**Universidade de São Paulo
Escola de Engenharia de São Carlos**

Tatiana Giorgenon Bonifácio

**Implementação de um protocolo *mesh multi-hop*
baseado em algoritmo de roteamento geográfico para
redes de sensores sem fio.**

**São Carlos
2010**

Tatiana Giorgenon Bonifácio

**Implementação de um protocolo *mesh multi-hop*
baseado em algoritmo de roteamento geográfico para
redes de sensores sem fio.**

Dissertação apresentada como
parte dos requisitos para obtenção
do título de Mestre em Ciências,
Programa de Engenharia Elétrica da
Escola de Engenharia de São
Carlos – Universidade de São Paulo

Área de concentração: Sistemas
Dinâmicos

Orientador:
Prof. Dr. Dennis Brandão

**São Carlos
2010**

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

B715i Bonifácio, Tatiana Giorgenon
Implementação de um protocolo mesh multi-hop
baseado em algoritmo de roteamento geográfico para
redes de sensores sem fio / Tatiana Giorgenon Bonifácio
; orientador Dennis Brandão. -- São Carlos, 2010.

Dissertação (Mestrado-Programa de Pós-Graduação em
Engenharia Elétrica e Área de Concentração em Sistemas
Dinâmicos) -- Escola de Engenharia de São Carlos da
Universidade de São Paulo, 2010.

1. Protocolo de comunicação sem fio. 2. Sensores sem
fio. 3. *Simple MAC*. 4. IEEE 802.15.4. 5. Roteamento
geográfico. I. Título.

FOLHA DE JULGAMENTO

Candidato(a): Engenheira TATIANA GIORGENON BONIFACIO.

Dissertação defendida e julgada em 13/01/2011 perante a Comissão Julgadora:

_____
Prof. Dr. **DENNIS BRANDÃO** – (Orientador)
(Escola de Engenharia de São Carlos/USP)

Aprovada

_____
Prof. Dr. **OMAR CARVALHO BRANQUINHO**
(Pontifícia Universidade Católica/PUC/campus de Campinas)

Aprovado

_____
Prof. Dr. **LEANDRO BUSS BECKER**
(Universidade Federal de Santa Catarina/UFSC)

Aprovada

_____
Prof. Titular **DENIS VINÍCIUS COURY**
Coordenador do Programa de Pós-Graduação em
Engenharia Elétrica

_____
Prof. Associado **PAULO CÉSAR LIMA SEGANTINE**
Presidente da Comissão da Pós-Graduação da EESC

DEDICATÓRIA

Dedico esta pesquisa aos meus pais, César e Tânia, meus maiores tutores da vida, às minhas irmãs, Danielle e Raquel, minhas maiores amigas, e à Tico, fonte insubstituível de carinho e alegria de toda a família.

AGRADECIMENTOS

Aos meus pais, César e Tânia, pelo amor, dedicação e incentivo de todos os dias. Sem o apoio, o carinho e a paciência de vocês no período do mestrado, o caminho até aqui teria sido mais difícil. Muito obrigada!

Às minhas irmãs, Danielle e Raquel, pelo amor, amizade eterna e companheirismo de anos.

Ao Prof. Dr. Dennis Brandão, pela oportunidade de realizar este projeto de pesquisa em que muito aprendi, pelo apoio e orientação sábia, e pela compreensão e paciência nos momentos difíceis.

Ao Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos – Universidade de São Paulo, pelo suporte ao aluno e a qualidade de ensino indiscutível.

Aos amigos de trabalho, Rodrigo Palucci Pantoni, Adriano Ogata e Maurício C. Antunes pela amizade, pelo conhecimento compartilhado e colaboração durante este projeto.

RESUMO

BONIFACIO, T.G. (2010). **Implementação de um protocolo *mesh multi-hop* baseado em algoritmo de roteamento geográfico para redes de sensores sem fio.** 157p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2010.

As tecnologias de redes sem fio passaram a atender não só às aplicações corporativas mais sofisticadas como também àquelas envolvendo desde o rastreamento de animais, monitoramento de estruturas e automação industrial até o gerenciamento de utilidades urbanas. Este projeto de pesquisa aborda o desenvolvimento de um protocolo de rede *mesh multi-hop* segundo a especificação IEEE 802.15.4, com o uso de coordenadas geográficas de cada nó para o roteamento de dados em aplicações de iluminação pública. Para o desenvolvimento do protocolo de rede, é considerada a biblioteca de rotinas de interface de rede denominada *Simple MAC* (SMAC). Tal biblioteca pode ser utilizada para a elaboração do protocolo de roteamento geográfico destinado à rede *mesh multi-hop*. Paralelamente, propõe-se a descrição de um método a ser adotado para a inserção do protocolo de roteamento geográfico no SMAC. Os códigos, desenvolvidos em linguagem de programação C, são portados para a plataforma de desenvolvimento dos nós de comunicação de um *toolkit* específico e submetidos a ensaios experimentais. Os resultados englobam a geração de conhecimento teórico e prático a respeito das redes de sensores sem fio, bem como o desenvolvimento de um *software* aberto para este tipo de rede. As propriedades de auto-organização da rede, robustez e baixa latência são comprovadas nos ensaios experimentais.

Palavras-chave: SMAC, *Simple Mac*, protocolo de roteamento, rede de sensores sem fio, IEEE 802.15.4.

ABSTRACT

GIORGENON BONIFACIO, T.G (2010). **Implementation of a multi-hop mesh protocol based on geographic routing algorithm for wireless sensor networks.** 157p. Thesis (Masters) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2010.

The technologies of wireless networks have to serve not only the most sophisticated business applications but also those that involve the tracking of animals, monitoring of structures, industrial automation and the management of urban utilities. This research project addresses the development of a protocol for multi-hop mesh network according to IEEE 802.15.4, with the use of geographical coordinates of each node for routing data in applications such as street lighting. In order to develop the network protocol, the routines library of the network interface, called Simple MAC (SMAC), is considered. This library can be used for establishing the geographic routing protocol destined for the multi-hop mesh network. In parallel it is proposed to describe a method to be adopted for the inclusion of geographic routing protocol in SMAC. The codes, developed in C programming language, are ported to the development platform of communication nodes of a specific toolkit and subjected to experimental tests. The results include the generation of theoretical and practical knowledge regarding the wireless sensor networks, as well as the development of open software for this type of network. The properties of self-organization of the network, robustness and low latency are proven in experimental tests.

Keywords: SMAC, Simple MAC, routing protocol, wireless sensor network, IEEE 802.15.4.

LISTA DE ILUSTRAÇÕES

Figura 1 - Artigos citados pelo IEEEExplore sobre redes de sensores sem fio	2
Figura 2 - Áreas de aplicação das tecnologias sem fio	12
Figura 3 - Representação de uma rede de sensores sem fio	13
Figura 4 - Transmissão multi-hop em uma rede de sensores sem fio	13
Figura 5 - Hardware básico de um nó sensor.....	14
Figura 6 - Topologias comuns de uma RSSF.....	20
Figura 7 - A pilha de protocolos de um nó sensor	21
Figura 8 - Modo de transmissão FHSS.....	23
Figura 9 - Modo de transmissão DSSS.....	24
Figura 10 - Multiplexação por: (a) divisão de frequência (b) divisão de tempo	25
Figura 11- Estratégias Greedy Routing.....	29
Figura 12 - Estratégias de encaminhamento de pacotes do GPRS	30
Figura 13 - Visão geral das tecnologias sem fio	34
Figura 14 - Modelo IEEE 802.15.4	37
Figura 15 - Seleção dos canais na presença do IEEE 802.15.4 e IEEE 802.11.....	39
Figura 16 - Exemplo de colisão durante CCA na técnica de controle de acesso ao meio CSMA/CA sem janela de tempo	43
Figura 17 - Exemplo de colisão durante transmissão de ACK na técnica de controle de acesso ao meio CSMA/CA sem janela de tempo	43
Figura 18 - Estrutura do superframe utilizado pelo padrão IEEE 802.15.4	43
Figura 19 - Algoritmo slotted CSMA/CA- IEEE 802.15.4.....	45
Figura 20 - Transferência de dados para o coordenador a) com beacon b) sem beacon.....	47
Figura 21 - Transferência de dados do coordenador a) com beacon b) sem beacon.....	48
Figura 22 - Mecanismo de formação da rede definido pelo IEEE 802.15.4	48
Figura 23 - Diagrama de bloco do MC1322X.....	55
Figura 24 - Diagrama de Blocos do SMAC - MC1322X	57
Figura 25 - Estrutura de pacote SMAC	58
Figura 26 - Modelo geral da máquina de estado das mensagens.....	60
Figura 27 - Estrutura de pacote message_t	63
Figura 28 - Algoritmo Greedy baseado na distância Euclidiana.....	64
Figura 29 - Critério Drop.....	65
Figura 30 - Quadro Hello	66
Figura 31 - Quadro de dados.....	67

Figura 32 - Quadro de acknowledgment.....	67
Figura 33 - Declaração dos quadros de mensagens.....	68
Figura 34 - Tabela de mensagens.....	70
Figura 35 - Formação da tabela de mensagens.....	71
Figura 36 - Tabela de vizinhos.....	72
Figura 37 - Formação da Tabela de Vizinhos.....	73
Figura 38 - Fluxograma lógico para repasse do pacote de dados.....	75
Figura 39 - Função de roteamento da mensagem.....	76
Figura 40 - Tempo de envio de um pacote.....	78
Figura 41 - Função enviar mensagem TX.....	80
Figura 42 - Função receber mensagem RX.....	81
Figura 43 - Representação gráfica do cenário de desenvolvimento - Rede A.....	82
Figura 44 - Representação gráfica do cenário de desenvolvimento - Rede B.....	83
Figura 45 - Estrutura Beekit.....	85
Figura 46 - Software Beekit.....	86
Figura 47 - Seleção do Codebase SMAC.....	87
Figura 48 - Seleção de projetos: (a) Template SMAC (b) Adição de um novo projeto.....	87
Figura 49 - (a) Seleção de tipo de hardware (b) Validação do projeto.....	88
Figura 50 - Exportando projeto do Beekit para o IAR – EW.....	88
Figura 51 - Exemplo de código SMAC.....	91
Figura 52 - Código fonte generic_app.c.....	92
Figura 53 - Código generic_app.c alterado – Parte 1.....	94
Figura 54 - Código generic_app.c alterado – Parte 2.....	95
Figura 55 - Código generic_app.c alterado – Parte 3.....	96
Figura 56 - Código generic_app.c alterado – Parte 4.....	97
Figura 57 - Código generic_app.c alterado – Parte 5.....	98
Figura 58 - Analisador de rede.....	99
Figura 59 - Rede de sensores sem fio - A.....	101
Figura 60 - Análise quantitativa das mensagens enviadas - Rede A.....	104
Figura 61 - Visão geral das mensagens enviadas.....	105
Figura 62 - Tempo entre mensagens para o pacote 0.....	106
Figura 63 - Tempo entre o envio de um quadro de dados e de ACK.....	107
Figura 64 - Análise de transmissão do pacote 0.....	108
Figura 65 - Histograma de densidade das mensagens Hello - Rede A.....	109
Figura 66 - Fluxo de mensagens de dados enviadas - Rede A.....	110

Figura 67 - Análise do reenvio de mensagens - pacote 6 - Rede A	111
Figura 68 - Análise do reenvio de mensagens - pacote 11 - Rede A	111
Figura 69 - Fluxo de mensagens ACK enviadas - Rede A	112
Figura 70 - Tempo das mensagens até o destino - Rede A	113
Figura 71 - Qualidade de link - Rede A	114
Figura 72 - Verificação do critério Drop	115
Figura 73 - Fluxo de mensagens ACK enviadas - Critério Drop	116
Figura 74 - Fluxo de mensagens de dados enviadas por fluxo - Rede A	117
Figura 75 - Fluxo de mensagens ACK enviadas por fluxo - Rede A	118
Figura 76 - Tempo das mensagens até o destino por fluxo - Rede A	118
Figura 77 - Visão detalhada dos atrasos - Rede A	119
Figura 78 - Qualidade de link por fluxo - Rede A	119
Figura 79 - Rede de sensores sem fio - B	120
Figura 80 - Análise quantitativa das mensagens enviadas - Rede B	122
Figura 81 - Histograma de densidade das mensagens Hello - Rede B	122
Figura 82 - Fluxo de mensagens de dados enviadas - Rede B	123
Figura 83 - Análise do reenvio de mensagens - pacote 6 - Rede B	124
Figura 84 - Análise do reenvio de mensagens - pacote 11 - Rede B	124
Figura 85 - Fluxo de mensagens ACK enviadas - Rede B	125
Figura 86 - Tempo das mensagens até o destino - Rede B	126
Figura 87 - Qualidade de link - Rede B	127
Figura 88 - Falhas consecutivas - Rede B	128
Figura 89 - Fluxo de mensagens ACK enviadas: Falhas consecutivas - Rede B	128
Figura 90 - Tempo das mensagens até o destino: Falhas consecutivas - Rede B	129

LISTA DE TABELAS

Tabela 1 - Faixas da banda ISM em diversas partes do mundo	22
Tabela 2 - Análise comparativa entre padrões sem fio.....	33
Tabela 3 - Parâmetros da Camada Física: IEEE 802.15.4.....	39
Tabela 4 - Funcionalidade dos Dispositivos FFD e RFD.	40
Tabela 5 - Características do MC1322X.....	56
Tabela 6 - Primitivas de serviços SMAC.....	59
Tabela 7 - Temporizador	77
Tabela 8 - Distância entre nós em função da potência de saída.....	100
Tabela 9 - Identificação dos nós da rede.....	101
Tabela 10 - Tráfego de mensagens	102
Tabela 11 - Falhas introduzidas na Rede A.....	103
Tabela 12 - Fluxo de mensagens na Rede A.....	103
Tabela 13 - Tempo entre tentativas de reenvio - Rede A	111
Tabela 14 - Resumo do comportamento do protocolo SMAC - Rede A.....	112
Tabela 15 - Falhas para verificação do critério Drop	114
Tabela 16 - Fluxo de mensagens na rede A - Critério Drop.....	115
Tabela 17 - Definição dos fluxos de mensagens até o destino	116
Tabela 18 - Ciclo de repetição de cada Fluxo	117
Tabela 19 - Falhas introduzidas na Rede B.....	121
Tabela 20 - Fluxo de mensagens na Rede B	121
Tabela 21 - Tempo entre tentativas de reenvio - Rede B	125
Tabela 22 - Resumo do comportamento do protocolo SMAC - Rede B.....	126

