*. IEEE, 2014. p. 1943–1948. ISBN 978-1-4799-4912-0. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7136489>>.

- [Guimaraes e Adionel 2009]GUIMARAES; ADIONEL, D. *Digital Transmission: A Simulation-Aided Introduction with VisSim/Comm*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2009. 863 p. ISBN 978-3-642-01358-4.
- [Guizani et al. 2010]GUIZANI, M. et al. *Network Modeling and Simulation: A Practical Perspective*. [S.l.]: Wiley, 2010. 304 p. ISBN 978-0-470-03587-0.
- [Gungor e Hancke 2009]GUNGOR, V. C.; HANCKE, G. P. Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. *IEEE Transactions on Industrial Electronics*, v. 56, n. 10, p. 4258–4265, 2009. ISSN 02780046.
- [Guo, Healy e Zhou 2012]GUO, W.; HEALY, W. M.; ZHOU, M. Impacts of 2.4-GHz ISM Band Interference on IEEE 802.15.4 Wireless Sensor Network Reliability in Buildings. *IEEE Transactions on Instrumentation and Measurement*, 2012.
- [Hou e Bergmann 2012]HOU, L.; BERGMANN, N. W. Novel Industrial Wireless Sensor Networks for Machine Condition Monitoring and Fault Diagnosis. *IEEE Transactions on Instrumentation and Measurement*, v. 61, n. 10, p. 2787–2798, oct 2012. ISSN 0018-9456. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6215047>>.
- [Huang et al. 2013]HUANG, P. et al. The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, v. 15, n. 1, p. 101–120, jan 2013. ISSN 1553-877X. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6188353>>.
- [Incel 2011]INCEL, O. D. A Survey on Multi-channel Communication in Wireless Sensor Networks. *Computer Networks*, v. 55, n. 13, p. 3081–3099, sep 2011. ISSN 13891286. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1389128611001873>>.
- [Jin-Shyan Lee 2006]Jin-Shyan Lee. Performance Evaluation of IEEE 802.15.4 for Low-rate Wireless Personal Area Networks. *IEEE Transactions on Consumer Electronics*, v. 52, n. 3, p. 742–749, aug 2006. ISSN 0098-3063. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1706465>>.

- [Karl e Willig 2007]KARL, H.; WILLIG, A. *Protocols and Architectures for Wireless Sensor Networks*. [S.l.]: John Wiley and Sons, 2007. 526 p. ISBN 978 0470 51923 3.
- [Khan, Pathan e Alrajeh 2012]KHAN, S.; PATHAN, A.-S. K.; ALRAJEH, N. A. *Wireless Sensor Networks*. [S.l.]: CRC Press, 2012. ISBN 9781466506060.
- [Lima-Filho et al. 2012]LIMA-FILHO, A. C. et al. Embedded System Integrated Into a Wireless Sensor Network for Online Dynamic Torque and Efficiency Monitoring in Induction Motors. *IEEE/ASME Transactions on Mechatronics*, v. 17, n. 3, p. 404–414, jun 2012. ISSN 1083-4435. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6177670>>.
- [Lo Bello e Toscano 2009]Lo Bello, L.; TOSCANO, E. Coexistence Issues of Multiple Co-Located IEEE 802.15.4/ZigBee Networks Running on Adjacent Radio Channels in Industrial Environments. *IEEE Transactions on Industrial Informatics*, v. 5, n. 2, p. 157–167, may 2009. ISSN 1551-3203. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4895702>>.
- [Mahalik 2007]MAHALIK, N. P. *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2007. 509 p. ISBN 978-3-540-37364-3.
- [Monika e S. 2014]MONIKA, A. K.; S., M. Network Simulator for Next Generation Networks - An Overview. *International Journal of Mobile Network Communications and Telematics (IJMNCT)*, 2014. ISSN 1839-5678.
- [Namboothiri e Sivalingam 2013]NAMBOOTHIRI, P. G.; SIVALINGAM, K. M. Throughput Analysis of Multiple Channel Based Wireless Sensor Networks. *Wireless Networks*, v. 19, n. 4, p. 461–476, may 2013. ISSN 1022-0038. Disponível em: <<http://link.springer.com/10.1007/s11276-012-0478-4>>.
- [Obaidat e Papadimitriou 2003]OBAIDAT, M. S.; PAPADIMITRIOU, G. I. *Applied System Simulation: Methodologies and Applications*. [S.l.]: Springer US, 2003. 515 p. ISBN 978-1-4020-7603-9.

- [Oliveira 2011]OLIVEIRA, E. E. de. *Protocolo de Decisão do Espectro para Rede de Sensores sem Fio*. 2011.
- [Oliveira 2008]OLIVEIRA, H. A. B. F. de. *Localização no Tempo e no Espaço em Redes de Sensores Sem Fio*. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2008.
- [Omnet++ Homepage]Omnet++ Homepage. *User Manual - OM-NeT++ version 4.6*. Data de acesso: 17/08/2015. Disponível em: <<https://omnetpp.org/doc/omnetpp/manual/usman.html>>.
- [Pal e Nasipuri 2013]PAL, A.; NASIPURI, A. DRCS: A Distributed Routing and Channel Selection Scheme for Multi-channel Wireless Sensor Networks. In: *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, 2013. p. 602–608. ISBN 978-1-4673-5077-8. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6529566>>.
- [R. et al. 2013]R., C. R. M. et al. A Comprehensive Overview on Different Network Simulators. *International Journal of Engineering and Technology (IJET)*, 2013. ISSN 0975-4024.
- [Rappaport 2009]RAPPAFORT, T. S. *Comunicações sem Fio - Princípios e Práticas*. [S.l.]: Prentice Hall PTR, 2009.
- [Remley et al. 2009]REMLEY, K. A. et al. Measurements in Harsh RF Propagation Environments to Support Performance Evaluation of Wireless Sensor Networks. *Sensor Review*, v. 29, n. 3, p. 211–222, jun 2009. ISSN 0260-2288. Disponível em: <<http://www.emeraldinsight.com/doi/abs/10.1108/02602280910967620>>.
- [Shu et al. 2011]SHU, L. et al. NetTopo: A Framework of Simulation and Visualization for Wireless Sensor Networks. *Ad Hoc Networks*, v. 9, n. 5, p. 799–820, jul 2011. ISSN 15708705. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1570870510001435>>.
- [Sikora e Groza 2005]SIKORA, A.; GROZA, V. F. Coexistence of IEEE 802.15.4 with Other Systems in the 2.4 GHz-ISM-Band. *IMTC 2005 – Instrumentation and Measurement Technology Conference*, 2005.