LISTA DE SIGLAS

ADC	ANALOG-TO-DIGITAL CONVERTERS
AES	ADVANCED ENCRYPTION STANDARD
AODV	AD-HOC ON-DEMAND DISTANCE VECTOR
APS	APPLICATION SUPPORT SUB-LAYER
ARQ	AUTOMATIC REPEAT REQUEST
ASM	ADVANCED SECURITY MODULE
BE	BACKOFF EXPONENT
BI	BEACON INTERVAL
BO	BEACON ORDER
CAP	CONTENTION ACCESS PERIOD
CCA	CLEAR CHANNEL ASSESSMENT
CDMA	CODE DIVISION MULTIPLE ACCESS
CFP	CONTENTION FREE PERIOD
CRC	CYCLIC REDUNDANCY CHECK
CSMA	CARRIER SENSE MULTIPLE ACCESS
CSMA-CD	CARRIER SENSE MULTIPLE ACCESS - COLLISION DETECTION
CSMA- CA	CARRIER SENSE MULTIPLE ACCESS - COLLISION AVOIDANCE
CTS	CLEAR-TO-SEND
CW	CONTENTION WINDOW
DMA	DIRECT MEMORY ACCESS
DSDV	DESTINATION-SEQUENCED DISTANCE-VECTOR
DSR	DYNAMIC SOURCE ROUTING
DSSS	DIRECT SEQUENCE SPREAD SPECTRUM
ED	ENERGY DETECTION
ETSI	EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE
FDMA	FREQUENCY DIVISION MULTIPLE ACCESS
FFD	FULL FUNCTION DEVICE
FHSS	FREQUENCY HOPPING SPREAD SPECTRUM
GEAR	GEOGRAPHICAL AND ENERGY AWARE ROUTING
GPS	GLOBAL POSITIONING SYSTEM
GPRS	GREEDY PERIMETER STATELESS ROUTING
GTS	GUARANTEED TIME SLOT
HCF	HART COMMUNICATION FOUNDATION
IEEE	INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
IETF	INTERNET ENGINEERING TASK FORCE
ISA	INTERNATIONAL SOCIETY OF AUTOMATION
ISM	INDUSTRIAL MEDICAL SCIENTIFIC
KBI	KEYBOARD INT
LLC	LOGICAL LINK CONTROL
LEACH	LOW-ENERGY ADAPTIVE CLUSTERING HIERARCHY
LPB	LOW POWER BOARD
LQI	LINK QUALITY INDICATOR
LR-WPAN	LOW-RATE WIRELESS PERSONAL AREA NETWORK
MAC	MEDIUM ACCESS CONTROL
MANETS	MOBILE AD HOC NETWORKS
MCPS	MAC CAMMOM PART SUB LAYER
MFR	MOST FORWARDED WITHIN R
MLME	MAC SUB LAYER MANAGMENT ENTETY
MPR	MAIOR PROGRESSO DIRECIONADO
NCB	NETWORK COORDINATOR BOARD
NFP	NEAREST WITH FORWARD PROGRESS
OLSR	OPTIMIZED LINK STATE ROUTING
OSI	OPEN SYSTEM INTERCONNECTION
OTAP	OVER THE AIR PROGRAMMER
PD	PHYSICAL DATA
PEGASIS	POWER-EFFICIENT GATHERING IN SENSOR INFORMATION SYSTEMS
PIB	PHYSICAL PAN INFORMATION BASE

PLME	PHYSICAL LAYER MANAGEMENT ENTITY
RF	RADIO FREQUENCY
RFD	REDUCED FUNCTION DEVICE
RMST	RELIABLE MULTI-SEGMENT TRANSPORT
RSSF	REDES DE SENSORES SEM FIO
RTS	REQUEST-TO-SEND
RTOS	REAL TIME OPERATING SYSTEM
SAP	SERVICE ACCESS POINTS
SMAC	SIMPLE MEDIUM ACCESS CONTROL
SMP	SENSOR MANAGEMENT PROTOCOL
SO	SUPERFRAME ORDER
SPI	SERIAL PERIPHERAL INTERFACE
SPIN	SENSOR PROTOCOLS FOR INFORMATION VIA NEGOTIATION
SRB	SENSOR REFERENCE BOARD
SSCS	SERVICE SPECIFIC CONVERGENCE SUBLAYER
SSI	SYNCHRONOUS SERIAL INTERFACE
TADAP	TASK ASSIGNMENT AND DATA ADVERTISEMENT PROTOCOL
TDMA	TIME DIVISION MULTIPLE ACCESS
TEEN	THRESHOLD SENSITIVE ENERGY EFFICIENT SENSOR NETWORK
TINYOS	TINY MICROTHREADING OPERATING SYSTEM
UART	UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER
XML	EXTENSIBLE MARKUP LANGUAGE
WLAN	WIRELESS LOCAL AREA NETWORK
WMAN	WIRELESS METROPOLITAN AREA NETWORK
WPAN	WIRELESS PERSONAL AREA NETWORK
WWAN	WIRELESS WIDE AREA NETWORK
WMN	WIRELESS MESH NETWORKS

SUMÁRIO

1 INTRODUÇÃO	1
2 MOTIVAÇÃO	5
3 OBJETIVOS	7
3.1 Geral	7
3.2 Específicos	7
4 ESTRUTURA DA DISSERTAÇÃO	9
5 REVISÃO BIBLIOGRÁFICA	11
5.1 Tecnologias de comunicação sem fio	11
5.2 Rede de Sensores Sem fio (RSSF)	12
5.3 Vantagens das Redes de Sensores Sem Fio	15
5.4 Aspectos relevantes das redes de sensores sem fio	16
5.4.1 Consumo de Energia	16
5.4.2 Tolerância às falhas	17
5.4.3 Sensoriamento	18
5.4.4 Processamento	18
5.4.5 Topologia da Rede	19
5.4.6 Escalabilidade	20
5.5 Arquitetura das redes de sensores sem fio	20
5.5.1 Descrição das Camadas	21
5.6 Padrões IEEE	32
6 IEEE 802.15.4	37
6.1 Camada Física – PHY	37
6.1.1 Parâmetros da Camada PHY	38
6.1.2 Camada de Enlace	39
6.2 Padrões IEEE 802.15.4 emergentes	51
7 TRABALHOS CORRELATOS	53
8 PROPOSTA DE TRABALHO	55
8.1 Sistema de comunicação sem fio	55
8.2 Padrão SMAC - <i>Simple</i> MAC	57
8.2.1 Tipos de Mensagens e Funções SMAC	59
8.2.2 Restrições das Mensagens	61
8.2.3 Estrutura de Pacote <code>message_t</code>	62
8.3 Protocolos de Roteamento	63
8.4 Protocolo Proposto	65
8.4.1 Quadro de mensagens	65
8.4.2 Formação da tabela de vizinhos	71
8.4.3 Critério de Repasse do Pacote	73
8.4.4 Cenário de desenvolvimento	82
9 MÉTODOS	85
9.1 Requisitos Iniciais	85
9.2 Método para implementação SMAC	88
9.3 Método para inserção do protocolo de rede no SMAC	91
10 AVALIAÇÃO DA PROPOSTA	99
10.1 Rede de sensores sem fio - A	100
10.1.1 Verificação do critério <i>Drop</i>	114
10.1.2 Redes A com três fluxos	116
10.2 Rede de sensores sem fio - B	120

ii CONCLUSÕES.....	131
REFERÊNCIAS BIBLIOGRÁFICAS	133

1 INTRODUÇÃO

A comunicação sem fio é um sistema de telecomunicação que utiliza ondas de rádio para transportar sinais e mensagens [1].

Os consideráveis avanços tecnológicos e científicos na última década possibilitaram o uso da comunicação sem fio em redes de sensores sem fio nas mais diversas aplicações, que vão desde o rastreamento de animais, monitoramento de estruturas, automação industrial até o gerenciamento de utilidades urbanas.

As redes de sensores sem fio (*wireless sensor networks* – WSN) são formadas por dispositivos (nós) compactos de dimensões reduzidas, elementos autônomos dotados de circuitos eletrônicos para prover alimentação, sensoriamento, processamento e comunicação via rádio frequência (RF).

Esta tecnologia, considerada hoje emergente, embora pesquisada há mais de uma década, recebeu, nos últimos anos, atenção especial da comunidade científica devido ao potencial apresentado para operar de forma confiável, em condições severas e com altíssima eficiência energética.

Segundo [2], o mercado mundial de redes de sensores sem fio encontra-se ainda em fase inicial e fragmentado em um significativo número de fornecedores e aplicações. Calcula-se que tal mercado movimentou US\$ 159,9 milhões em 2005 e que, em 2012, tal cifra crescerá a US\$ 1.853,9 milhões, contabilizando um crescimento anual de 41,9%.

Estima-se ainda que, em 2012, a participação dos EUA e da Europa neste mercado será de respectivamente 31,2% e 28,9%, com parcela de mesma ordem para Ásia e menos de 10% para o restante do mundo.

O promissor campo utilizando sensores, em específico, redes de sensores sem fio, tem atraído para si um volume de investimentos significativos em pesquisas e tem provado possuir uma ampla área de aplicabilidade.

Para ilustrar tal afirmativa, a Figura 1 apresenta o número de artigos científicos catalogados anualmente na base de dados do IEEE (*Institute of Electrical and Electronics Engineers*), importante entidade de classe internacional para engenharia.

Figura 1 - Artigos citados pelo IEEEExplore sobre redes de sensores sem fio
Consulta em junho de 2010

Os sensores ou nós sensores, como são conhecidos, além de possuírem baixo custo, apresentam ainda um baixo consumo de energia e capacidade de comunicação a curtas distâncias.

Quando combinados, possuem uma grande vantagem em comparação às redes tradicionais, das quais podemos citar a arquitetura flexível em grande escala e alta confiabilidade nas informações coletadas.

No escopo de redes de sensores sem fio, quatro entidades internacionais estão envolvidas na definição de padrões e normas – IEEE, ISA, ETSI e IETF – além de associações de indústrias que definiram alguns dos protocolos atuais como o Zigbee.

O IEEE (*Institute of Electrical and Electronics Engineers*) elabora, tradicionalmente, normas técnicas para as camadas física e de enlace.

Pertinentes às redes de sensores sem fio podem-se destacar as normas IEEE 802.15.4 (usada nos protocolos WirelessHart [3], ZigBee [4] e IETF 6LoWPan), IEEE 802.15.1 (usada no Bluetooth/WiBree) e os protocolos IEEE 802.11x (Wi-Fi).

Dentre estas, a que se consolida com padrão internacional para a redes de sensores sem fio é a IEEE 802.15.4 [5], que atualmente sofre revisões para atender às demandas de aplicações industriais (IEEE802.15.4e¹), como o uso de múltiplos canais

¹ <http://www.ieee802.org/15/pub/TG4e.html>

de frequência de RF (*Frequency Hopping Spread Spectrum* - FHSS) e as de redes de grande escala em cenários urbanos (IEEE 802.15.4g²), como por exemplo as usadas para *smart grid*³.

O IETF (*Internet Engineering Task Force*), importante entidade para a elaboração de padrões de redes de computadores para as camadas intermediárias (camadas 3 a 6 do modelo de referência ISO-OSI), formou, em 2008, o grupo de trabalho ROLL “*Routing Over Low power and Lossy networks*”.

Tal grupo de trabalho definiu os requisitos de redes sem fio de aplicação urbana, ou *Urban WSN* [6], e propôs como protocolo de rede o “6LoWPAN” [7], baseado no protocolo IPv6 (*Internet Protocol* - versão 6) e adequado a redes de nós de baixa capacidade computacional e disponibilidade energética.

O ETSI (*European Telecommunications Standards Institute*) e a ISA (*International Society of Automation*) estão mais voltados para a definição de padrões de sistemas para comunicação M2M (projeto de normalização ETSI BRAN) e de aplicação em automação industrial (comitê ISA100).

Em relação ao uso de sensoriamento sem fio, particularmente em automação industrial, o reconhecimento e adoção de um padrão internacional foram atingidos em abril deste ano, com a aprovação do protocolo WirelessHart como a primeira norma internacional da IEC denominada IEC62591 - *Industrial communication networks - Wireless communication network and communication profiles – WirelessHART*. O setor industrial foi o primeiro dentre os mencionados a convergir para uma norma internacional.

De fato, a especificação IEEE 802.15.4 revolucionou a comunicação nas redes de sensores sem fio e está em evolução. Dentro deste cenário, onde os esforços de padronização internacional começam a render frutos, há ainda muito campo de pesquisa e oportunidades de desenvolvimento e otimização de sistemas baseados em sensores sem fio.

Sendo assim, este trabalho, propõe um estudo de uma nova plataforma (SMAC - *Simple MAC*), também fundamentada no padrão IEEE 802.15.4. É realizado, sobre tal plataforma, o desenvolvimento de um protocolo para a camada de rede baseado em um

---

² <http://www.ieee802.org/15/pub/TG4g.html>

³ *smart grid*: termo utilizado neste contexto para designar sistemas de controle e monitoramento da transmissão e principalmente distribuição de energia elétrica em áreas urbanas.

algoritmo de roteamento geográfico a ser empregado em uma rede de sensores sem fio *mesh multi-hop*, utilizando, para isso, um *toolkit* específico.

2 MOTIVAÇÃO

Embora o campo da tecnologia das redes de sensores sem fio apresentada na Seção 1 esteja em evolução e crescimento, em relação ao desenvolvimento científico e tecnológico neste setor, nota-se que há ainda muitas questões abertas, possibilidades de estudos e aplicações e de transferência de tecnologia das universidades para as empresas.

O número de pesquisas dentro de entidades de ensino relacionadas a redes de sensores sem fio cresceu intensamente nos últimos anos, conforme visto na Figura 1. No meio acadêmico, o estudo das redes sem fio é comumente empregado através de *softwares* de simulação de rede de comunicação, devido principalmente à restrição de equipamentos e tecnologias empregadas no mundo real.

Existem diversos simuladores de redes utilizados pelos pesquisadores, os principais são: J-SIM [8], Omnet++ [9], NS2 [10] e ShoX [11]. Tais simuladores são empregados em testes e validações de algoritmos de roteamento implementados em redes de sensores sem fio, o que possibilita um controle importante sobre os distintos parâmetros que influem a configuração de uma rede

Uma vez que a comunicação sem fio consome muita bateria, nós sensores devem gastar o mínimo de energia possível para transmitir dados [12,13]. Desta forma, a análise do consumo energético dos nós sensores é um quesito amplamente encontrado nos resultados gráficos das simulações de redes de sensores sem fio.

Em [14], [15] e [16] são apresentados respectivamente os protocolos da camada de rede: TEEN (*Threshold sensitive Energy Efficient sensor Network*), PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*) e o protocolo LEACH (*Low-Energy Adaptive Clustering Hierarchy*). Esses protocolos de roteamento são dedicados ao controle da dissipação de energia em uma rede de sensores sem fio e cada um visa ampliar o tempo de vida da rede através de um roteamento estratégico e diferenciado.

No entanto, além da simulação virtual das redes de sensores sem fio, a área acadêmica pode investir no estudo das redes de sensores sem fio através de *kits* de desenvolvimento para aplicações sem fio, fornecidos por diversas empresas e fabricantes. Tais *kits* didáticos, normalmente, contemplam módulos que dispõem de

hardware e *software* necessários para demonstração e implementação de soluções de comunicação sem fio, por exemplo, através do padrão IEEE 802.15.4 e ZigBee.

Desta forma, é possível conhecer e avaliar a solução sem fio existente nos módulos de comunicação do *kit* ou, conforme alguns fabricantes disponibilizam, customizar uma nova solução sem fio através da alteração da pilha protocolar do padrão sem fio presente nos módulo de comunicação.

Como alternativa a essa última análise, seria necessária a aquisição de um código fonte de uma pilha de comunicação sem fio para ser utilizada como base para modificação das camadas protocolares desejadas. A adaptação do código fonte de uma pilha de comunicação ZigBee, principalmente nas camadas de rede e de aplicação, está disponível no mercado, entretanto, tal alternativa é demasiadamente cara.

No entanto, existem protocolos especialmente desenhados para redes de sensores sem fio, que podem ser utilizados para o desenvolvimento de aplicações proprietárias. Assim, não é necessário pagar pela dispendiosa pilha protocolar de comunicação ZigBee para modificá-la. O protocolo SMAC (*Simple MAC*) [17] é um exemplo destes protocolos, possui uma implementação simplificada da camada MAC e permite a manipulação da sua pilha de protocolo.

A preocupação em como transferir um algoritmo de roteamento implementado em um *software* de simulação de redes, por exemplo, NS2, para o protocolo SMAC é significativa na medida em que permite transferir um protocolo de ambiente virtual de simulação de redes para um módulo de comunicação sem fio em ambiente real.

Isto permitirá ensaios práticos que visem testar os protocolos depurados no simulador em ambiente real, onde os principais fatores a serem introduzidos são a instabilidade e a dinâmica da qualidade dos *links* de RF, fatores inerentes à comunicação em RF [18] e de difícil caracterização em ensaios nos simuladores.