- [Silva et al. 2012]SILVA, I. et al. Reliability and Availability Evaluation of Wireless Sensor Networks for Industrial Applications. *Sensors*, 2012. Disponível em: <<http://www.mdpi.com/1424-8220/12/1/806>>.
- [Society 2006]SOCIETY, I. C. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). In: <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>. [S.l.: s.n.], 2006.
- [Soua e Minet 2015]SOUA, R.; MINET, P. Multichannel Assignment Protocols in Wireless Sensor Networks: A Comprehensive Survey. *Pervasive and Mobile Computing*, v. 16, p. 2–21, jan 2015. ISSN 15741192. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1574119214000613>>.
- [Sousa e Lopes 2011]SOUSA, M.; LOPES, W. Desafios em Redes de Sensores sem Fio. *Revista de Tecnologia da Informação e Comunicação*, v. 1, n. 1, p. 41–47, oct 2011. ISSN 22375112. Disponível em: <<http://www.bibliotekevvirtual.org/index.php/2013-02-07-03-02-35/2013-02-07-03-03-11/90-rtic/v01n01/119-v01n01a08.html>>.
- [Stehlík 2011]STEHLÍK, M. *Comparison of Simulators for Wireless Sensor Networks*. 2011.
- [Sundani et al. 2011]SUNDANI, H. et al. Wireless Sensor Network Simulators: A Survey and Comparisons. *International Journal Of Computer Networks (IJCN)*, 2011.
- [Tabish Niaz, Ahmad Khan e Shafi 2011]Tabish Niaz, M.; Ahmad Khan, A.; SHAFI, I. Passive Source Localization in a Randomly Distributed Wireless Sensor Networks. *International Journal of Computer Applications*, v. 24, n. 6, p. 1–5, jun 2011. ISSN 09758887. Disponível em: <<http://www.ijcaonline.org/volume24/number6/pxc3873955.pdf>>.
- [Tang et al. 2007]TANG, L. et al. Channel Characterization and Link Quality Assessment of IEEE 802.15.4-Compliant Radio for Factory Environments. *IEEE Transactions on Industrial Informatics*, v. 3, n. 2, p. 99–110, may 2007. ISSN 1551-3203. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4214857>>.
- [Tanghe et al. 2008]TANGHE, E. et al. The Industrial Indoor Channel: Large-scale and Temporal Fading at 900, 2400, and 5200 MHz. *IEEE Transactions on Wireless Com-*

- munications*, v. 7, n. 7, p. 2740–2751, jul 2008. ISSN 1536-1276. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4570240>>.
- [Tennina et al. 2013] TENNINA, S. et al. *IEEE 802.15.4 and ZigBee as Enabling Technologies for Low-Power Wireless Systems with Quality-of-Service Constraints*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2013. 169 p. ISBN 978-3-642-37368-8.
- [Wang e Yao 2013] WANG, N.-C.; YAO, K. Characterizing Fading Channel Under Abrupt Temporal Variations. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013. p. 5056–5060. ISBN 978-1-4799-0356-6. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6638624>>.
- [Wehrle, Gunes e Gross 2010] WEHRLE, K.; GUNES, M.; GROSS, J. *Modeling and Tools for Network Simulation Hardcover*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2010. 256 p. ISBN 978-3-642-12330-6.
- [Weingartner, Lehn e Wehrle 2009] WEINGARTNER, E.; LEHN, H. vom; WEHRLE, K. A Performance Comparison of Recent Network Simulators. In: *2009 IEEE International Conference on Communications*. IEEE, 2009. p. 1–5. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5198657>>.
- [Wu et al. 2008] WU, Y. et al. Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks. In: *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. IEEE, 2008. p. 1193–1201. ISBN 978-1-4244-2026-1. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4509770>>.
- [Wykret 2014] WYKRET, T. F. *Coexistência em Redes de Sensores sem Fio: Técnicas de Seleção Dinâmica de Canal*. 2014.
- [Yang, Xu e Gidlund 2011] YANG, D.; XU, Y.; GIDLUND, M. Wireless Coexistence Between IEEE 802.11- and IEEE 802.15.4-Based Networks: A Survey. *International Journal of Distributed Sensor Networks*, v. 2011, p. 1–17, 2011. ISSN 1550-1329. Disponível em: <<http://www.hindawi.com/journals/ijdsn/2011/912152/>>.
- [Zhao e Guibas 2004] ZHAO, F.; GUIBASL, L. *Wireless Sensor Networks: An Information Processing Approach*. [S.l.]: Morgan Kaufmann, 2004. 376 p. ISBN 978-1558609143.