A elaboração de uma metodologia para migração dos códigos de simulação para códigos de *firmware* dos módulos de comunicação sem fio, cuja plataforma é o SMAC, traz para pesquisadores a possibilidade de uma nova forma de avaliação e de experimentação na área de WSN.

Acredita-se ainda, que o trabalho aqui proposto possa contribuir para a área estudada e fortalecer laços de cooperação científica entre grupos de pesquisa afins.

3 OBJETIVOS

Tendo em vista que este trabalho tem como propósito vincular seus resultados para auxiliar no desenvolvimento de aplicações de redes de sensores sem fio, o projeto é dedicado a um cenário específico.

Os resultados aqui apresentados irão auxiliar na implementação da camada de rede dos nós de comunicação sem fio do projeto de pesquisa “Redes de Sensores Sem Fio de Aplicação Urbana” do Laboratório de Automação Industrial da EESC-USP. Tal projeto aborda a utilização de um sistema de comunicação sem fio, baseado no protocolo IEEE 802.15.4, para monitoramento e controle de variáveis relacionadas à iluminação pública em áreas metropolitanas. Os objetivos gerais e específicos pertencentes ao escopo deste trabalho são apresentados a seguir:

3.1 Geral

Como o projeto é baseado em aplicações de redes de sensores sem fio fundamentadas na especificação IEEE 802.15.4 e os perfis de aplicação que interagem diretamente com a pilha SMAC e a metodologia de implantação desta, é previsto um estudo detalhado do padrão IEEE 802.15.4 e do protocolo SMAC.

3.2 Específicos

1. Descrição de uma metodologia para implementação do protocolo de rede *mesh* com base no protocolo SMAC (*Simple Mac*).
2. Integrar um algoritmo de roteamento geográfico no protocolo *mesh* citado e que seja adequado para a aplicação de um sistema inteligente de gestão da iluminação pública.
3. Os códigos desenvolvidos em linguagem de programação C deverão ser portados para a plataforma de desenvolvimento dos nós de comunicação de um *toolkit* específico e submetidos a ensaios experimentais para coleta de dados.

Embora a escolha do algoritmo de roteamento seja para um cenário de aplicação que aborda rede de sensores sem fio para controle e monitoramento da iluminação

pública, a metodologia que será desenvolvida neste trabalho é independente da aplicação e do algoritmo de roteamento escolhido.

É prudente permitir a portabilidade da metodologia desenvolvida para aplicações sem fio semelhantes (comercial, residencial, etc.) e que envolvam protocolos da camada de rede diferentes do determinado por este trabalho. Desta forma, outros algoritmos de roteamento vinculado a uma aplicação diferente da aqui apresentada, poderão ser empregados para validar a metodologia desenvolvida.

Os códigos gerados no projeto para a operação dos nós sensores são disponibilizados, nesta dissertação, na forma de código aberto para a equipe do projeto e para a comunidade científica.

4 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está estruturado em onze capítulos, como segue: no capítulo 5 é realizada uma revisão bibliográfica das redes de sensores sem fio abordando suas principais características e arquitetura. O capítulo 6 apresenta uma descrição detalhada das camadas física e de enlace do padrão IEEE 802.15.4. No capítulo 7 é abordada uma breve discussão sobre os trabalhos correlatos á esta pesquisa. O capítulo 8 traz os detalhes da proposta de trabalho. É exibido neste capítulo um estudo do protocolo SMAC e do protocolo de roteamento geográfico, bem como são abordadas as técnicas desenvolvidas para o funcionamento do protocolo proposto. O capítulo 9 envolve uma descrição dos métodos desenvolvidos tanto para implementação do protocolo SMAC como para implementação do protocolo de roteamento geográfico no SMAC. No capítulo 10 é realizada uma avaliação do protocolo proposto. São abordados os resultados e comentários a respeito dos ensaios experimentais realizados nas duas redes de comunicação sem fio propostas. Por fim, o capítulo 11 traz as conclusões gerais do trabalho e as possíveis extensões do protocolo proposto.

5 REVISÃO BIBLIOGRÁFICA

Este projeto de pesquisa insere-se na área relacionada a redes de sensores sem fio. Esta Seção apresenta os conceitos pertinentes e do atual cenário científico e tecnológico na área.

5.1 Tecnologias de comunicação sem fio

De acordo com o alcance e a taxa de transferência empregada nas tecnologias de comunicação sem fio, pode-se destacar a existência de quatro grandes grupos:

- WPAN - *Wireless Personal Area Network* - Neste grupo estão as tecnologias sem fio de pequeno alcance (entre 10 e 100 metros) e baixa taxa de transmissão. É um padrão para redes que interligam dispositivos pessoais ou redes de sensores sem fio, definido pelo IEEE 802.15. Esta área compreende as redes sem fio que utilizam dispositivos como os PDA's, PC's ou periféricos, sendo o campo de tecnologias como o Bluetooth (responsabilidade do subgrupo IEEE 802.15.1) e redes de sensores sem fio (definido pelo IEEE 802.15).

- WLAN - *Wireless Local Area Network* – Aqui estão as tecnologias sem fio destinadas à interligação de redes locais, com alcance entre 100 e 300 metros. Trata-se de padrão implementado como extensão ou alternativa para as redes com cabeamento convencional (par metálico ou fibra óptica).

- WMAN - *Wireless Metropolitan Area Network* - Neste grupo tem-se as tecnologias que tratam dos acessos de banda larga para redes em áreas metropolitanas, com alcance em torno de 6 km;

- WWAN - *Wireless Wide Area Network* - Neste grupo estão as tecnologias voltadas para redes de longa distância em telecomunicações, atendendo aos serviços de voz e alguns serviços de dados.

A Figura 2 mostra a área de aplicação de algumas destas tecnologias com relação ao alcance e à taxa de transmissão. É dentro do grupo WPAN que se encontra o padrão IEEE 802.15.4, concebido para dar suporte às redes de baixas taxas de dados e transmissão de rádio frequência de curto alcance, cenário compatível com as redes de sensores sem fio.

Figura 2 - Áreas de aplicação das tecnologias sem fio

Apontada como uma das principais tecnologias do século XXI [19], este tipo de rede sem fio permite acompanhar, monitorar, estudar, compreender e atuar sobre um determinado fenômeno ou acontecimento.

De acordo com [20], a interconexão de sensores através de redes sem fio, com a finalidade de executar uma tarefa de sensoriamento maior, irá revolucionar a coleta e processamento de informações nos mais diversos tipos de aplicações.

5.2 Rede de Sensores Sem fio (RSSF)

A evolução das redes de sensores sem fio está aliada ao grande avanço ocorrido nas áreas de processadores de baixo consumo, dispositivos de sensoriamento, sistemas embarcados e comunicação sem fio.

Suas perspectivas, a curto prazo, já são de aplicação em distintos cenários, desde o monitoramento de pacientes em clínicas e hospitais, monitoramento ambiental, automação industrial e de serviços urbanos, por exemplo [21].

Estas redes são compostas por um grande número de dispositivos autônomos e compactos, chamados nós sensores [22]. Os nós sensores são dispositivos eletrônicos de baixo custo, baixa potência e com múltiplas funcionalidades.

É função dos nós sensores realizar a coleta e o processamento de dados, além de se comunicarem entre si, usualmente através de um canal RF (rádio frequência), realizando o roteamento das mensagens.

De acordo com a Figura 3, a mensagem percorre a rede sem fio até chegar a um *gateway*⁴ que irá encaminhar esta mensagem, através de uma conexão cabeada, tipicamente *Ethernet*, até uma central de controle (*Host*), onde será possível medir, processar, analisar e apresentar os dados coletados.

Figura 3 - Representação de uma rede de sensores sem fio

Em razão das baixas taxas de transmissão, do curto alcance e da ausência de infra-estrutura pré-existente, as redes de sensores sem fio ficaram conhecidas como redes pessoais sem fio de baixas taxas (*Low-Rate Wireless Personal Area Networks - LR-WPANs* [23]).

A forma de comunicação entre os nós nestas redes é baseada em um esquema de retransmissão *multi-hop*⁵ (múltiplos saltos), apresentada na Figura 4. A maneira como os nós se comunicam com os outros fora de seu alcance de transmissão é por meio da formação de uma cadeia de encaminhamento, na qual os nós fonte e destino estão nas extremidades [24].

Figura 4 - Transmissão *multi-hop* em uma rede de sensores sem fio

⁴ É através do gateway que ocorre a comunicação da rede de sensores sem fio com outras redes.

⁵ *Multi-hop*: baseia-se no conceito de que nós de comunicação intermediários têm a função de receber mensagens de dados de seus vizinhos e retransmiti-los de um nó a outro, até atingir o seu destino.

As redes de sensores sem fio se caracterizam por não se basearem em estrutura ou topologia fixa pré-existente, isso implica que a rede deve se auto-organizar para formar uma entidade coordenada de comunicação. Quando os nós se movem, são desligados ou ligados, entram ou saem da rede, o protocolo de comunicação deve ser capaz de se ajustar a tais mudanças na topologia sem a intervenção humana.

Ademais, grande parcela dos nós pode estar localizada demasiadamente longe do ponto de coleta de dados ou do ponto de acesso de dados para se comunicarem diretamente com estes. Nós sensores imediatamente próximos ao emissor são utilizados, neste caso, para retransmitir as mensagens adiante, utilizando a comunicação *multi-hop*, segundo protocolos de roteamento próprios.

Em termos de *hardware*, um nó sensor é composto de quatro subsistemas: i) um subsistema computacional que consiste de um microprocessador ou micro controlador; ii) um subsistema de comunicação composto de um rádio de alcance curto para comunicação sem fio; iii) um subsistema de sensoriamento que liga o nó ao mundo físico e que consiste de um grupo de transdutores e atuadores; e iv) um subsistema fonte de energia que alimenta o resto do nó, o qual é composto de baterias e um conversor CC-CC [22]. O objetivo deste conversor é prover uma tensão constante para os subsistemas do nó sensor.

Figura 5 - Hardware básico de um nó sensor

A unidade de sensoriamento do subsistema de sensoriamento e atuação é responsável por coletar as informações do ambiente, sendo constituída por um dispositivo-transdutor e um conversor de sinal analógico-digital.

Quando um evento específico é detectado, o conversor analógico-digital converte o sinal analógico, produzido pelo dispositivo, em um sinal digital, o qual alimenta a unidade de processamento do subsistema computacional [25].

A “inteligência” do nó sensor é proveniente da unidade de processamento do subsistema computacional. Esta unidade possui um microprocessador com memória RAM (volátil), que garante ao nó sensor a capacidade de processar e armazenar dados temporariamente.

A unidade de comunicação pertencente ao subsistema de comunicação conecta o nó sensor à RSSF, permitindo a transmissão e recepção de pacotes de dados. O dispositivo que constitui esta unidade, denominado transceptor, pode realizar a comunicação sem fio do nó através de um meio ótico (emissão de raios infravermelhos) ou utilizar frequências de rádio.

A unidade do subsistema de fonte de energia é um dos componentes mais importantes do nó sensor. Algumas baterias que compõem estas unidades de suprimento são recarregáveis (por exemplo, via uso de células solares), porém, a maior parte das arquiteturas de nós sensores utiliza baterias com tempo de vida limitado [26].

5.3 Vantagens das Redes de Sensores Sem Fio

Dentre as principais vantagens de se utilizar uma rede de sensores sem fio sobre uma rede cabeada destacam-se:

- Fácil e rápida instalação da rede. O modo de integração dos dispositivos é simples e fácil, exigindo apenas a fixação dos sensores no processo, não sendo necessária nenhuma estrutura de cabos, caixas de junção e nem de serviços adicionais, tais como projeto de rotas de cabos, revisão de desenhos etc.
- Ausência de desgaste mecânico do meio de transmissão (meio físico é o ar).
- Instalação em áreas de difícil passagem de cabos.
- Baixo custo de instalação com redução de custo considerável por eliminar o custo do cabeamento e demais estruturas e serviços que seriam necessários para instalação de uma rede com fios.

É importante ressaltar que, diferentemente das redes de comunicação cabeadas, da Internet, ou das redes de comunicação industrial, as redes de sensores sem fio devem atender a requisitos particularmente restritivos, no que diz respeito à eficiência energética, escalabilidade, processamento e espaço de memória. Também devem ser capazes de garantir confiabilidade e segurança na transmissão de mensagens por *links*

de rádio frequência instáveis e sujeitos a interferências, ataques intencionais e a obstáculos físicos.

5.4 Aspectos relevantes das redes de sensores sem fio

Pertinente às características particulares das redes de sensores sem fio apontadas na Seção 5.3, os algoritmos e protocolos projetados para redes convencionais não são aplicáveis a estas redes.

Em relação aos desafios e considerações envolvidos no projeto de uma rede de sensores sem fio, é necessário realizar uma análise de aspectos relacionados aos nós sensores.

Tais aspectos envolvem consumo de energia, tolerância a falhas, sensoriamento e processamento de dados, e outros relacionados à estrutura da rede, tais como topologia e escalabilidade. Estes aspectos são abordados a seguir.

5.4.1 Consumo de Energia

Os nós em uma rede de sensores sem fio são equipados com uma fonte de energia limitada. Em uma rede de sensores com múltiplos saltos, cada nó atua como um elemento que origina os dados e também é responsável pelo roteamento.

Esse aspecto pode causar mudanças significativas de topologia e pode exigir uma reorganização da rede. Por essa razão, a conservação de energia apresenta uma importância adicional.

O esforço no desenvolvimento de protocolos visa atender aos requisitos de vida útil da bateria de um sensor, de forma a dispor um mecanismo eficiente e igualitário de consumo de energia entre os nós da rede de sensores [25].

Em relação ao requisito de eficiência energética, por se tratarem de dispositivos alimentados a baterias (ou com alimentação irrestrita em algumas aplicações), o protocolo de acesso ao meio físico (*medium access control* - MAC) deve ser capaz de gerenciar os circuitos de RF de transmissão e recepção, de modo a deixá-los desligados na maior parte do tempo. Os protocolos modernos operam em ciclos de trabalho (*duty cycle*) da ordem de 1% [27].

Embora ligar e desligar os circuitos de RF de um nó seja relativamente fácil, a dificuldade encontrar-se em saber quando os circuitos de RF dos nós vizinhos estarão

ligados, para que se possam transmitir mensagens com sucesso. Para resolver esta questão, faz-se necessário dispor de um sistema de sincronismo e coordenação temporal entre os nós de comunicação [28].

Ainda em relação ao problema da eficiência energética, deve-se utilizar um protocolo de roteamento que seja consciente do nível de energia de cada nó vizinho durante o cálculo da rota de uma mensagem, evitando, assim, o esgotamento precoce da bateria de um nó de que seja sobrecarregado de retransmissões. Mesmo em aplicações onde o fluxo de mensagens seja alto, a carga de comunicação deve ser adequadamente distribuída entre os nós adequados [29].

Dentre os protocolos que adotam estratégias que visem minimizar o consumo energético em redes de sensores sem fio destacam-se o LEACH (*Low-Energy Adaptive Clustering Hierarchy*) [16], o PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*) [15] e o GEAR (*Geographical and Energy Aware Routing*) [30].

5.4.2 Tolerância às falhas

Em uma rede de sensores sem fio, falhas são possíveis e aceitáveis e a rede deve corrigi-las de maneira automática e natural. Nós sensores podem falhar por diversos motivos como falta de energia, falta de visibilidade para outro nó da rede ou até mesmo algum dano físico. Um sistema de tolerância de falhas possui a responsabilidade de manter as funcionalidades de uma rede de sensores sem fio permanentes, sem qualquer interrupção devido à falha de algum sensor [25].