Apêndice A

Código em C++ do protocolo de múltiplos canais TMCP

Código Fonte A.1: Código do protocolo multicanal TMCP

```
1  #include <iostream>
2  #include <vector>
3  #include <deque>
4  #include <cmath>
5  #include <string.h>
6  #include <cstdlib>
7
8  using namespace std;
9
10 #define MAX 1010 /* Quantidade maxima de nos */
11 #define RANGE 30
12 #define NCHANNELS 4
13
14 /* Modela a estrutura de um no */
15 typedef struct {
16     vector<int> conn;
17     int id;
18     /* Qtde. de nos que ele pode se conectar de modo que encontre um caminho com
19     o menor numero de saltos possiveis ate o sink (melhor caminho)*/
20     vector<int> p;
21     int c; /* Canal que vai utilizar */
22     int i; /* Qtde. de nos que interferem nele */
23     int x,y,z; /* Posicao dos nos */
24     int nexthop; /* Quem e a conexao desse no */
25 } node;
26
27 /* Estrutura que modela o grafo */
```

```

28 typedef struct {
29     node nodes[MAX];
30     int d[MAX]; /* Nos que foram descobertos na busca usado no algoritmo BFS */
31     int s[MAX]; /* Quantidade de saltos a partir de cada um dos nos ate o sink */
32 } graph;
33
34 /* Compara dois nos pra fazer a ordenacao de acordo com a quantidade de pais ,
35 considerado o menor caminho com o p.size(). Usado na implementacao do Quick Sort.
36 Se primeiro valor > segundo, retorna positivo. Se mesma quantidade, retorna zero */
37 int compare(const void* a, const void* b) {
38     node n1 = *(node*)a;
39     node n2 = *(node*)b;
40     return n1.p.size() - n2.p.size();
41 }
42
43 /* Inicializa o grafo para todos os nos */
44 void initGraph(graph &g) {
45     for(int i = 0; i < MAX; i++) {
46         g.nodes[i].p.clear();
47         g.nodes[i].c = -1;
48         g.nodes[i].i = 0;
49         g.nodes[i].x = g.nodes[i].y = g.nodes[i].z = 0;
50         g.nodes[i].conn.clear();
51         g.d[i] = g.s[i] = 0;
52         g.nodes[i].nexthop = 0;
53     }
54 }
55
56 /* Inclui novo no nesse grafo */
57 void insertNode(graph &g, int id, int x, int y, int z) {
58     g.nodes[id].x = x;
59     g.nodes[id].y = y;
60     g.nodes[id].z = z;
61     g.nodes[id].id = id;
62 }
63
64 /* Distancia entre dois nos */
65 double calculateDistance(node src, node dst) {
66     double dist = sqrt(pow((src.x-dst.x),2.0) + pow((src.y-dst.y),2.0) + pow((src.z-dst
        .z),2));
67     return dist;
68 }
69
70 /* Identificadores dos nos (src e dst). Vai no array de nos do grafo e insere nova
71 conexao no vetor de conexoes */
72 void insertConnection(graph &g, int src, int dst) {
73     g.nodes[src].conn.push_back(dst);

```

```

74 }
75
76 /* Retorna valor booleano se no possui pai nessa arvore */
77 bool hasParentinTree(vector<int> parents , vector<int> tree) {
78     for(int p = 0; p < parents.size(); p++) {
79         int id = parents[p];
80         for(int i = 0; i < tree.size(); i++) {
81             if(id == tree[i]) return true;
82         }
83     }
84     return false;
85 }
86
87 /* Se possui pai, calcula interferencia que este no causa */
88 int calculateInterference(node n, vector<int> tree) {
89     int cont = 0;
90     for(int i = 0; i < n.conn.size(); i++) {
91         for(int j = 0; j < tree.size(); j++) {
92             if(n.conn[i] == tree[j]) cont++;
93         }
94     }
95     return cont;
96 }
97
98 /* Calcula se o no e pai */
99 bool isParent(node n, int p) {
100     for(int i = 0; i < n.p.size(); i++) {
101         if(p == n.p[i]) return true;
102     }
103     return false;
104 }
105
106 /* Algoritmo BFS a partir do sink node (node 0). Calcula o menor caminho
107 de cada um dos nos ate ele. Adiciona a cada no os possiveis pais que fornecem
108 o menor caminho. */
109 void BFS(graph &g) {
110     deque<int> fila;
111     g.d[0] = 1;
112     g.s[0] = 0;
113     fila.push_front(0);
114     while(fila.size() > 0) {
115         int x = fila.back();
116         fila.pop_back();
117         for(int i = 0; i < g.nodes[x].conn.size(); i++) {
118             if(!g.d[g.nodes[x].conn[i]]) {
119                 g.d[g.nodes[x].conn[i]] = 1;
120                 fila.push_front(g.nodes[x].conn[i]);

```

```

121         g.nodes[g.nodes[x].conn[i]].p.push_back(x);
122         g.s[g.nodes[x].conn[i]] = g.s[x]+1;
123     }
124     else { /* Busca o menor caminho */
125         if(g.s[g.nodes[x].conn[i]] > g.s[x]+1) {
126             g.s[g.nodes[x].conn[i]] = g.s[x]+1; /* Atualiza
127                 distancia */
128             g.nodes[g.nodes[x].conn[i]].p.clear();
129             g.nodes[g.nodes[x].conn[i]].p.push_back(x); /*
130                 Atualiza pai */
131         }
132     }
133 }
134 }
135 }
136 }
137
138 int main() {
139     int N; /* Numero de nos */
140     int x,y,z; /* Le a posicao */
141     graph g; /* Grafo que modela a rede */
142     vector<int> trees[NCHANNELS];
143     initGraph(g);
144
145     cin >> N;
146
147     cin >> x >> y >> z; /* Le a posicao do sink node */
148     insertNode(g, 0, x, y, z);
149
150     for(int i = 1; i < N; i++) {
151         cin >> x >> y >> z; /* Le a posicao do no i */
152         insertNode(g, i, x, y, z);
153     }
154
155     for(int i = 0; i < N; i++) {
156         for(int j = 0; j < N; j++) {
157             if(i == j) continue;
158             if(calculateDistance(g.nodes[i], g.nodes[j]) <= RANGE) {
159                 insertConnection(g,i,j);
160             }
161         }
162     }
163
164     /* Procura pais e define o nivel do no na arvore */

```

```

165     BFS(g);
166     for(int i = 0; i < NCHANNELS; i++) {
167         trees[i].push_back(0); /* Adiciona o sink node em todas as arvores */
168     }
169     int level = 1; /* Inicia a analise a partir dos nos mais proximos ao sink */
170     int h = 0;
171     for(int i = 0; i < N; i++) { /* Encontrando a altura da arvore */
172         if(h < g.s[i]) h = g.s[i];
173     }
174     int k=0;
175     do {
176         node list[MAX];
177         k=0;
178         for(int i = 0; i < N; i++) {
179             /* Criando a lista de nos do nivel atual que estao sem canal */
180             if(g.s[i] == level && g.nodes[i].c == -1) {
181                 list[k] = g.nodes[i];
182                 k++;
183             }
184         }
185         /* Ordena a lista de acordo com o numero de pais */
186         qsort(list,k, sizeof(node), compare);
187         int mininter = N+1;
188         int besttree = 0;
189         for(int i = 0; i < k; i++) { /* Para cada no na lista */
190             cout << "Defining channel for node " << list[i].id << endl;
191             mininter = N+1;
192             besttree = 0;
193             for(int t = 0; t < NCHANNELS ; t++) { /* Para cada arvore */
194                 cout << "Analyzing tree " << t << endl;
195                 /* O no possui pai nessa arvore? */
196                 bool hasParent = hasParentinTree(list[i].p, trees[t]);
197                 /* Se sim, qual a interferencia que este no causa? */
198                 if(hasParent) {
199                     int interf = calculateInterference(list[i], trees[t
200                     ]);
201                     cout << "Interference in the tree: " << interf <<
202                     endl;
203                     /* Minimiza a interferencia (define a melhor arvore
204                     ) */
205                     if(mininter > interf) {
206                         mininter = interf;
207                         besttree = t;
208                     }
209                 }
210                 else if(mininter == interf) {
211                     if(trees[besttree].size() > trees[t].size()
212                     ) {