Segundo [26], a falha de um sensor que faz parte da rota de um pacote não deve impedir que o pacote chegue ao seu destino, visto que a topologia da rede deve dar caminhos alternativos para que os pacotes alcancem os seus destinos, seja por meio da adoção de mecanismos de retransmissão de pacotes ou apenas pela duplicação destes pacotes na rede.

Para garantir a disponibilidade de rotas alternativas para os pacotes, os protocolos de roteamento precisam reorganizar a rede periodicamente, traçando novas rotas em substituição àquelas interrompidas pela falha de um nó. Procura-se, assim, uma maior tolerância a falhas, ou seja, é necessário haver mecanismos de auto-organização para que a rede continue a executar a sua função.

5.4.3 Sensoriamento

A função primária de uma aplicação relacionada à rede de sensores sem fio é a atividade de sensoriamento, provida pelos sensores integrantes dos nós da rede [26]. O tipo de informação a ser coletada depende do dispositivo físico de sensoriamento utilizado, que, por sua vez, é escolhido com base no objetivo da aplicação, como, por exemplo, coletar medições de temperatura. No contexto das aplicações de RSSFs, a forma de obtenção dos dados pode ser classificada como:

- (i) Contínua, quando os dados são coletados continuamente;
- (ii) Reativa, quando os dados são fornecidos em resposta a uma consulta do usuário ou a um evento específico do ambiente; e
- (iii) Periódica, quando os dados são coletados segundo condições previamente configuradas na aplicação.

Algumas propostas, como a elaborada no Projeto TinyDB [31], da Universidade da Califórnia, suportam modelos híbridos, que consistem na coexistência de diferentes tipos de coleta de dados.

5.4.4 Processamento

Conforme mencionado na Seção 5.4.1, o consumo de energia com comunicação deve ser prioritariamente minimizado. Tendo em vista que este custo é proporcional ao volume total de dados transmitido ao longo da rede, processamentos aplicados aos dados podem levar a reduções significativas do volume de dados enviado através da rede.

O processamento de dados em nós sensores pode envolver operações de agregação, filtragem e outras técnicas de sumarização. De acordo com [30], o processamento de dados pode envolver não apenas operações aplicadas aos dados coletados do ambiente, mas também à fusão destes dados com aqueles recebidos de outros nós sensores, originando novos pacotes de dados.

A agregação dos dados em rede [32,33] é uma das estratégias mais exploradas na redução do volume de dados transmitido dos nós sensores para a estação-base. Esta estratégia operacional consiste na aplicação progressiva de operações de agregação aos dados, à medida que os pacotes vão passando de um nó para outro na rede.

De qualquer maneira, os nós sensores são componentes muito limitados de memória de processamento. Dessa forma, os programas desenvolvidos devem ser otimizados para apresentar execução eficiente e ocupar reduzido espaço de armazenamento. Essas restrições se fazem importantes devido à grande limitação no consumo de energia no propósito de aumentar a vida útil da bateria, conforme consta na Seção 5.4.1.

5.4.5 Topologia da Rede

Os nós em uma rede de sensores sem fio são tipicamente organizados em três tipos de topologias de redes para interligar o *gateway*, os nós finais e os nós roteadores. Os nós roteadores são similares aos nós finais, já que podem adquirir dados de medição, mas também pode-se utilizá-los para transmitir dados medidos ao longo de outros nós.

A primeira e mais básica topologia é a estrela, na qual cada nó mantém uma única via de comunicação direta com o *gateway* [20]. Esta topologia é simples, mas restringe a distância total que a rede pode alcançar.

Para aumentar a distância que uma rede pode alcançar, pode-se implementar uma topologia *cluster* ou árvore. Nesta arquitetura mais complexa, cada nó mantém um único caminho para o *gateway*, mas podem utilizar outros nós para rotear os dados ao longo desse caminho. Entretanto, a desvantagem desta topologia é que, se um nó roteador perder a comunicação, todos os nós que dependem desse nó roteador perdem sua via de comunicação com o *gateway* [23].

A topologia rede de malha ou *mesh* remedia este problema utilizando vias de comunicação redundantes para aumentar a confiabilidade do sistema. Em uma rede *mesh*, os nós mantêm múltiplas vias de comunicação com o *gateway*, de modo que, se um nó roteador perder a comunicação, a rede automaticamente redireciona os dados por um caminho diferente.

A topologia *mesh*, embora muito confiável, sofre de um aumento na latência da rede, pois os dados devem fazer múltiplos saltos antes de chegarem ao *gateway* [34]. A Figura 6 ilustra os três tipos de topologias de redes abordados nesta seção.

Figura 6 - Topologias comuns de uma RSSF
 Fonte: [34]

Em particular, independente da topologia de rede escolhida, com um número tão alto de nós sensores que devem funcionar sem intervenção e sujeitos a falhas frequentes, a manutenção da topologia da rede é algo fundamental para o seu funcionamento.

Além disso, existem casos em que uma topologia não é conhecida previamente, levando a um arranjo topológico que pode apresentar uma complexidade maior para o roteamento de dados.

5.4.6 Escalabilidade

A ordem de grandeza do número de nós de uma rede de sensores sem fio pode variar das centenas aos milhares. A escalabilidade representa um atributo importante para compor protocolos de redes de sensores sem fio para transmissão de dados e gerenciamento no consumo de energia elétrica.

Uma rede deve ser capaz de manter suas funcionalidades, independente da quantidade de nós presentes nesta rede. Adicionalmente, protocolos de roteamento podem ser escaláveis o suficiente para responder a eventos específicos do ambiente.

5.5 Arquitetura das redes de sensores sem fio

A pilha de protocolos, de acordo com o modelo OSI (*Open Systems Interconnection*), usada pelos nós sensores em uma rede sem fio, é mostrada na Figura 7. Esta pilha de protocolos consiste de uma camada de aplicação, uma camada de transporte, uma camada de rede, uma camada de enlace, uma camada física e planos de gerenciamento de energia, gerenciamento de mobilidade e gerenciamento de tarefas.

Os planos de energia, mobilidade e tarefas monitoram a energia, movimentação e distribuição de tarefas entre os nós sensores. Esses planos ajudam os nós a coordenar as tarefas de sensoriamento e reduzir o consumo total de energia.

Figura 7 - A pilha de protocolos de um nó sensor

O plano de mobilidade detecta e registra movimentos dos nós sensores, de modo que uma rota de volta ao usuário seja sempre mantida. Assim, os nós podem manter uma lista de quem são seus nós vizinhos. Sabendo quem são seus vizinhos, o nó pode balancear o uso de energia e as tarefas executadas.

O plano de gerenciamento de tarefas balanceia e escalona as tarefas de sensoriamento dadas a uma região especificada. Estes planos de gerenciamento são necessários para que os nós possam trabalhar em conjunto de uma forma eficiente quanto ao uso de energia, possam rotear os dados dentro da rede e possam compartilhar recursos entre os nós. Sem eles cada nó trabalharia apenas individualmente. Do ponto de vista da rede de sensores como um todo, é mais eficiente que os nós possam colaborar entre si, de modo que o tempo de vida da rede se prolongue.

5.5.1 Descrição das Camadas

A seguir serão apresentadas as principais características de cada uma das camadas da pilha de protocolos de um nó sensor. Devido ao fato de que este projeto de pesquisa está centrado na camada de rede de um nó sensor, será dada uma ênfase maior nesta camada.

5.5.1.1 Camada Física

A camada física é responsável pela seleção de frequências, geração da frequência portadora, detecção de sinal, modulação e codificação [35].

As faixas de frequência de uso livre são chamadas de ISM (*Industrial, Scientific, and Medical*) como alusão às suas aplicações mais comuns e têm sido amplamente sugerida para redes de sensores sem fio [25].

De acordo com [36], os modos de transmissão amplamente utilizados para a camada física são:

- FHSS - *Frequency Hopping Spread Spectrum* (ISM de 2,4GHz, 1 ou 2Mbps);
- DSSS - *Direct Sequence Spread Spectrum* (ISM de 2,4GHz, 1 ou 2Mbps).

As faixas de frequência ISM não são necessariamente as mesmas em todas as regiões do planeta. Na Tabela 1 são apresentadas as faixas adotadas na Comunidade Européia, EUA, Japão e Brasil [37].

As faixas de frequência para uso com espalhamento espectral foram definidas no Brasil pela ANATEL, através da resolução nº 365 de 10 de maio de 2004 [38].

Tabela 1 - Faixas da banda ISM em diversas partes do mundo

Bandas ISM	EUA	EUROPA	JAPÃO	BRASIL
900 MHz	902 - 928 MHz	902 - 928 MHz	...	902 - 907,5 MHz 915 - 928 MHz
2.4GHz	2.4 - 2.4835 GHz	2.4 - 2.4835 GHz	2.481 - 2.497 GHz	2.4 - 2.4835 GHz
5 GHz	5.15 - 5.35 GHz 5.725 - 5.825 GHz	5.15 - 5.35 GHz 5.47 - 5.725 GHz	5.15 - 5.25 GHz	5.725 - 5.875 GHz

A técnica do espalhamento espectral (*spread spectrum*) se caracteriza por ampla largura de banda e baixa potência de sinal e possui uma série de vantagens em relação ao seu antecessor, a banda estreita (*narrow band*) [39]. Diferentemente da transmissão em banda estreita, o espalhamento espectral, utiliza uma faixa de frequência muito maior do que a necessária para carregar a informação, são menos susceptíveis a interferência e usam menos potência para transmitir um sinal do que a que seria necessária para transmitir o mesmo sinal na banda estreita.

O método FHSS faz parte da técnica *spread-spectrum* que, basicamente, consiste em espalhar a informação por uma banda muito maior do que a necessária para a sua

transmissão. Para tal, o método FHSS divide a banda total em vários canais de pequena largura de banda [40]. Desta forma, transmissor e receptor saltam por estes canais conforme uma sequência pseudo-aleatória conhecida por ambos.

Figura 8 - Modo de transmissão FHSS

Fonte: [41]

Adicionalmente, o FHSS provê segurança na transmissão de dados, uma vez que é necessário conhecer-se tanto a sequência pseudo-aleatória que determina os saltos de frequência, bem como o tempo de permanência em cada frequência, para que se possa escutar inadvertidamente a conversação entre duas estações que utilizem a técnica FHSS [40]. Ou seja, o sinal é recebido por quem conhece a sequência de saltos e aparece como ruído para os outros possíveis receptores.

Já na técnica de espalhamento do modo de transmissão DSSS, o sinal de informação é multiplicado por um sinal codificador com característica pseudo-randômica, conhecido como “*chip sequence*” ou *pseudo ruído* (“*noise*” ou *PN-code*) [41]. Assim, cada bit é representado por múltiplos *chips* utilizando-se o código de espalhamento.

O sinal codificador é um sinal binário gerado numa frequência muito maior do que a taxa do sinal de informação. Ele é usado para modular a portadora de modo a expandir a largura da banda do sinal de radiofrequência transmitido.

Esta modulação aumenta a banda de frequência ocupada pelo sinal, o que significa o espalhamento deste sinal na frequência. No receptor o sinal de informação é recuperado através de um processo complementar, usando um gerador de código local similar e sincronizado com o código gerado na transmissão.

Figura 9 - Modo de transmissão DSSS

Fonte: [41]

Além dessas características, a camada física é responsável por uma série de serviços, tais como a ativação e desativação do rádio, seleção do canal de frequência, transmissão e recepção dos dados, detecção de níveis de energia no canal (ED), medição da qualidade do enlace (LQI) para os pacotes recebidos e verificação da ocupação do canal (CCA) para evitar colisões [42].

5.5.1.2 Camada de Enlace

A camada de enlace tem como responsabilidade a multiplexação dos fluxos de dados, detecção dos quadros, acesso ao meio e controle de erro [43]. Ela garante uma comunicação ponto a ponto e ponto a multiponto em uma rede de comunicação.

Geralmente, a camada de enlace é quebrada em duas subcamadas: a de Controle de Enlace Lógico - LLC (*Logical Link Control*), responsável por efetuar verificações de erro e a Controle de Acesso ao Meio - MAC (*Medium Access Control*), responsável por receber e entregar dados ao meio.

A detecção de erros normalmente acontece pela utilização da técnica CRC (*Cyclic Redundancy Check*), ou seja, a camada de enlace do transmissor adiciona bits redundantes no final de cada quadro e, no receptor, essa camada verifica a consistência desses bits.

Pertence à função da subcamada MAC controlar o acesso ao meio comum a todos os nós. Nas redes de comunicação, frequentemente o tempo de transmissão de um nó em um canal de comunicação é menor do que o tempo em que ele fica ocioso. Dessa forma, é possível compartilhar a frequência e os espaços de tempo disponíveis entre vários nós. Para isso funcionar, é necessário que os nós utilizem um protocolo de acesso múltiplo ao meio, para determinar quando e quem irá transmitir.

Os três protocolos MAC mais comuns e difundidos para controle de acesso ao meio são FDMA, TDMA e CDMA [44].

- **FDMA** (*Frequency Division Multiple Access* - Acesso Múltiplo por Divisão de Frequência): Esse método divide a faixa de frequência disponível em faixas menores e cada nó pode utilizá-la individualmente [43]. Na divisão de frequência, uma faixa de frequência do recurso de comunicação é destinada a um nó por um longo período de tempo ou permanentemente. O recurso de comunicação pode conter simultaneamente diversos sinais separados por faixas de frequência. A Figura 10(a) ilustra esta distribuição.

- **TDMA** (*Time Division Multiple Access* - Acesso Múltiplo por Divisão de Tempo): No compartilhamento dos recursos de comunicação, baseado em divisão de tempo, cada nó tem o direito de utilizar todo o canal por um curto período de tempo, chamado janela de tempo [44]. A Figura 10(b) ilustra um exemplo da utilização de um compartilhamento TDMA, com três nós. Cada nó tem direito a uma janela de tempo. O tempo entre cada janela de tempo é chamado de tempo de guarda, e é utilizado como uma zona de tolerância para reduzir a interferência entre os sinais de duas janelas adjacentes.

- **CDMA** (*Code Division Multiple Access* - Acesso Múltiplo por Divisão de Código): Esse método introduz códigos de pseudo-ruído nos sinais, permitindo que vários nós utilizem a mesma faixa de frequência simultaneamente para transmitir [43]. Basicamente, no método CDMA, cada nó na rede tem um código diferente dos seus vizinhos e pode acessar a rede a qualquer momento, Figura 10(c).

Figura 10 - Multiplexação por: (a) divisão de frequência (b) divisão de tempo (c) divisão de código

Embora existam diversos protocolos MAC, tais protocolos possuem restrições quando aplicados às redes de sensores sem fio [25]. O acesso múltiplo por divisão do tempo - TDMA, por exemplo, exige uma elevada taxa de transmissão e coordenação entre os nós, enquanto o acesso múltiplo por divisão de código - CDMA e os protocolos baseados em contenção da especificação IEEE 802.11 requerem que os nós escutem o canal para receber um possível tráfego, o que provoca um elevado dispêndio de energia.

Tais aspectos implicam em modificações ou desenvolvimento de novos protocolos de acesso ao meio, específicos para redes de sensores sem fio, como o protocolo CSMA (*Carrier Sense Multiple Access*), muito utilizado em redes sem fio, que consiste em escutar o meio antes de transmitir.

Desta maneira, o número de colisões diminui, já que, se o meio está ocupado, o transmissor deverá esperar até o meio ficar disponível [45]. No entanto, ainda há o problema do nó oculto, no qual dois nós não podem ouvir um ao outro, devido, por exemplo, a um obstáculo, e ambos se comunicam com um terceiro nó simultaneamente, ocorrendo uma colisão.

Estudos adicionais para controle de acesso ao meio em redes de sensores sem fio incluem modificações no protocolo tradicional de múltiplo acesso com detecção da portadora (CSMA), proposto por [46], e um esquema híbrido baseado em TDMA/FDMA introduzido por [47].