```

```

208                                     mininter = interf;
209                                     besttree = t;
210                                     }
211                                 }
212                            }
213                    }
214                    int bestparent = 0;
215                    int minparent = N+1;
216                    /* Dentro da melhor arvore, qual o pai com menos conexoes na arvore
                        final? */
217                    for(int j = 0; j < trees[besttree].size(); j++) {
218                        if(isParent(list[i], trees[besttree][j])) {
219                            cout << "Interference of " << trees[besttree][j] <<
220                                ": " << g.nodes[trees[besttree][j]].i << endl;
221                            if(minparent > g.nodes[trees[besttree][j]].i) {
222                                minparent = g.nodes[trees[besttree][j]].i;
223                                bestparent = trees[besttree][j];
224                            }
225                        }
226                    }
227                    cout << "Node " << list[i].id << " with parent: " << bestparent <<
228                        " and channel: " << besttree << endl;
229                    /* Define o canal e o pai */
230                    g.nodes[list[i].id].nexthop = bestparent;
231                    g.nodes[list[i].id].c = besttree;
232                    g.nodes[bestparent].i++; /* Uma interferencia a mais neste pai */
233                    trees[besttree].push_back(list[i].id); /* Adiciona o no na arvore
                        */
234                }
235                level++;
236            } while(level <= h);
237
238            for(int i = 0; i < N; i++) {
239                cout << g.nodes[i].id << " " << g.s[g.nodes[i].id] << " " << g.nodes[i].i
240                    << endl;
241                for(int j = 0; j < g.nodes[i].p.size(); j++) {
242                    cout << g.nodes[i].p[j] << " ";
243                }
244                cout << endl;
245                cout << "Next hop: " << g.nodes[i].nexthop << endl;
246                cout << "Channel: " << g.nodes[i].c << endl;
247            }
248            cout << "# Coordenadas dos nos" << endl; /* Coordenadas escritas no padrao do
                script de simulacao do Castalia */
249            for(int i = 0; i < N; i++) {
250                cout << "SN.node["<<i<<"].xCoor = " << g.nodes[i].x <<endl;
251                cout << "SN.node["<<i<<"].yCoor = " << g.nodes[i].y <<endl;

```

```
249         cout << "SN.node["<<i<<"].zCoor = " << g.nodes[i].z <<endl;
250         cout << endl;
251     }
252     cout << "# Frequencia (canal) dos nos" << endl; /* Frequencia escrita no padrao do
                script de simulacao do Castalia */
253     for(int i = 0; i < N; i++) {
254         cout << "SN.node["<<i<<"].Communication.Radio.carrierFreq = " << g.nodes[i
                ].c*5+2405.0 << endl;
255     }
256
257     return 0;
258 }
```

Apêndice B

**Tabelas de localização dos nós nos
ambientes, incluindo identificação dos
nós folhas e respectivos pais**

Tabela B.1: Localização dos nós no ambiente com área de 200x200 m com múltiplos canais.

Área 200x200m 2 canais		Área 200x200m 4 canais		Área 200x200m 8 canais		
Nós sorvedouros	MHz	Nós sorvedouros	MHz	Nós sorvedouros	MHz	
SN.node[0].xCoor = 100	2405	SN.node[0].xCoor = 100	2405	SN.node[0].xCoor = 100	2405	
SN.node[0].yCoor = 100		SN.node[0].yCoor = 100		SN.node[0].yCoor = 100		
SN.node[1].xCoor = 100	2410	SN.node[1].xCoor = 100	2410	SN.node[1].xCoor = 100	2410	
SN.node[1].yCoor = 100		SN.node[1].yCoor = 100		SN.node[1].yCoor = 100		
Nós Finais/Roteadores		SN.node[2].xCoor = 100	2415	SN.node[2].xCoor = 100	2415	
SN.node[2].xCoor = 92	2405	SN.node[2].yCoor = 100		SN.node[2].yCoor = 100		
SN.node[2].yCoor = 171		2410	SN.node[3].xCoor = 100	2420	SN.node[3].xCoor = 100	2420
SN.node[3].xCoor = 53	SN.node[3].yCoor = 100		SN.node[3].yCoor = 100			
SN.node[3].yCoor = 186	Nós Finais/Roteadores		SN.node[4].xCoor = 100	2425		
SN.node[4].xCoor = 189	2405	SN.node[4].xCoor = 92	2420		SN.node[4].yCoor = 100	2430
SN.node[4].yCoor = 56		SN.node[4].yCoor = 171		SN.node[5].xCoor = 100		
SN.node[5].xCoor = 102	2410	SN.node[5].xCoor = 53	2410	SN.node[5].yCoor = 100	2435	
SN.node[5].yCoor = 22		SN.node[5].yCoor = 186		SN.node[6].xCoor = 100		
SN.node[6].xCoor = 173	2410	SN.node[6].xCoor = 189	2415	SN.node[6].yCoor = 100	2440	
SN.node[6].yCoor = 199		SN.node[6].yCoor = 56		SN.node[7].xCoor = 100		
SN.node[7].xCoor = 179	2405	SN.node[7].xCoor = 102	2420	SN.node[7].yCoor = 100	Nós Finais/Roteadores	
SN.node[7].yCoor = 50		SN.node[7].yCoor = 22				
SN.node[8].xCoor = 151	2410	SN.node[8].xCoor = 173	2415	SN.node[8].xCoor = 92	2415	
SN.node[8].yCoor = 101		SN.node[8].yCoor = 199		SN.node[8].yCoor = 171		
SN.node[9].xCoor = 96	2405	SN.node[9].xCoor = 179	2405	SN.node[9].xCoor = 53	2410	
SN.node[9].yCoor = 166		SN.node[9].yCoor = 50		SN.node[9].yCoor = 186		
SN.node[10].xCoor = 140	2405	SN.node[10].xCoor = 151	2410	SN.node[10].xCoor = 189	2415	
SN.node[10].yCoor = 161		SN.node[10].yCoor = 101		SN.node[10].yCoor = 56		
SN.node[11].xCoor = 65	2410	SN.node[11].xCoor = 96	2415	SN.node[11].xCoor = 102	2420	
SN.node[11].yCoor = 13		SN.node[11].yCoor = 166		SN.node[11].yCoor = 22		
SN.node[12].xCoor = 104	2410	SN.node[12].xCoor = 140	2420	SN.node[12].xCoor = 173	2435	
SN.node[12].yCoor = 110		SN.node[12].yCoor = 161		SN.node[12].yCoor = 199		
SN.node[13].xCoor = 91	2405	SN.node[13].xCoor = 65	2410	SN.node[13].xCoor = 179	2425	
SN.node[13].yCoor = 26		SN.node[13].yCoor = 13		SN.node[13].yCoor = 50		
SN.node[14].xCoor = 150	2410	SN.node[14].xCoor = 104	2405	SN.node[14].xCoor = 151	2430	
SN.node[14].yCoor = 13		SN.node[14].yCoor = 110		SN.node[14].yCoor = 101		
SN.node[15].xCoor = 77	2405	SN.node[15].xCoor = 91	2415	SN.node[15].xCoor = 96	2435	
SN.node[15].yCoor = 26		SN.node[15].yCoor = 26		SN.node[15].yCoor = 166		