Como evolução da técnica CSMA, esse protocolo teve o CSMA-CD (CSMA - *Collision Detection*) [48], que detecta uma colisão, e o CSMA-CA (CSMA - *Collision Avoidance*) [44], que procura evitar que as colisões ocorram. Dessa forma, sua eficiência aumentou e tem sido utilizado amplamente em redes de sensores sem fio.

5.5.1.3 Camada de Rede

A principal função da camada de rede é prover o serviço de roteamento. Tal serviço pode ser definido como o processo pelo qual a rede consegue identificar o destinatário das mensagens e encontrar um caminho entre a origem e o destino desta mensagem.

Os algoritmos de roteamento, responsáveis pela determinação das rotas e pelo transporte das mensagens, podem ser simples ou complexos, dependendo do cenário

envolvido na comunicação sem fio. São encontrados na literatura algoritmos de roteamento próprios para redes *Ad Hoc*⁶ e outros específicos para redes *mesh*.

Em redes sem fio *Ad Hoc*, ou MANETs (*Mobile Ad Hoc Networks*) [49], a maioria ou todos os nós movem-se livremente e não necessitam de nenhuma infraestrutura de comunicação. Os nós são conectados diretamente entre eles, atuando como roteadores em uma topologia que tende a mudar com frequência. Cada nó utiliza um protocolo de roteamento, que permite descobrir rotas para qualquer estação da rede. O principal objetivo de uma rede *Ad Hoc* é formar e manter conectados os nós para transportar dados através de múltiplos saltos [50].

As redes sem fio *mesh*⁷ (*Wireless Mesh Networks - WMN*) são consideradas um tipo especial de rede móvel *Ad Hoc* e envolvem diversos nós dispostos, relativamente fixos entre si [51].

Trata-se de uma rede sem fio auto-configurável, que interconecta um conjunto de nós relativamente fixos, capazes de repassar pacotes de dados entre si encaminhando-os ao seu destino através de múltiplos saltos via enlaces sem fio. Por posição relativamente fixa dos nós é entendido que, embora os nós possam não ser absolutamente imóveis, qualquer mudança de posição é limitada dentro de certo alcance que não comprometa a comunicação do nó. A implicação disso é que os caminhos de roteamento podem ser criados e provavelmente ser estáveis. Isto reduz substancialmente a necessidade de sobrecarga de roteamento de mensagens.

O uso das redes sem fio *mesh*, juntamente com as redes de sensores sem fio, está crescendo principalmente em áreas sem infra-estrutura de telecomunicação [52]. De acordo com [53], as redes de sensores sem fio podem ser definidas como uma rede *mesh* constituída de pequenos nós sensores que se comunicam utilizando radiofrequência como meio de comunicação.

Uma parte significativa dos protocolos de roteamento utilizados em redes de sensores sem fio é derivada de algoritmos de roteamento de redes *Ad Hoc*.

Os protocolos para redes *Ad Hoc* são classificados em dois grupos: pró-ativos baseados em tabela e pró-reativos que coletam as informações de roteamento por demanda do emissor. Neste último grupo, cada rota é criada por iniciativa do nó

---

⁶ Em uma rede *Ad Hoc* sem fio a comunicação entre os nós ocorre de forma direta (não há necessidade de encaminhar todas as mensagens para um ponto central) e inexistem requerimentos para qualquer tipo de infra-estrutura ou uma topologia fixa pré-existente.

⁷ O conceito de comunicação onde um nó pode usar todos os seus vizinhos para rotear suas mensagens é chamado de *mesh*.

emissor. Ou seja, somente quando um nó solicita um caminho. O processo se inicia pela etapa de descoberta de rotas e, quando uma rota é estabelecida, esta permanece em manutenção até que o nó destino fique inacessível ou até que a rota não mais seja atual.

O ponto crítico neste tipo de protocolo é o alto custo energético associado ao procedimento de descoberta e manutenção de rotas por meio de mensagens de controle. Os protocolos DSR (*Dynamic Source Routing*) [54] e AODV (*Ad-hoc On-demand Distance Vector*) [55] são exemplos de protocolos reativos.

Os protocolos pró-ativos baseiam-se na utilização de tabelas de rota em cada nó da rede, com informações pertinentes ao roteamento local de mensagens para os demais nós. Estas tabelas permitem que o nó tenha uma visão global da topologia da rede. Um procedimento de atualização periódica para manutenção é necessário para que tais tabelas acompanhem a dinâmica da topologia da rede. As informações tabeladas, pertinentes ao roteamento, podem ser relativas ao posicionamento geográfico dos nós na rede, qualidade do sinal entre nós vizinhos, nível de energia residual dos nós, entre outros [56].

As limitações destes protocolos devem-se à banda utilizada para mensagens de controle e à memória necessária para armazenar as rotas para diferentes destinos, considerando-se redes com milhares de nós que possuem restrições de largura de banda e memória. Os protocolos DSDV (*Destination-Sequenced Distance-Vector*) [57] e o OLSR (*Optimized Link State Routing*) [58] são exemplos de protocolos de roteamento pró-ativos.

Como se pode verificar, algoritmos de roteamento tradicionais possuem altos custos de memória de dados, no caso de protocolos pró-ativos ou custos altos em ocupação de banda para mensagens de controle, no caso de protocolos reativos. Assim, novos protocolos de comunicação mais adequados devem ser propostos, sendo os algoritmos de roteamento geográfico uma das abordagens mais promissoras [59].

Protocolos de roteamento geográfico utilizam informações de localização para calcular uma rota eficiente para um determinado destino. Tais protocolos requerem baixo poder computacional e baixa sobrecarga de comunicação.

Consoantes com tais diretrizes e ao cenário avaliado neste projeto, pesquisas correlatas [60,61] indicam que os protocolos de roteamento mais indicados para redes de sensores sem fio urbanas seriam os baseados em coordenadas geográficas, coordenadas virtuais, distâncias em saltos (*hops*), em custo energético para topologias hierárquicas baseadas em gradiente [62], ou até mesmo baseado em IP [63].

Em relação aos algoritmos que melhor se adaptam ao caso de roteamento baseado em posicionamento geográfico, destacam-se aqueles denominados de *Greedy Packet Forwarding* [64] que trata do repasse criterioso de pacotes. Ao contrário de técnicas como *flooding*⁸, esses tipos de algoritmos não se preocupam em definir uma rota antes de repassar os pacotes. A idéia é que, quando um nó intermediário recebe um pacote de mensagem, deve repassá-lo ao nó adjacente, cujo algoritmo julga estar mais próximo do destinatário final, sem analisar o que há dali para frente. O algoritmo compara o endereço do destinatário (contido na mensagem) com os endereços dos seus vizinhos imediatos e toma a decisão [65].

Na literatura existem inúmeros algoritmos que partem da mesma idéia, dentre eles são destacados o MFR (*Most Forwarded Within R*) [64,65, 66], NFP (*Nearest With Forward Progress*) [64,65], *Compass Rounting* [64, 65, 66,67] e o GPRS (*Greedy Perimeter Stateless Routing*) [68].

No algoritmo MFR, ou seja, maior progressão compreendida em R, o termo R é o raio de alcance de cada nó. Neste critério de roteamento, prioriza-se aquele nó adjacente que possui maior distância do nó intermediário em questão e encontra-se no mesmo quadrante que o destinatário final. Esta técnica tem baixo custo computacional e dá maior peso ao salto de maior progressão.

No entanto, falha quanto ao direcionamento, pois avalia com igual peso todos os nós compreendidos num ângulo de abertura de 90° [65]. Na Figura 11, S é o nó intermediário que recebe um pacote endereçado ao nó D. Segundo a análise pelo critério MFR, o nó escolhido seria C.

Figura 11- Estratégias *Greedy Routing*
Fonte: [65]

No algoritmo NFP, a mensagem é preferivelmente repassada àquele nó adjacente que possui menor distância do nó intermediário em questão e que se encontra no mesmo quadrante do destinatário final. Na Figura 11, este nó seria A. Através de pequenos saltos por vez, consegue-se diminuir bastante a probabilidade de colisões e o gasto de

---

⁸ Pacotes enviados por *broadcast* para todos os destinos com a expectativa de atingir o destino desejado.

energia envolvido na transmissão dos pacotes. No entanto, a velocidade de transmissão é prejudicada, piorando quanto maior for a distância entre remetente e destinatário final [65].

Já o algoritmo *Compass Routing* dá prioridade àqueles nós mais próximos à linha reta que liga o nó intermediário e o nó destinatário final. Este método dá maior peso à direção com que se encaminham os pacotes de mensagem, garantindo a entrega [64]. Na Figura 11, este nó seria B.

Em relação ao protocolo GPRS, cada nó encaminha os pacotes de dados para o vizinho mais próximo do destinatário. Na Figura 12(a), o nó fonte f deseja enviar dados ao nó destino d . O círculo em torno de f indica sua área de alcance de rádio. Assim, o nó m é, entre os vizinhos de f , o mais próximo de d (o arco tracejado indica a distância entre m e d) e é escolhido para encaminhar os dados.

Cada nó conhece a posição de seus vizinhos através do envio de sondas ou através de mensagens de dados. Essa estratégia de encaminhamento possui a vantagem de exigir que cada nó mantenha somente informações acerca de seus vizinhos de um salto, aumentando a escalabilidade do protocolo de roteamento.

Figura 12 - Estratégias de encaminhamento de pacotes do GPRS

Fonte: [68]

Em algumas situações, o encaminhamento *Greedy* atinge um ponto chamado de ótimo local (ou máximo local), no qual o encaminhamento não é possível. A Figura 12(b) apresenta um exemplo de ótimo local. O nó f deseja encaminhar uma mensagem em direção ao nó d , entretanto o nó d não possui nenhum vizinho mais próximo de f que ele mesmo (o arco tracejado descreve a distância entre d e f e o círculo representa a área de alcance de rádio de f). Em situações como essa um mecanismo de recuperação é necessário.

Para tanto, o GPSR possui uma forma de operação chamada de modo perímetro. Ao perceber a ocorrência de um ótimo local, uma marca é adicionada ao pacote indicando a operação no modo perímetro. Em seguida, o pacote é encaminhado segundo

a regra da mão direita, um método utilizado para percorrer grafos apresentado na Figura 12(c). Partindo do nó m em direção a n , o próximo arco que deve ser atravessado é o seguinte no sentido anti-horário a partir do arco (m, n) . Ou seja, o arco (n, o) . O percurso completo pelo grafo é $m \rightarrow n \rightarrow o \rightarrow p \rightarrow m$. No exemplo da Figura 12(b), o percurso de recuperação utilizado é a sequência $f \rightarrow m \rightarrow n \rightarrow o \rightarrow d$.

Além do problema relacionado ao ótimo local em redes de sensores sem fio, a transmissão neste tipo de rede apresenta outros contratempos para o roteamento. Um deles é a existência de enlaces assimétricos, ou seja, enlaces que não possuem necessariamente as mesmas características em ambas as direções. Isso faz com que a recepção de um sinal não forneça informação nenhuma sobre a qualidade da conexão no sentido inverso. Tal fato implica que as informações de roteamento obtidas para um sentido não tenham quase nenhuma utilidade para o outro sentido.

O desempenho da maioria dos algoritmos de roteamento é influenciado pela assimetria de dados, pois prejudica as técnicas de descoberta de vizinhos (roteamento pró-ativo e geográfico) e de caminho reverso⁹ (roteamento sob demanda) [69].

Ainda, os protocolos de roteamento devem considerar as restrições de qualidade de *link* e de energia disponível nos nós, bem como prover um mecanismo de redundância, manutenção e reformulação de rotas para topologias intrinsecamente dinâmicas.

5.5.1.4 Camada de Transporte

A camada de transporte é a parte central de toda a hierarquia de protocolos. Sua tarefa é prover o transporte econômico e confiável de dados. Isto inclui controle de fluxo de dados entre o emissor e o receptor, além da ordenação dos pacotes e correção de erros [70].

Os problemas de transporte seguro e controle de congestionamento ainda são tópicos que necessitam de discussões e propostas de soluções para atender aos requisitos das redes de sensores sem fio. São apontados como uma nova solução para a camada de transporte em redes de sensores sem fio os protocolos ESRT (*Event-to-Sink Reliable Transport in Wireless Sensor Networks*) [71] e o RMST (*Reliable Multi-Segment Transport*) [72].

---

⁹ Os protocolos de encaminhamento que utilizam a técnica do caminho reverso se baseiam na hipótese de que, se há um caminho do nó A para um nó B, existe também um caminho reverso de B para A.

5.5.1.5 Camada de Aplicação

É de responsabilidade da camada de aplicação realizar a interface entre o protocolo de comunicação e o aplicativo que requisitou ou receberá a informação por meio da rede.

Apesar das diversas propostas de aplicações das redes de sensores sem fio, os protocolos para camada de aplicação ainda são pouco explorados e são áreas para novas pesquisas. Dentro dessa perspectiva, em [25] são sugeridos três protocolos para camada de aplicação para as redes de sensores sem fio: o SMP (*sensor management protocol* ou protocolo de gerenciamento de sensores), o TADAP (*task assignment and data advertisement protocol* ou protocolo de designação de tarefas e anúncio de dados) e o SQDDP (*sensor query and data dissemination protocol* ou protocolo de consulta de sensores e de disseminação de dados).

5.6 Padrões IEEE

O IEEE (*Institute of Electrical and Electronics Engineers*) é um órgão regulatório estruturado em grupos de trabalhos para desenvolver padrões de comunicação em redes. O objetivo dessa padronização é possibilitar a compatibilidade entre dispositivos de diferentes fabricantes.

De especial interesse são os grupos de trabalho 802 que tratam de redes locais WLAN (*Wireless Local Area Network*) e redes metropolitanas WMAN (*Metropolitan Area Network*) abordadas na Seção 5.1. Dentro do grupo de trabalho IEEE 802, responsável por toda a área de WLAN, estão subdivididos outros grupos dos quais citamos:

- IEEE 802.3 – Ethernet
- IEEE 802.11 – *Wireless LAN* (Wi-Fi)
- IEEE 802.15 – *Wireless Personal Area Network* (WPAN)
  - IEEE 802.15.1 – Padrão Bluetooth e WiBree
  - IEEE 802.15.4 – Padrão ZigBee e WirelessHart
- IEEE 802.16 – *Broadband Wireless Access* (Wimax)

Devido à grande quantidade de soluções wireless disponíveis como Bluetooth, Wi-Fi, Wimax, é preciso filtrar aquelas que se adéquem melhor às características requisitadas para um determinado projeto. Em aplicações envolvendo redes de sensores

sem fio, devem ser consideradas as restrições e as necessidades específicas que essas redes impõem.

A tecnologia Bluetooth apresenta baixa taxa de transmissão (1Mbps) e baixo alcance, porém, o número de nós em uma rede é muito baixo quando comparado às redes de sensores sem fio onde o número de nós é relativamente alto.

A tecnologia Wi-Fi possui uma maior largura de banda (11/54Mbps) do que a Bluetooth. Entretanto, o consumo de energia é bastante dispendioso e, dessa forma, não apresenta características favoráveis para aplicações em redes de sensores sem fio onde o fornecimento de energia não for atendido.

A tecnologia Wimax apresenta maiores taxas de transmissão (32/132Mbps) e alcance (50km), sendo ideal para aplicações que exigem baixa latência, enlaces de comunicação geograficamente distante e onde o fornecimento de energia não seja um problema, caso que também não se aplica a uma rede de sensores sem fio.

O conceito da padronizada baixa taxa de redes locais sem fio (LR-WPANs) emergiu fundamentado na necessidade de redes de sensores sem fio de baixo custo, possibilitando monitoração e controle de funções não críticas em aplicações como áreas residenciais, comerciais e industriais.