Tabela B.2: Continuação da Tab. B.1.

Área 200x200m 2 canais	MHz	Área 200x200m 4 canais	MHz	Área 200x200m 8 canais	MHz
SN.node[16].xCoor = 61	2405	SN.node[16].xCoor = 150	2420	SN.node[16].xCoor = 140	2440
SN.node[16].yCoor = 199		SN.node[16].yCoor = 13		SN.node[16].yCoor = 161	
SN.node[17].xCoor = 52	2410	SN.node[17].xCoor = 77	2410	SN.node[17].xCoor = 65	2410
SN.node[17].yCoor = 150		SN.node[17].yCoor = 26		SN.node[17].yCoor = 13	
SN.node[18].xCoor = 198	2410	SN.node[18].xCoor = 61	2405	SN.node[18].xCoor = 104	2405
SN.node[18].yCoor = 183		SN.node[18].yCoor = 199		SN.node[18].yCoor = 110	
SN.node[19].xCoor = 45	2410	SN.node[19].xCoor = 52	2415	SN.node[19].xCoor = 91	2435
SN.node[19].yCoor = 18		SN.node[19].yCoor = 150		SN.node[19].yCoor = 26	
SN.node[20].xCoor = 54	2405	SN.node[20].xCoor = 198	2410	SN.node[20].xCoor = 150	2440
SN.node[20].yCoor = 46		SN.node[20].yCoor = 183		SN.node[20].yCoor = 13	
		SN.node[21].xCoor = 45	2405	SN.node[21].xCoor = 77	2415
		SN.node[21].yCoor = 18		SN.node[21].yCoor = 26	
		SN.node[22].xCoor = 54	2405	SN.node[22].xCoor = 61	2405
		SN.node[22].yCoor = 46		SN.node[22].yCoor = 199	
				SN.node[23].xCoor = 52	2420
				SN.node[23].yCoor = 150	
				SN.node[24].xCoor = 198	2430
				SN.node[24].yCoor = 183	
				SN.node[25].xCoor = 45	2425
				SN.node[25].yCoor = 18	
				SN.node[26].xCoor = 54	2405
				SN.node[26].yCoor = 46	

Tabela B.3: Nós folhas em cada ambiente (200x200 m) e seus respectivos pais (roteadores).

Ambiente 1 (200x200 2 canais)	Ambiente 2 (200x200 4 canais)	Ambiente 3 (200x200 8 canais)
SN.node[6].Application.paiNext = "8"	SN.node[8].Application.paiNext = "11"	SN.node[12].Application.paiNext = "15"
SN.node[18].Application.paiNext = "8"	SN.node[18].Application.paiNext = "14"	SN.node[22].Application.paiNext = "18"
SN.node[16].Application.paiNext = "9"	SN.node[20].Application.paiNext = "10"	SN.node[24].Application.paiNext = "14"

Tabela B.4: Nós folhas em cada ambiente (400x400 m) e seus respectivos pais (roteadores).