Mediante tal fato, dentro do grupo IEEE 802, há um subgrupo, o 802.15, que cuida de redes pessoais (WPAN – *Wireless Personal Area Network*). E, especificamente, dentro deste, há outro subgrupo em que a preocupação é a criação de redes de sensores com baixo consumo e baixa taxa de transferência de dados, o IEEE 802.15.4, também conhecido como *Low Rate – Wireless Personal Area Network* (LR-WPAN).

Uma análise comparativa entre o padrão IEEE 802.15.4 e os padrões Wi-Fi (IEEE 802.11) e Bluetooth (IEEE 802.15.1) é apresentada na Tabela 2.

Tabela 2 – Análise comparativa entre padrões sem fio

PADRÃO	TAXA DE DADOS	CONSUMO	NÚMERO DE NÓS
Wi-Fi (IEEE 802.11b/g)	54Mbps	Hora	32
Bluetooth (IEEE 802.15.1)	1Mbps	Dias	7
ZigBee (IEEE 802.15.4)	250Kbps	Anos	255-65535

Dentre os padrões, a norma técnica especificada pelo IEEE 802.15.4 é a que foi consolidada como padrão internacional para as redes de sensores sem fio. Tal padrão é particularmente apropriado para as redes de sensores sem fio e apresenta pontos relevantes e pertinentes a essas redes quando comparado com os demais padrões.

Aspectos relacionados ao alcance, taxa de transmissão, consumo energético, custo e complexidade são favorecidos por esta norma e beneficiam as aplicações envolvendo redes de sensores sem fio.

Para ilustrar tal afirmativa, a Figura 13 apresenta uma visão geral dos padrões de tecnologias sem fio considerando tais aspectos. O IEEE 802.15.4 fornece um padrão com características de complexidade ultra-baixa, custo reduzido e uma potência extremamente baixa para a conectividade sem fio entre sensores a curtas distâncias.

Figura 13 - Visão geral das tecnologias sem fio

Diante do exposto, os dispositivos para rede de sensores sem fio disponíveis no mercado evoluem para atender características únicas, como alto grau de integração, baixa largura de banda, curto alcance, memória escassa e limitada capacidade de processamento entre outras características de *hardware* de baixo custo.

Tais dispositivos possibilitaram o uso das redes de sensores sem fio nos mais diferentes tipos de cenários, que vão desde o rastreamento de animais, monitoramento de estruturas, automação industrial até o gerenciamento de utilidades urbanas. Este último cenário pertence ao escopo do projeto de pesquisa “Redes de Sensores Sem Fio

de Aplicação Urbana” do Laboratório de Automação Industrial da EESC-USP e ao qual o projeto aqui apresentado está vinculado.

6 IEEE 802.15.4

O protocolo IEEE 802.15.4 define as características para a camada física e o controle de acesso ao meio das LR-WPAN (*Low-Rate Wireless Personal Area Network*).

Semelhante à arquitetura dos outros padrões, a camada de ligação lógica é formada por duas subcamadas: a subcamada de Controle de Acesso ao Meio ou simplesmente MAC e a subcamada de Controle Lógico do *Link* também chamada de subcamada LLC (*Logic Link Control*) [73].

A LLC é padronizada em 802.2 [74] e é comum entre os padrões 802 bem como a 802.3, 802.11, e 802.15.1, enquanto a subcamada MAC é específica para implementação na camada física [75]. Basicamente a subcamada MAC fornece o elo de ligação entre a camada física e a camada de enlace (SSCS, IEEE 802.2 LLC).

O protocolo IEEE 802.15.4 funciona essencialmente com essas duas camadas. A camada MAC foi projetada para permitir topologias múltiplas com baixa complexidade e a camada física projetada para acomodar as necessidades de interfaces de baixo custo, permitindo níveis elevados de integração [73].

A seguir serão apresentadas as principais características da camada física e a camada de enlace para o protocolo IEEE 802.15.4.

Figura 14 - Modelo IEEE 802.15.4
Fonte: [5]

6.1 Camada Física – PHY

A camada física é responsável por uma série de serviços, tais como a ativação e desativação do *transceiver*¹⁰, seleção do canal de frequência, transmissão e recepção dos

¹⁰ Equipamento que é transmissor e receptor.

dados, detecção de níveis de energia no canal (ED - *Energy Detection*), medição da qualidade do enlace (LQI - *Link Quality Indication*) para os pacotes recebidos e verificação da ocupação do canal (CCA - *Clear Channel Assessment*) para gerenciamento de acesso ao canal via CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*) [76]. A mesma provê uma interface entre a subcamada MAC e o canal do rádio físico (*firmware*¹¹ e *hardware*).

A medida da detecção de energia do canal (ED) é usada pela camada de rede como parte do algoritmo de seleção de canal. Ela corresponde a uma estimativa da potência do sinal recebido dentro da largura de banda do canal IEEE 802.15.4.

A medida LQI é uma caracterização da intensidade e/ou qualidade do pacote recebido. A medida pode ser implementada usando a medida ED, uma estimativa da relação sinal/ruído ou uma combinação desses métodos. O LQI é reportado como um inteiro de 8 bits. Os valores máximos e mínimos de LQI são associados aos valores de mais baixa e alta qualidade dos sinais IEEE 802.15.4 detectáveis pelo receptor, e os outros valores estariam uniformemente distribuídos entre esses dois limites.

O CCA é executado de acordo com a configuração de um dos métodos descritos abaixo:

- Energia acima do nível: CCA reportará o estado do meio como ocupado após detectar um nível de energia acima do nível ED;
- Detecta somente a portadora: CCA reportará o estado do meio como ocupado após a detecção do sinal da portadora. Este sinal pode estar acima ou abaixo do nível ED;
- Detecta portadora com energia acima do nível: CCA reportará o estado do meio como ocupado após a detecção da portadora com energia acima do nível ED.

6.1.1 Parâmetros da Camada PHY

A camada física do padrão IEEE 802.15.4 trabalha sobre as bandas de frequência ISM (*Industrial Medical Scientific*) e pode operar em 3 faixas de frequência: 2450 MHz (amplamente utilizada e adotada pelo Brasil) com 16 canais, 915 MHz (apenas nos USA) com 10 canais e 868 MHz com apenas 1 canal (padrão Europeu) [76].

---

¹¹ *Software* que controla diretamente o *Hardware*.

Tabela 3 - Parâmetros da Camada Física: IEEE 802.15.4

Banda de Frequência	Cobertura	Taxa de transmissão (kbps)	Número de Canais
868MHz	Europa	20	1
915 MHz	ISM Américas	40	10
2400MHz	ISM Mundial	250	16

São alocados pelo padrão IEEE 80.15.4 o total de 27 canais com três diferentes taxas de dados: 16 canais com uma taxa de dados de 250 Kbps em 2.4 GHz, 10 canais com uma taxa de dados de 40 Kbps na banda de 915 MHz e 1 canal com uma taxa de dados de 20 Kbps na banda de 868 MHz.

A forma com que tais canais estão organizados permite a coexistência do padrão IEEE 802.15.4 com outros padrões, como por exemplo, o IEEE 802.11, em um mesmo ambiente [76]. Um exemplo desta compatibilidade é descrita na Figura 15. É observado que, apesar do IEEE 802.11 utilizar uma porcentagem grande da faixa de frequência disponível, a utilização dos canais 15, 20, 25, 26 do IEEE 802.15.4 ainda é permitida.

Figura 15 - Seleção dos canais na presença do IEEE 802.15.4 e IEEE 802.11

Todas as frequências utilizadas pelo padrão IEEE 802.15.4 suportam espalhamento espectral com seqüenciamento direto - DSSS (*Direct Sequence Spread Spectrum*) para o modo de acesso ao canal de comunicação. Esta técnica de espalhamento espectral foi apresentada na Seção 5.5.1.1.

6.1.2 Camada de Enlace

De acordo com a Figura 14, a camada de enlace do padrão IEEE 802.15.4 é dividida em duas subcamadas: a MAC e a LLC. A camada responsável pelo controle de *link* lógico (LLC) é padrão para todas as redes sem fio, enquanto que a camada MAC relaciona-se com o *hardware* com o qual ela vai trabalhar [77].

Mediante isso, são apresentadas, a seguir, as características particulares da subcamada MAC pertinentes à norma técnica IEEE 802.15.4.

6.1.2.1 Dispositivos da rede LR-WPAN

São definidos pela subcamada MAC dois tipos básicos de dispositivos, o RFD (*reduced function device*) e o FFD (*full function device*) [78].

Os dispositivos RFD são equipados com um conjunto limitado de funcionalidade da subcamada MAC e podem funcionar apenas como dispositivos finais que coletam dados e transmitem a informação para seus vizinhos [79]. Sua comunicação é restrita a dispositivos FFD.

Os RFD são dispositivos normalmente simples, onde a comunicação é feita apenas entre dois nós e, por serem impossibilitados de fazer roteamento, são utilizados nas margens das redes.

Por outro lado, dispositivos FFD são equipados com um conjunto completo de funcionalidades da subcamada MAC. São dispositivos de construção mais complexa e podem atuar como dispositivos finais, roteadores ou coordenadores da rede.

Uma rede de sensores sem fio consiste de múltiplos dispositivos FFD e RFD, com um dos FFD designado como coordenador da rede. O resumo das funcionalidades de cada um desses dispositivos em uma rede sem fio é apresentado na Tabela 4.

Tabela 4 - Funcionalidade dos Dispositivos FFD e RFD.

DISPOSITIVO	TIPO DE DISPOSITIVO FÍSICO (IEEE 802.15.4)	FUNÇÃO
Coordenador	FFD	Criação e formação da rede, atribui endereços. Existe um por rede.
Roteador	FFD	Permite que mais nós se juntem à rede ao aumentar seu alcance. Efetua funções de controle e monitoramento. Sua existência é opcional.
Dispositivo final	RFD ou FFD	Efetua ações de controle ou monitoramento através de um dispositivo que lhe seja associado (sensor, controlador, atuador, etc).

Tais dispositivos combinam entre si para formar um dos três diferentes tipos de topologias de rede suportada pelo padrão IEEE 802.15.4: estrela, árvore ou a *mesh*. A Figura 6 da Seção 5.4.5 aborda as características de cada uma dessas topologias em conjunto com os dispositivos FFD e RFD.

Todos os dispositivos operando na rede, em qualquer topologia, terão, cada um, um único endereço estendido de 64 bits. Entretanto, um dispositivo RFD pode obter endereços de apenas 16 bits do coordenador da rede (FFD).

6.1.2.2 Mecanismo de Acesso ao Meio

A subcamada MAC trata de todo acesso ao canal de rádio físico e é responsável pelas seguintes tarefas: geração e sincronização de *beacons*; suporte de associação e desassociação na rede; suporte opcional à segurança do dispositivo; gerenciamento de acesso ao canal via CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*); manutenção dos tempos reservados (*slots* GTS) e validação e reconhecimento de mensagem [79].

Os *beacons* são pacotes de controle que delimitam os quadros utilizados pelo coordenador para sincronizar com os demais dispositivos da rede [78]. Uma rede pode ser configurada com *beacon* habilitado ou desabilitado.

No modo de operação com *beacon* desabilitado, o padrão IEEE 802.15.4 utiliza a técnica *unslotted* CSMA/CA (Acesso múltiplo com detecção de portadora sem janelas de tempo) para controle de acesso ao meio. Em tal técnica, uma transmissão inicia com a espera por um tempo aleatório de períodos de *backoff* entre 0 e $2^{BE} - 1$, onde *BE* (*Backoff Exponential*) pode ter um valor entre *macMinBE* e *macMaxBE* (por padrão, 3 e 5 respectivamente) [80].

Um período de *backoff* é o tempo necessário para transmitir 20 símbolos. Um símbolo equivale a 4 bits, se utilizar o modo de operação da camada física em 2.4GHz [69]. Uma vez que o tempo selecionado expira, o nó verifica se o canal está disponível para transmissão (CCA – *Clear Channel Assessment*).

Esse procedimento é executado no tempo de 8 símbolos. Se o canal estiver ocupado, o nó incrementa *BE* (limitado a *macMaxBE*) e repete o processo de espera. O limite de tentativas de acesso ao canal é determinado por *macMaxCSMABackoffs* (4 tentativas por padrão).

Se, após essas tentativas, o nó não conseguir ter acesso ao canal, é declarada uma falha de acesso ao canal. Se o canal estiver livre, executa-se o procedimento de troca de recepção para transmissão e transmite-se o pacote. O tempo de execução da troca de modo do rádio é de 12 símbolos.

Na transmissão de um pacote, pode ser solicitada a confirmação de recebimento (ACK - *acknowledgement*). O procedimento de acesso ao canal não é realizado para envio do ACK, ou seja, o nó que recebeu o pacote simplesmente executa a troca de recepção para transmissão e envia imediatamente a confirmação.

Se ocorrer uma falha na recepção do ACK, o nó que enviou o pacote tentará transmiti-lo novamente, após a espera de um tempo conhecido por *macAckWaitDuration* (54 símbolos no modo 2.4GHz). É declarada uma falha de transmissão por colisão se o reconhecimento não for recebido após um número de tentativas determinado por *macMaxFrameRetries* (3 tentativas por padrão).

De acordo com [69], verifica-se que as características do *unslotted* CSMA/CA tornam a rede mais susceptível a colisões e incita uma maior disputa pela utilização do meio do que em modos que utilizam sinal piloto para sincronização.

Além disso, dispositivos IEEE 802.15.4 operando em *unslotted* CSMA/CA deixam de escutar o canal durante a inversão do rádio, havendo riscos de colisões. Estas colisões podem ocorrer no início da transmissão de um pacote ou na transmissão do pacote de reconhecimento.

Como apresentado na Figura 16, o tempo de verificação de atividade no canal somado ao tempo de inversão de rádio é de 20 símbolos. A colisão no início da transmissão ocorre se outro dispositivo iniciar o procedimento de CCA durante os 12 primeiros símbolos deste processo, pois, se isto ocorrer, o procedimento de CCA será completado com o canal ainda livre.

Uma colisão na transmissão do pacote de reconhecimento é exemplificada na Figura 17. Como a duração de um CCA (8 símbolos) é inferior ao tempo de inversão do rádio (12 símbolos), existe uma janela de colisão de 4 símbolos. Assim, se um nó iniciar o procedimento de CCA nesta janela para transmissão de um ACK, ocorrerá uma colisão.

Figura 16 - Exemplo de colisão durante CCA na técnica de controle de acesso ao meio CSMA/CA sem janela de tempo

Fonte: [69]

Figura 17 - Exemplo de colisão durante transmissão de ACK na técnica de controle de acesso ao meio CSMA/CA sem janela de tempo

Fonte: [69]

Em uma rede com o modo *beacon* habilitado, o coordenador da rede transmite um *beacon*, periodicamente, no qual os outros dispositivos o usam para sincronização e para determinação de quando estão liberados para transmissão e recepção de mensagens.

A mensagem *beacon* é usada para definir uma estrutura chamada de *superframe* em que todos os nós na rede serão sincronizados [5]. Esta estrutura é mostrada na Figura 18.

Figura 18 - Estrutura do *superframe* utilizado pelo padrão IEEE 802.15.4

Fonte: [5]

A estrutura *superframe* é limitada pelos *beacons*. Na mesma existe um período ativo durante o qual a comunicação se processa e um período inativo (opcional), durante o qual os dispositivos podem desligar seus transceptores para conservar energia. O período ativo é dividido em 16 períodos de tempo iguais (*slots*).

Imediatamente seguindo o *beacon* vem o período de acesso de contenção (CAP - *Contention Access Period*). Durante este período os dispositivos devem se comunicar usando o mecanismo *slotted* CSMA-CA. Esta técnica é similar ao *unslotted* CSMA-CA, exceto pelo fato de que os períodos de *backoff* são alinhados com os limites dos *slots* [81].

O CAP deve conter, pelo menos, oito períodos ativos, mas pode chegar a até 16. Seguindo o CAP, temos um opcional período livre de contenção (CFP - *Contention Free Period*), que pode ter até sete períodos ativos.

Em um CFP o coordenador reserva períodos (GTS – *Guaranteed Time Slot*) para um determinado dispositivo. Durante um GTS, um dispositivo tem acesso exclusivo ao canal e não executa CSMA-CA [78].

A estrutura do *superframe* é definida pelos valores dos atributos *macBeaconOrder* (BO) e *macSuperframeOrder* (SO) [82]. O intervalo de tempo no qual o coordenador transmitirá seus quadros *beacon* é definido como BI (*Beacon Interval*) e se relaciona com o parâmetro BO (*macBeaconOrder*) da seguinte maneira:

$$BI = aBaseSuperframeDuration * 2^{BO} \text{ simbolos} \quad (1)$$

$$0 \leq BO \leq 14$$

Em que $aBaseSuperframeDuration = 960$ símbolos.

Se BO=15, o *superframe* não existirá e o valor do *macSuperframeOrder*, SO, será ignorado. O atributo MAC *macSuperframeOrder* (SO) define o comprimento da porção ativa do *superframe* [81]. O intervalo de tempo relativo à parte ativa do *superframe* SD (*Superframe Duration*) se relaciona com o parâmetro SO (*macSuperframeOrder*) da seguinte maneira:

$$SD = aBaseSuperframeDuration * 2^{SO} \text{ simbolos} \quad (2)$$

$$0 \leq SO \leq BO \leq 14$$

Para valores iguais de SO e BO, a duração do *superframe* será igual à duração do *beacon* e, nesse caso, não existirá a parte inativa do *superframe*. A porção ativa de cada *superframe* possui uma duração de $aBaseSlotDuration * 2^{SO}$, onde $aBaseSlotDuration = 60$ símbolos.

Consoante a tais diretrizes, as redes que optarem por usar a estrutura *superframe* configurarão o atributo *macBeaconOrder* (BO) com um valor entre 0 e 14 e o *macSuperframeOrder* com um valor entre 0 e o valor de *macBeaconOrder*. Além disso, tais redes utilizarão a técnica *slotted CSMA/CA* durante o período CAP para enviar seus quadros.

O algoritmo *slotted CSMA/CA* também é baseado em uma unidade de tempo chamada de *Backoff Period* (BP), que é igual à $aUnitBackoffPeriod = 80\text{bits}$ (0.32ms). Este algoritmo depende principalmente de três variáveis: (1) o *Backoff Exponent* (BE) que permite a computação do atraso *backoff*, (2) a *Contention Window* (CW) que representa o número de BPs durante o qual o canal deve ser sentido inativo antes de acessar o canal, (3) o *Number of Backoff* (NB) representa o número de vezes que o algoritmo CSMA/CA foi requisitado enquanto tentava acessar o canal [83]. A Figura 19 apresenta o fluxograma do algoritmo *slotted CSMA/CA*.

Figura 19 - Algoritmo *slotted CSMA/CA*- IEEE 802.15.4

Fonte: [83]

Primeiramente, o *Number of Backoff* e a *Contention Window* são inicializados (NB = 0 e CW = 2). O *Backoff Exponent* é também inicializado para BE = 2 ou BE =

$\min(2, macMinBE)$, dependendo do valor da extensão de vida da bateria. O $macMinBE$ é uma constante e tem valor padrão igual a 3.

Em seguida, o algoritmo também inicia um contador de número randômico de BP uniformemente gerado entre 0 e $2^{BE} - 1$. O contador deve iniciar no limite do BP para garantir sincronismo. Quando o cronômetro expirar, o algoritmo executa a operação CCA aos limites BPs para avaliar a atividade do canal. Se o canal estiver ocupado, CW é reinicializado para $CW_{ini} = 2$, e os parâmetros NB e BE são incrementados. BE não deve exceder $aMaxBE$ (valor padrão fixado em 5).

Se o número máximo de *backoff* ($NB = macMaxCSMABackoffs = 5$) é alcançado, o algoritmo informa uma falha para as camadas de cima. Caso contrário, volta para o *Step 2* e a operação *backoff* é reiniciada. O protocolo também permite um número de tentativas depois de cada fracasso que corresponde ao $aMaxFrameRetrie$ (valor padrão igual a 3).

Uma vez que o canal estiver ocioso, CW é decrementado. A operação CCA é repetida se CW for diferente de 0. Se o canal é sentido novamente como inativo, o nó tenta transmitir desde que o restante do BPs no atual CAP seja suficiente para transmitir o quadro e o reconhecimento subsequente. Caso contrário, o CCA e o quadro de transmissão são ambos diferidos para o próximo *superframe*. Isto é chamado de deferência de CCA.

Mediante a isso, em redes com quadros de sinalização para sincronismo, o mecanismo *slotted CSMA/CA* deve ser criteriosamente avaliado, uma vez que seu comportamento é afetado por seus parâmetros de inicialização. Tais parâmetros correspondem aos: (1) o mínimo *backoff exponent* ($macMinBE$), (2) o máximo *backoff exponent* ($aMaxBE$), (3) o valor inicial do CW (CW_{init}) e (4) o número de máximo de *backoffs* ($macMaxCSMABackoffs$) [80].

Embora as redes de sensores sem fio possam optar por este modo de comunicação com sinal de sincronização, a norma técnica elaborada pelo IEEE 802.15.4 não oferece um suporte a este modo de comunicação.

6.1.2.3 Sentido de Comunicação

São definidos três sentidos de comunicação pelo padrão IEEE 802.15.4 para as redes de sensores sem fio nos modos de operação com ou sem a estrutura do *superframe* [78].

No primeiro caso, para o modo de operação com a estrutura do *superframe* habilitada, quando um dispositivo desejar enviar dados para um coordenador, primeiramente deve-se esperar o recebimento de um *beacon* para que ocorra a sincronização com o *superframe* [78]. Em um tempo apropriado, a informação coletada na rede é transmitida para o coordenador usando o *slotted CSMA-CA*. O coordenador, opcionalmente, pode enviar uma mensagem de reconhecimento para garantir que o pacote chegou corretamente. Este procedimento é ilustrado na Figura 20 (a).

Entretanto, quando a estrutura do *superframe* não é habilitada, um dispositivo não necessita esperar o recebimento de um *beacon* para enviar uma informação [73]. Em um tempo apropriado, a informação é transmitida utilizando o *unslotted CSMA-CA*, conforme descrito na Figura 20 (b).

Figura 20 - Transferência de dados para o coordenador a) com *beacon* b) sem *beacon*
Fonte: [5]

No segundo caso, para o modo de operação com a estrutura do *superframe* habilitada, quando um coordenador desejar enviar dados para um dispositivo na rede ele indica explicitamente no *beacon* sua intenção de transmissão [78]. Ao receber um *beacon*, um dispositivo de rede percebe que uma mensagem está pendente e, dessa forma, envia uma requisição para o coordenador, autorizando-o a transmitir a informação.

Ao receber a autorização, o coordenador envia uma mensagem de reconhecimento informando que a autorização chegou corretamente e, em seguida, a informação pendente é transmitida utilizando-se *slotted CSMA-CA*. Este procedimento é ilustrado na Figura 21 (a).

Entretanto, quando a estrutura do *superframe* não está habilitada, o procedimento é ligeiramente diferente. No caso, dispositivos de rede são configurados para enviarem mensagens periodicamente ao coordenador para saber se informações

estão pendentes. Caso alguma informação esteja pendente, o coordenador as envia para os dispositivos conforme a descrição da Figura 21 (b).

Figura 21 - Transferência de dados do coordenador a) com *beacon* b) sem *beacon*
Fonte: [5]

Outra atribuição do padrão IEEE 802.15.4 relaciona-se com os mecanismos necessários para a formação da rede. Mesmo que o modo de transmissão baseado na estrutura do *superframe* esteja desabilitado, na fase de formação da rede as trocas de mensagens utilizam o *superframe* [83].

O mecanismo é inicializado após o coordenador ser configurado. Este dispositivo servirá de destino para os pacotes gerados na rede. Após um nó sensor ser inicializado, uma verificação nos canais de comunicação é realizada com a finalidade de encontrar *beacons* enviados pelo coordenador principal ou por outros dispositivos que já conseguiram entrar na rede.

No momento que um dispositivo encontra um coordenador da rede, uma série de mensagens é trocada, visando a entrada na rede por parte do dispositivo, conforme ilustrado na Figura 22.

Figura 22 - Mecanismo de formação da rede definido pelo IEEE 802.15.4
Fonte: [5]

Inicialmente, o dispositivo envia uma mensagem ao coordenador solicitando a entrada na rede. O coordenador, ao receber esta mensagem, responde com uma mensagem de reconhecimento. Nesse momento, o coordenador tem um tempo limite (*macResponseWaitTime*) para verificar a possibilidade da entrada do dispositivo na rede [73].

Por outro lado, quando o dispositivo recebe a mensagem de reconhecimento, um atraso de (*macResponseWaitTime*) tempo também ocorre. Após esse tempo, o dispositivo envia uma requisição de dados com a finalidade de confirmar a disponibilidade do coordenador em adicionar o dispositivo na rede. Em seguida, mensagens de confirmação são enviadas pelo coordenador para confirmar a entrada na rede.

6.1.2.4 Serviços da Camada de Enlace

A camada MAC é responsável por estabelecer uma interface de comunicação entre a subcamada SSCS¹² (*Service Specific Convergence Sublayer*) e a camada física, além de fornecer serviços às camadas superiores (LLC - IEEE 802.2) através da supracitada SSCS, ou diretamente, através de outros tipos de LLC proprietárias [84]. Essa interface é possível devido a dois serviços, conhecido como SAP (*Service Access Points*), que são:

- MCPS-SAP (*MAC Cammom Part Sub Layer - Service Access Points*).
- MLME-SAP (*MAC Sub Layer Managment Entety - Service Access Points*).

Cada camada disponibiliza vários serviços, tanto para as camadas inferiores e suas subcamadas, assim como para as camadas superiores. Os serviços são baseados em primitivas que executam funções específicas para cada camada.

As primitivas efetuadas pelo MLME realizam as funções de gerenciamento e as primitivas efetuadas pelo MCPS figuram as relações de troca de dados. Estas primitivas são importantes, pois fazem a interação com as entidades responsáveis por certas funções similares nas camadas inferiores e nas camadas superiores, podendo trocar informações entre si [73].

---

¹² A camada SSCS é uma interface para a camada de controle de enlace lógico (LLC – IEEE 802.2), subcamada MAC, e as camadas superiores.

A camada física interage com a camada MAC por meio de suas primitivas: Serviços de Dados, providos por intermédio do PD-SAP (*Physical Data - Service Access Point*), e de Serviço de Gerência através do seu PLME-SAP (*Physical Layer Management Entity - Service Access Point*), provido pela entidade de gerência da camada (PLME), conforme visto na Figura 14.

Pelo PD-SAP é disponibilizada a primitiva PD-DATA, responsável pela troca de dados entre camadas MAC pares. E pelo PLME-SAP são disponibilizadas as primitivas: PLME-CA (informa a avaliação de canal vazio), PLME-ED (detecção de energia), PLME – GET/SET (escrita e leitura de atributos do PIB), e PLME-TRXENABLE (habilita ou desabilita o *transceiver*). Faz parte da camada física também, o PIB (*Physical PAN Information Base*), que é a base de informações necessárias para o correto funcionamento da camada e se encontra sob domínio do PLME [85].

Em suma, a subcamada MAC é dividida em duas subcamadas que se comunicam entre si; a MCPS (*MAC Cammom Part Sub Layer*) e a MLME (*MAC Sub Layer Managment Entety*) que acedem respectivamente à camada física inferior pelo PD-SAP (*Physical Data Service Acess Poin*) e pelo PLME (*Physical Layer Managment Entity*).

No padrão IEEE 802.15.4, as primitivas são especificadas tanto na camada física como na camada de ligação lógica; e somam um total de 49, das quais 35 pertencem à camada lógica, especificamente à subcamada MAC, e 14 ficam localizadas na camada física [73].

Das 35 primitivas da subcamada MAC as principais são as seguintes:

- As de serviços de dados MAC:

o MCPS-DATA: troca de pacotes de dados entre MAC e camada superior (SSCS).

- As de serviços de gerenciamento MAC:

- o MLME-ASSOCIATE/DISASSOCIATE: (Des)associa em rede;
  - o MLME-SYNC / SYNC-LOSS: (Des)sincronização do Dispositivo;
  - o MLME-SCAN: verifica os canais de rádio;
  - o MLME-GET / SET: recupera/escreve os parâmetros MAC PIB;
  - o MLME-START / BEACON-NOTIFY: gerenciamento de *beacon*;
  - o MLME-POLL: sincronização sem *beacon*;
  - o MLME-GTS: gerência de GTS;
  - o MLME-ORPHAN: gerenciamento de dispositivo órfão;
-

o MLME-RX-ENABLE: habilita/desabilita o sistema de rádio.

Em alguns protocolos os serviços MLME e MCPS podem ser manipulados pelo usuário para o desenvolvimento de projetos baseados na camada enlace e para o desenvolvimento das camadas superiores. O protocolo SMAC, objeto de estudo deste trabalho, permite essa manipulação de serviços e será abordado com mais ênfase na Seção 8.2.

6.2 Padrões IEEE 802.15.4 emergentes

O estudo de artigos técnicos e científicos relacionados a redes de sensores sem fio aponta que existe uma tendência clara de definição de normas internacionais tanto em relação a redes de sensores sem fio de aplicação industrial como para as de aplicação urbana.

Destaca-se o protocolo WirelessHart [3] como principal a ser considerado em aplicações industriais, tendo sido recentemente aprovado em março de 2010 pela *International Electrotechnical Commission* (IEC) como padrão internacional IEC 62591-Ed. 1.0. Tal protocolo foi concebido sobre o protocolo tradicional HART e o protocolo proprietário sem fio de acesso ao meio físico denominado TSMP (*Time Synchronized Mesh Protocol*) [86] e sobre o padrão de comunicação IEEE 802.15.4.

O próprio protocolo IEEE 802.15.4 está também em evolução, o *Working Group* IEEE 802.15.4 trabalha para a adição de mecanismos de sincronismo temporal (TDMA - *Time Division Multiple Access*) e saltos em frequência (FHSS - *Frequency-Hopping Spread Spectrum*) ao protocolo atual, em convergência com os mecanismos adotados no TSMP.

Para tal, foi criado o Grupo de Trabalho 4e para atender às demandas de aplicações industriais, norma técnica IEEE 802.15.4e [87], e o Grupo de Trabalho 4g para as redes de grande escala em cenários urbanos, norma técnica IEEE 802.15.4g [88], como por exemplo, as redes inteligentes de concessionárias (SUN - *Smart Utility Network*) [89].

Destacam-se ainda os protocolos Zigbee [4] e ISA 100 [90] como importantes no cenário atual. O ISA 100 é um protocolo ainda em fase de elaboração, mas que deverá manter compatibilidade com o WirelessHart, embora seu escopo de aplicação seja mais amplo. Já o Zigbee é um protocolo consagrado para interconexão de eletrônicos de

consumo e para automação residencial que teve sua primeira especificação ratificada em 2004.

Em relação a redes urbanas, nota-se o fortalecimento do protocolo IETF IPv6 sobre o IEEE 802.15.4 (RFC 4944). Este novo protocolo poderá, no futuro, substituir o Zigbee por adotar mecanismos mais sofisticados de comunicação, além de ser intrinsecamente integrado aos protocolos da Internet, pois está baseado em IP.

Dentro deste cenário, somado aos esforços para padronização internacional da tecnologia sem fio, existe ainda um campo de pesquisa muito amplo e oportuno para as entidades acadêmicas e científicas no desenvolvimento e otimização de sistemas baseados em redes de sensores sem fio.