Ambiente 4 (400x400 2 canais)	Ambiente 5 (400x400 4 canais)	Ambiente 6 (400x400 8 canais)
SN.node[3].Application.paiNext = "48"	SN.node[5].Application.paiNext = "50"	SN.node[9].Application.paiNext = "54"
SN.node[5].Application.paiNext = "7"	SN.node[7].Application.paiNext = "9"	SN.node[11].Application.paiNext = "13"
SN.node[6].Application.paiNext = "5"	SN.node[8].Application.paiNext = "7"	SN.node[12].Application.paiNext = "11"
SN.node[7].Application.paiNext = "49"	SN.node[9].Application.paiNext = "35"	SN.node[13].Application.paiNext = "39"
SN.node[8].Application.paiNext = "30"	SN.node[10].Application.paiNext = "32"	SN.node[14].Application.paiNext = "36"
SN.node[9].Application.paiNext = "45"	SN.node[11].Application.paiNext = "24"	SN.node[15].Application.paiNext = "28"
SN.node[10].Application.paiNext = "9"	SN.node[12].Application.paiNext = "52"	SN.node[16].Application.paiNext = "15"
SN.node[11].Application.paiNext = "48"	SN.node[13].Application.paiNext = "50"	SN.node[17].Application.paiNext = "54"
SN.node[12].Application.paiNext = "7"	SN.node[14].Application.paiNext = "9"	SN.node[18].Application.paiNext = "13"
SN.node[13].Application.paiNext = "14"	SN.node[15].Application.paiNext = "16"	SN.node[19].Application.paiNext = "50"
SN.node[14].Application.paiNext = "24"	SN.node[16].Application.paiNext = "47"	SN.node[20].Application.paiNext = "30"
SN.node[15].Application.paiNext = "19"	SN.node[17].Application.paiNext = "21"	SN.node[21].Application.paiNext = "25"
SN.node[16].Application.paiNext = "48"	SN.node[18].Application.paiNext = "50"	SN.node[22].Application.paiNext = "54"
SN.node[17].Application.paiNext = "13"	SN.node[19].Application.paiNext = "15"	SN.node[23].Application.paiNext = "19"
SN.node[18].Application.paiNext = "30"	SN.node[20].Application.paiNext = "37"	SN.node[24].Application.paiNext = "41"
SN.node[19].Application.paiNext = "21"	SN.node[21].Application.paiNext = "23"	SN.node[25].Application.paiNext = "27"
SN.node[20].Application.paiNext = "48"	SN.node[22].Application.paiNext = "50"	SN.node[26].Application.paiNext = "54"
SN.node[21].Application.paiNext = "7"	SN.node[23].Application.paiNext = "9"	SN.node[27].Application.paiNext = "13"
SN.node[23].Application.paiNext = "12"	SN.node[25].Application.paiNext = "14"	SN.node[29].Application.paiNext = "18"
SN.node[25].Application.paiNext = "44"	SN.node[27].Application.paiNext = "46"	SN.node[31].Application.paiNext = "41"
SN.node[28].Application.paiNext = "35"	SN.node[30].Application.paiNext = "37"	SN.node[34].Application.paiNext = "41"
SN.node[29].Application.paiNext = "19"	SN.node[31].Application.paiNext = "21"	SN.node[35].Application.paiNext = "25"
SN.node[30].Application.paiNext = "27"	SN.node[32].Application.paiNext = "29"	SN.node[36].Application.paiNext = "33"
SN.node[31].Application.paiNext = "30"	SN.node[33].Application.paiNext = "52"	SN.node[37].Application.paiNext = "36"
SN.node[32].Application.paiNext = "48"	SN.node[34].Application.paiNext = "50"	SN.node[38].Application.paiNext = "54"
SN.node[35].Application.paiNext = "2"	SN.node[37].Application.paiNext = "26"	SN.node[41].Application.paiNext = "8"
SN.node[36].Application.paiNext = "19"	SN.node[38].Application.paiNext = "21"	SN.node[42].Application.paiNext = "25"
SN.node[38].Application.paiNext = "7"	SN.node[40].Application.paiNext = "45"	SN.node[44].Application.paiNext = "49"
SN.node[39].Application.paiNext = "19"	SN.node[41].Application.paiNext = "21"	SN.node[45].Application.paiNext = "25"
SN.node[40].Application.paiNext = "34"	SN.node[42].Application.paiNext = "6"	SN.node[46].Application.paiNext = "30"
SN.node[41].Application.paiNext = "50"	SN.node[43].Application.paiNext = "32"	SN.node[47].Application.paiNext = "56"
SN.node[42].Application.paiNext = "14"	SN.node[44].Application.paiNext = "37"	SN.node[48].Application.paiNext = "20"
SN.node[43].Application.paiNext = "4"	SN.node[45].Application.paiNext = "26"	SN.node[49].Application.paiNext = "10"
SN.node[44].Application.paiNext = "2"	SN.node[46].Application.paiNext = "4"	SN.node[50].Application.paiNext = "40"
SN.node[46].Application.paiNext = "43"	SN.node[48].Application.paiNext = "42"	SN.node[52].Application.paiNext = "46"
SN.node[47].Application.paiNext = "50"	SN.node[49].Application.paiNext = "52"	SN.node[53].Application.paiNext = "56"
SN.node[48].Application.paiNext = "10"	SN.node[50].Application.paiNext = "12"	SN.node[54].Application.paiNext = "16"
SN.node[50].Application.paiNext = "22"	SN.node[52].Application.paiNext = "28"	SN.node[56].Application.paiNext = "51"

Tabela B.5: Localização dos nós no ambiente com área de 400x400 m com múltiplos canais.

Área 400x400m 2 canais		Área 400x400m 4 canais		Área 400x400m 8 canais	
Nós sorvedouros	MHz	Nós sorvedouros	MHz	Nós sorvedouros	MHz
SN.node[0].xCoor = 200	2405	SN.node[0].xCoor = 200	2405	SN.node[0].xCoor = 200	2405
SN.node[0].yCoor = 200		SN.node[0].yCoor = 200		SN.node[0].yCoor = 200	
SN.node[1].xCoor = 200	2410	SN.node[1].xCoor = 200	2410	SN.node[1].xCoor = 200	2410
SN.node[1].yCoor = 200		SN.node[1].yCoor = 200		SN.node[1].yCoor = 200	
Nós Finais/Roteadores		SN.node[2].xCoor = 200	2415	SN.node[2].xCoor = 200	2415
SN.node[2].xCoor = 142	2405	SN.node[2].yCoor = 200		SN.node[2].yCoor = 200	
SN.node[2].yCoor = 265		2410	SN.node[3].xCoor = 200	2420	SN.node[3].xCoor = 200
SN.node[3].xCoor = 313	SN.node[3].yCoor = 200		SN.node[3].yCoor = 200		
SN.node[3].yCoor = 73	Nós Finais/Roteadores		SN.node[4].xCoor = 200	2425	
SN.node[4].xCoor = 184	2410	SN.node[4].xCoor = 142	2415		SN.node[4].yCoor = 200
SN.node[4].yCoor = 221		SN.node[4].yCoor = 265		SN.node[5].xCoor = 200	
SN.node[5].xCoor = 88	2405	SN.node[5].xCoor = 313	2405	SN.node[5].yCoor = 200	2435
SN.node[5].yCoor = 77		SN.node[5].yCoor = 73		SN.node[6].xCoor = 200	
SN.node[6].xCoor = 154	2405	SN.node[6].xCoor = 184	2410	SN.node[6].yCoor = 200	2440
SN.node[6].yCoor = 11		SN.node[6].yCoor = 221		SN.node[7].xCoor = 200	
SN.node[7].xCoor = 82	2405	SN.node[7].xCoor = 88	2415	SN.node[7].yCoor = 200	
SN.node[7].yCoor = 170		SN.node[7].yCoor = 77		Nós Finais/Roteadores	
SN.node[8].xCoor = 377	2410	SN.node[8].xCoor = 154	2415	SN.node[8].xCoor = 142	2425
SN.node[8].yCoor = 386		SN.node[8].yCoor = 11		SN.node[8].yCoor = 265	
SN.node[9].xCoor = 298	2410	SN.node[9].xCoor = 82	2415	SN.node[9].xCoor = 313	2415
SN.node[9].yCoor = 257		SN.node[9].yCoor = 170		SN.node[9].yCoor = 73	
SN.node[10].xCoor = 385	2410	SN.node[10].xCoor = 377	2410	SN.node[10].xCoor = 184	2410
SN.node[10].yCoor = 241		SN.node[10].yCoor = 386		SN.node[10].yCoor = 221	
SN.node[11].xCoor = 332	2410	SN.node[11].xCoor = 298	2415	SN.node[11].xCoor = 88	2435
SN.node[11].yCoor = 69		SN.node[11].yCoor = 257		SN.node[11].yCoor = 77	
SN.node[12].xCoor = 72	2405	SN.node[12].xCoor = 385	2405	SN.node[12].xCoor = 154	2435
SN.node[12].yCoor = 92		SN.node[12].yCoor = 241		SN.node[12].yCoor = 11	
SN.node[13].xCoor = 81	2410	SN.node[13].xCoor = 332	2405	SN.node[13].xCoor = 82	2435
SN.node[13].yCoor = 371		SN.node[13].yCoor = 69		SN.node[13].yCoor = 170	
SN.node[14].xCoor = 172	2410	SN.node[14].xCoor = 72	2415	SN.node[14].xCoor = 377	2430
SN.node[14].yCoor = 370		SN.node[14].yCoor = 92		SN.node[14].yCoor = 386	
SN.node[15].xCoor = 296	2405	SN.node[15].xCoor = 81	2410	SN.node[15].xCoor = 298	2415
SN.node[15].yCoor = 0		SN.node[15].yCoor = 371		SN.node[15].yCoor = 257	

Tabela B.6: Continuação da Tab. B.5.

Área 400x400m 2 canais	MHz	Área 400x400m 4 canais	MHz	Área 400x400m 8 canais	MHz
SN.node[16].xCoor = 336	2410	SN.node[16].xCoor = 172	2410	SN.node[16].xCoor = 385	2415
SN.node[16].yCoor = 104		SN.node[16].yCoor = 370		SN.node[16].yCoor = 241	
SN.node[17].xCoor = 25	2410	SN.node[17].xCoor = 296	2415	SN.node[17].xCoor = 332	2415
SN.node[17].yCoor = 371		SN.node[17].yCoor = 0		SN.node[17].yCoor = 69	
SN.node[18].xCoor = 304	2410	SN.node[18].xCoor = 336	2405	SN.node[18].xCoor = 72	2435
SN.node[18].yCoor = 365		SN.node[18].yCoor = 104		SN.node[18].yCoor = 92	
SN.node[19].xCoor = 209	2405	SN.node[19].xCoor = 25	2410	SN.node[19].xCoor = 81	2440
SN.node[19].yCoor = 52		SN.node[19].yCoor = 371		SN.node[19].yCoor = 371	
SN.node[20].xCoor = 370	2410	SN.node[20].xCoor = 304	2420	SN.node[20].xCoor = 172	2420
SN.node[20].yCoor = 46		SN.node[20].yCoor = 365		SN.node[20].yCoor = 370	
SN.node[21].xCoor = 128	2405	SN.node[21].xCoor = 209	2415	SN.node[21].xCoor = 296	2435
SN.node[21].yCoor = 115		SN.node[21].yCoor = 52		SN.node[21].yCoor = 0	
SN.node[22].xCoor = 251	2405	SN.node[22].xCoor = 370	2405	SN.node[22].xCoor = 336	2415
SN.node[22].yCoor = 204		SN.node[22].yCoor = 46		SN.node[22].yCoor = 104	
SN.node[23].xCoor = 25	2405	SN.node[23].xCoor = 128	2415	SN.node[23].xCoor = 25	2440
SN.node[23].yCoor = 19		SN.node[23].yCoor = 115		SN.node[23].yCoor = 371	
SN.node[24].xCoor = 167	2410	SN.node[24].xCoor = 251	2415	SN.node[24].xCoor = 304	2425
SN.node[24].yCoor = 268		SN.node[24].yCoor = 204		SN.node[24].yCoor = 365	
SN.node[25].xCoor = 164	2405	SN.node[25].xCoor = 25	2415	SN.node[25].xCoor = 209	2435
SN.node[25].yCoor = 394		SN.node[25].yCoor = 19		SN.node[25].yCoor = 52	
SN.node[26].xCoor = 259	2405	SN.node[26].xCoor = 167	2420	SN.node[26].xCoor = 370	2415
SN.node[26].yCoor = 211		SN.node[26].yCoor = 268		SN.node[26].yCoor = 46	
SN.node[27].xCoor = 241	2410	SN.node[27].xCoor = 164	2415	SN.node[27].xCoor = 128	2435
SN.node[27].yCoor = 266		SN.node[27].yCoor = 394		SN.node[27].yCoor = 115	
SN.node[28].xCoor = 303	2405	SN.node[28].xCoor = 259	2405	SN.node[28].xCoor = 251	2415
SN.node[28].yCoor = 377		SN.node[28].yCoor = 211		SN.node[28].yCoor = 204	
SN.node[29].xCoor = 279	2405	SN.node[29].xCoor = 241	2410	SN.node[29].xCoor = 25	2435
SN.node[29].yCoor = 17		SN.node[29].yCoor = 266		SN.node[29].yCoor = 19	
SN.node[30].xCoor = 292	2410	SN.node[30].xCoor = 303	2420	SN.node[30].xCoor = 167	2420
SN.node[30].yCoor = 328		SN.node[30].yCoor = 377		SN.node[30].yCoor = 268	
SN.node[31].xCoor = 345	2410	SN.node[31].xCoor = 279	2415	SN.node[31].xCoor = 164	2425
SN.node[31].yCoor = 351		SN.node[31].yCoor = 17		SN.node[31].yCoor = 394	
SN.node[32].xCoor = 329	2410	SN.node[32].xCoor = 292	2410	SN.node[32].xCoor = 259	2425
SN.node[32].yCoor = 65		SN.node[32].yCoor = 328		SN.node[32].yCoor = 211	

Tabela B.7: Continuação da Tab. B.5.

Área 400x400m 2 canais	MHz	Área 400x400m 4 canais	MHz	Área 400x400m 8 canais	MHz
SN.node[33].xCoor = 143	2405	SN.node[33].xCoor = 345	2405	SN.node[33].xCoor = 241	2430
SN.node[33].yCoor = 249		SN.node[33].yCoor = 351		SN.node[33].yCoor = 266	
SN.node[34].xCoor = 191	2410	SN.node[34].xCoor = 329	2405	SN.node[34].xCoor = 303	2425
SN.node[34].yCoor = 282		SN.node[34].yCoor = 65		SN.node[34].yCoor = 377	
SN.node[35].xCoor = 222	2405	SN.node[35].xCoor = 143	2415	SN.node[35].xCoor = 279	2435
SN.node[35].yCoor = 328		SN.node[35].yCoor = 249		SN.node[35].yCoor = 17	
SN.node[36].xCoor = 307	2405	SN.node[36].xCoor = 191	2420	SN.node[36].xCoor = 292	2430
SN.node[36].yCoor = 27		SN.node[36].yCoor = 282		SN.node[36].yCoor = 328	
SN.node[37].xCoor = 136	2405	SN.node[37].xCoor = 222	2420	SN.node[37].xCoor = 345	2430
SN.node[37].yCoor = 235		SN.node[37].yCoor = 328		SN.node[37].yCoor = 351	
SN.node[38].xCoor = 6	2405	SN.node[38].xCoor = 307	2415	SN.node[38].xCoor = 329	2415
SN.node[38].yCoor = 236		SN.node[38].yCoor = 27		SN.node[38].yCoor = 65	
SN.node[39].xCoor = 252	2405	SN.node[39].xCoor = 136	2405	SN.node[39].xCoor = 143	2435
SN.node[39].yCoor = 99		SN.node[39].yCoor = 235		SN.node[39].yCoor = 249	
SN.node[40].xCoor = 98	2410	SN.node[40].xCoor = 6	2420	SN.node[40].xCoor = 191	2440
SN.node[40].yCoor = 247		SN.node[40].yCoor = 236		SN.node[40].yCoor = 282	
SN.node[41].xCoor = 375	2405	SN.node[41].xCoor = 252	2415	SN.node[41].xCoor = 222	2425
SN.node[41].yCoor = 271		SN.node[41].yCoor = 99		SN.node[41].yCoor = 328	
SN.node[42].xCoor = 154	2410	SN.node[42].xCoor = 98	2410	SN.node[42].xCoor = 307	2435
SN.node[42].yCoor = 382		SN.node[42].yCoor = 247		SN.node[42].yCoor = 27	
SN.node[43].xCoor = 90	2410	SN.node[43].xCoor = 375	2410	SN.node[43].xCoor = 136	2415
SN.node[43].yCoor = 230		SN.node[43].yCoor = 271		SN.node[43].yCoor = 235	
SN.node[44].xCoor = 166	2405	SN.node[44].xCoor = 154	2420	SN.node[44].xCoor = 6	2410
SN.node[44].yCoor = 362		SN.node[44].yCoor = 382		SN.node[44].yCoor = 236	
SN.node[45].xCoor = 216	2410	SN.node[45].xCoor = 90	2420	SN.node[45].xCoor = 252	2435
SN.node[45].yCoor = 290		SN.node[45].yCoor = 230		SN.node[45].yCoor = 99	
SN.node[46].xCoor = 45	2410	SN.node[46].xCoor = 166	2415	SN.node[46].xCoor = 98	2420
SN.node[46].yCoor = 172		SN.node[46].yCoor = 362		SN.node[46].yCoor = 247	
SN.node[47].xCoor = 310	2405	SN.node[47].xCoor = 216	2410	SN.node[47].xCoor = 375	2405
SN.node[47].yCoor = 391		SN.node[47].yCoor = 290		SN.node[47].yCoor = 271	
SN.node[48].xCoor = 353	2410	SN.node[48].xCoor = 45	2410	SN.node[48].xCoor = 154	2420
SN.node[48].yCoor = 145		SN.node[48].yCoor = 172		SN.node[48].yCoor = 382	
SN.node[49].xCoor = 135	2405	SN.node[49].xCoor = 310	2405	SN.node[49].xCoor = 90	2410
SN.node[49].yCoor = 246		SN.node[49].yCoor = 391		SN.node[49].yCoor = 230	

Tabela B.8: Continuação da Tab. B.5.

Área 400x400m 2 canais	MHz	Área 400x400m 4 canais	MHz	Área 400x400m 8 canais	MHz
SN.node[50].xCoor = 309	2405	SN.node[50].xCoor = 353	2405	SN.node[50].xCoor = 166	2440
SN.node[50].yCoor = 287		SN.node[50].yCoor = 145		SN.node[50].yCoor = 362	
		SN.node[51].xCoor = 135	2405	SN.node[51].xCoor = 216	2405
		SN.node[51].yCoor = 246		SN.node[51].yCoor = 290	
		SN.node[52].xCoor = 309	2405	SN.node[52].xCoor = 45	2420
		SN.node[52].yCoor = 287		SN.node[52].yCoor = 172	
				SN.node[53].xCoor = 310	2405
				SN.node[53].yCoor = 391	
				SN.node[54].xCoor = 353	2415
				SN.node[54].yCoor = 145	
				SN.node[55].xCoor = 135	2405
				SN.node[55].yCoor = 246	
				SN.node[56].xCoor = 309	2405
				SN.node[56].yCoor = 287	