É dentro desta vertente que este projeto de pesquisa se insere. É proposto o estudo de uma nova plataforma (SMAC - *Simple MAC*) para redes de sensores sem fio, também fundamentada no padrão IEEE 802.15.4. O SMAC, além de ser um protocolo simples, possui código fonte aberto, o que possibilita o estudo do protocolo e o desenvolvimento de rotinas complementares para adaptá-lo às necessidades da aplicação sem fio desejada.

Acredita-se que o trabalho aqui apresentado possa gerar conhecimento teórico e prático a respeito das redes de sensores sem fio, de forma a contribuir para a área estudada.

---

7 TRABALHOS CORRELATOS

O trabalho proposto por [91] apresenta uma abordagem baseada em rede sem fio ZigBee para controle e monitoramento de um sistema de iluminação pública baseado em lâmpadas de LED. É proposto um protocolo com mensagens específicas para a aplicação, bem como a utilização de um sistema operacional de tempo real (RTOS) para a implementação do nó da rede e gerenciamento das tarefas relacionadas ao sistema. Para a definição do protocolo, utilizou-se, como referência, o padrão DALI (*Digital Addressable Lighting Interface*), o qual foi desenvolvido especificadamente para aplicações com iluminação e oferece recursos simples para controle e monitoramento da lâmpada ligada ao nó. No entanto, os testes realizados envolveram apenas um LED, pois a lâmpada de LED ainda está em desenvolvimento.

Em [84] é desenvolvido um sistema de comunicação sem fio que possa monitorar variáveis analógicas e/ou digitais nos métodos de elevação artificial de petróleo e gás natural. O protocolo SMAC foi empregado para estabelecer a comunicação sem fio entre os sensores de uma rede com topologia estrela. Um dos pontos principais deste estudo foi a avaliação do enlace de comunicação com relação à quantidade de pacotes perdidos em diferentes ambientes (*indoor* e *outdoor*).

O trabalho [85] relatou um estudo investigativo do padrão IEEE 802.15.4 e da tecnologia ZigBee empregada em automação residencial. O objetivo era mostrar uma solução para a interligação de módulos dos subsistemas de automação residencial, com o uso de da tecnologia ZigBee através de uma rede *mesh multi-hop*.

A arquitetura do sistema de automação residencial apresentada é totalmente modular. Ou seja, é um conjunto de sistemas distribuídos representados pelos subsistemas autônomos e funcionais, que são processados nos nós da rede de sensores/atuadores. A rede sem fio desenvolvida era composta por um dispositivo coordenador, quatro dispositivos roteadores e nove dispositivos finais. O coordenador da rede atua como um *gateway* para prover acesso remoto ao controle da residência através da Internet, bem como transmitir informações da rede de sensores para ela. A tecnologia empregada supriu as necessidades das características de infra-estrutura, e de computação distribuída dos sistemas de automação residencial, com a utilização da rede em malha.

Em [92] são mostrados o projeto e a implementação de um sensor magnético sem fio para substituição da forma de comunicação convencional, com fios, atualmente existente nos campos de produção de petróleo com o método *Plunger Lift*¹³. A rede estabelecida possui topologia estrela e os nós sensores de comunicação sem fio basearam-se na plataforma SMAC.

Em suma, são encontrados na literatura diversos trabalhos relacionados às redes de sensores sem fio relativos aos padrões emergentes. Alguns propõem a utilização do protocolo industrial WirelessHart [93, 94, 95, 96], outros o emprego do protocolo ZigBee [97, 98, 99] e alguns o protocolo ISA100 [100,101].

Adicionalmente, novos artigos científicos estão apontando no meio acadêmico, relativos às revisões do protocolo IEEE 802.15.4, responsáveis por atender demandas específicas. O trabalho [102] apresenta um estudo do desempenho do protocolo IEEE 802.15.4e especialmente criado para aplicações industriais e o trabalho [89] aponta para um estudo do protocolo IEEE 802.15.4g dedicado às redes urbanas.

A maioria dos trabalhos encontrados pertinentes às redes sem fio, em conjunto com a plataforma SMAC, possui os nós sensores dispostos fisicamente em topologia estrela. Tal topologia permite que cada nó tenha uma única via de comunicação com o *gateway* e restringe a distância total que a rede pode alcançar.

No entanto, este projeto de pesquisa abrange um trabalho inédito na criação de uma camada de rede no protocolo SMAC. Esta camada permitirá a adequação do protocolo SMAC para aplicações em rede de sensores sem fio *mesh multi-hop*.

---

¹³ O *plunger lift* é um método de elevação artificial não contínuo, que tem como principal elemento um pistão presente na coluna de produção (*tubing*).

---

8 PROPOSTA DE TRABALHO

Este projeto de pesquisa propõe o desenvolvimento de um protocolo de rede para o roteamento de dados em redes de sensores sem fio *mesh multi-hop*, segundo a especificação IEEE 802.15.4. Tais redes podem ser utilizadas para a aplicação em *smart grid* em iluminação pública, objeto de estudo do projeto “Redes de Sensores Sem Fio de Aplicação Urbana”, ao qual o presente trabalho está vinculado.

Esta seção abordará os requisitos necessários para o desenvolvimento da proposta de trabalho.

8.1 Sistema de comunicação sem fio

Para o desenvolvimento do projeto foi escolhida a plataforma de fabricação da *Freescale*, baseada no componente MC1322X [103], que consiste de um processador de baixo consumo ARM7TDMI-S de 32 bits integrado a um *transceiver* RF IEEE 802.15.4 operante na faixa de frequência de 2,4GHz, vide Figura 23.

Figura 23 - Diagrama de bloco do MC1322X
Fonte: [103]

A Tabela 5 apresenta de forma sucinta os principais aspectos característicos do módulo MC1322X.

Tabela 5 – Características do MC1322X

ASPECTO	CARACTERÍSTICA MC1322X
Fonte de alimentação	2.0 até 3,6 V
Potência de saída do transmissor	-30dBm, podendo ser programado até 4dBm
Sensibilidade do receptor	-96dBm para uma <i>Packet Error Rate</i> em torno de 1%
Faixa Operacional	-40 °C à 105 °C
Extensos recursos de memória	128 KB de memória serial FLASH
	96 KB de RAM
	80 KB de ROM
Mínima potência de dissipação	22 mA para transmissão RX com rádio e MCU ativados
	29 mA para transmissão TX com rádio e MCU ativados
	3,3 mA com MCU ativado (rádio desligado)
	0,8 mA com MCU no modo <i>idle</i> (rádio desligado)
	0,4 uA com MCU no modo <i>off</i>
Diversidade de periféricos	UART (<i>Universal Asynchronous Receiver Transmitter</i>)
	SPI (<i>Serial Peripheral Interface</i>)
	KBI (<i>Keyboard Int</i>)
	SSI (<i>Synchronous Serial Interface</i>)
	ADC (<i>Analog-to-Digital Converters</i>)

Dentre as opções disponíveis para módulos de comunicação sem fio, o módulo MC1322X da *Freescale* demonstrou ser uma alternativa eficaz para o desenvolvimento do trabalho, quando comparado a outros módulos de comunicação sem fio de baixa potência. O módulo CC2430 [104] da *Texas Instruments*, bastante referenciado no meio acadêmico, é caracterizado por um alto desempenho, baseado na especificação ZigBee / IEEE 802.15.4 e integra um microprocessador de núcleo 8051 destinado a redes sem fio que requerem um consumo ultra-baixo de potência. Este módulo possui um consumo de corrente para transmissão RX e TX respectivamente de 27mA e 25mA, e para o modo *off* do micro controlador o consumo de corrente é de somente 0,9 uA.

O estudo comparativo entre esses dois módulos demonstra que a escolha pelo módulo MC1322X, operando em conjunto com o microprocessador ARM7, é viável, uma vez que seus parâmetros de consumo de corrente estão aceitáveis, e conduzem a uma mínima dissipação de potência, conforme desejado.

A solução *Freescale*, baseada na plataforma MC1322X, é composta por uma placa 1322X-SRB (*Sensor Reference Board*), uma placa 1322X-NCB (*Network Coordinator Board*), uma placa 1322X-LPB (*Low Power Board*), uma placa 1322X-USB, o software *Beekit* e acessórios para *debugging*¹⁴.

Esta plataforma permite a utilização da biblioteca de rotinas de interface SMAC para elaboração de projetos envolvendo redes de sensores sem fio. Tal protocolo é objeto de estudo deste trabalho e será portado para a plataforma de desenvolvimento dos nós de comunicação mencionados acima. Neste caso, o estudo detalhará os parâmetros e serviços do protocolo SMAC na seção seguinte.

8.2 Padrão SMAC - *Simple MAC*

O protocolo SMAC [105] é fundamentado na linguagem de programação C e pode ser empregado no desenvolvimento de aplicações proprietárias de transmissão e recepção em RF, utilizando os circuitos integrados dos módulos MC1322X apresentados na Seção 8.1.

Tal protocolo tem a camada física baseada no padrão IEEE 802.15.4 e possui uma implementação simplificada da camada MAC. O diagrama de blocos do protocolo SMAC - MC1322X é apresentado na Figura 24.

Figura 24 - Diagrama de Blocos do SMAC - MC1322X

Fonte: [105]

O SMAC possui dois módulos opcionais relativos aos módulos de Segurança e OTAP (*Over the Air Programmer*) [105]. O módulo OTAP permite que os usuários façam *update* das placas remotamente, não havendo necessidade de uma conexão física.

¹⁴ Do inglês *debugging*, se refere à etapa de depuração de erros.

Já o módulo de Segurança é um componente de *software* que permite a codificação e decodificação das mensagens através de seu API (*Application Programming Interface*). O processo de codificação é executado pelo módulo ASM (*Advanced Security Module*), que implementa o algoritmo de segurança AES¹⁵.

O formato geral da estrutura de pacote SMAC é apresentado na Figura 25 e consiste nos seguintes componentes básicos:

- Sequência de Preâmbulo, um nó inicia uma transmissão enviando 32 bits (4 bytes) de sequência de preâmbulo;
- Delimitador SOF (*Star of Frame*), que contém 1 byte para delimitação de início do quadro;
- Tamanho do quadro SMAC, composto por 1 byte;
- Byte de Código, para identificação dos quadros SMAC;
- Campo manipulável pelo usuário formado por 123 bytes;
- FCS (*Frame Check Sequence*) é composto por 2 bytes.

Figura 25 - Estrutura de pacote SMAC

São oferecidos pelo protocolo SMAC comandos para criar redes ponto-a-ponto e estrela. No entanto, tal protocolo permite a adaptação para topologias de redes mais complexas, como a topologia *mesh*, mediante a manipulação de sua biblioteca de rotinas interface.

A capacidade de manipulação das primitivas de serviços permite a implementação de projetos baseados na camada MAC e no desenvolvimento das camadas superiores. O protocolo SMAC permite essa manipulação e oferece comandos para que os usuários criem sua própria camada de rede. Desta forma, a camada de rede a ser desenvolvida chama as primitivas de serviço da camada MAC para habilitar uma aplicação específica do usuário.

¹⁵ AES (*Advanced Encryption Standard*) é um padrão de criptografia avançada de 128 bits para a transmissão/recepção de dados com a função de auxiliar na segurança dos pacotes enviados.

No SMAC, há vinte e uma primitivas das quais uma é relacionada ao serviço MCPS, dezoito ao serviço MLME e uma primitiva denominada DRV para configuração de relógio em tempo real. As primitivas do protocolo SMAC são apresentadas e descritas na Tabela 6.

Tabela 6 – Primitivas de serviços SMAC

PRIMITIVAS	DESCRIÇÃO
DRVConfigureRTC	Configuração do relógio de tempo real
MCPSDataRequest	Transmissão de pacotes
MLMEDozeRequest	Comuta o <i>transceiver</i> para o modo <i>Doze</i>
MLMEEnergyDetect	Mede o nível de energia do canal
MLMEGetChannelRequest	Retorna o canal RF atual
MLMEGetPromiscuousMode	Retorna o estado atual do modo de comunicação promísua
MLMEGetRficVersion	Retorna o número da versão do MC1322X
MLMEHibernateRequest	Comuta para o modo hibernar
MLMELinkQuality	Indica o valor da potência (em dBm)
MLMEPAOutputAdjust	Ajusta a potência de saída do transmissor
MLMEPHYResetIndication	Primitiva chamada quando um <i>soft-reset</i> ocorre para resetar o rádio
MLMEPHYSoftReset	Primitiva chamada quando um <i>soft-reset</i> ocorre para resetar MC1322X
MLMEPHYXtalAdjust	Ajusta o valor do gatilho do <i>clock</i> de referência do <i>transceiver</i>
MLMERadioInit	Inicializa os parâmetros do rádio
MLMERXDisableRequest	Comuta o MC1322X do modo de recepção para o modo livre (<i>idle</i>)
MLMERXEnableRequest	Põe o <i>transceiver</i> em modo de recepção
MLMEScanRequest	Examina cada um dos canais
MLMESetChannelRequest	Seleciona o canal e a frequência utilizada
MLMESetPromiscuousMode	Permite habilitar ou limpar o modo de comunicação promísua
MLMESetWakeupSource	Configura o dispositivo para capacidade de <i>wake up</i> automático
MLMETestMode	Testa o <i>transceiver</i> MC1322X

A partir da manipulação das funções de primitivas de serviços citadas acima é possível inserir um algoritmo de roteamento geográfico no protocolo SMAC, conforme sugerido neste trabalho.

8.2.1 Tipos de Mensagens e Funções SMAC

São definidos quatro tipos de mensagens para o SMAC: *Reception* (RX), *Transmission* (TX), *Energy Detect* (ED) e *Time out* (TO).

Essas mensagens são processadas pelo módulo de Gerenciamento de Rádio presente na Figura 24. Este módulo possui uma fila circular das mensagens que precisam ser processadas como uma estrutura de FIFO (*First In – First Out*). Ou seja, as mensagens são atendidas na mesma ordem como elas chegam [105].

As mensagens são incluídas na fila por meio de uma das três funções baseadas em três importantes primitivas de serviços presentes na Tabela 6:

- MCPSTDataRequest (para mensagens TX);
- MLMERXEnableRequest (para mensagens RX);
- MLMEEnergyDetect (para mensagens ED).

Assim, por exemplo, a função MCPSTDataRequest() prepara o tipo de mensagem a ser enviada e chama a função handle_new_message responsável por alocar uma nova mensagem na fila.

Cada tipo de mensagem RX, TX, ED e TO têm o seu próprio ciclo de vida. Este ciclo é definido pelos estados diferentes que cada uma dessas mensagens pode ter.

O diagrama com o modelo geral da máquina de estados das mensagens é ilustrado na Figura 26. Esta máquina de estados especifica a ordenação admissível dos estados das mensagens durante seu ciclo de vida.

xx – Message type

Figura 26 - Modelo geral da máquina de estado das mensagens

Fonte: [105]

Primeiramente as mensagens passadas para o Módulo de Gerenciamento de Rádio estão em um estado inicial de "requested". Em seguida, elas são direcionadas para o módulo MACA (*MAC Accelerator*), presente no módulo MC1322X apresentado na Figura 23. O Módulo de Gerenciamento de Rádio processa as mensagens, de acordo com as interrupções geradas pelo módulo MACA [106].

Como parte do protocolo IEE 802.15.4, os pacotes devem ser gerados e transmitidos, recebidos e verificados e a energia do canal deve ser medida através de uma avaliação de canal livre (CCA). Além disso, as combinações ou seqüências de eventos são necessárias, como uma resposta ACK após um pacote recebido.

O módulo MACA facilita essas atividades através do controle do *transceiver*, diminuindo a sobrecarga de funções da CPU. A função dedicada DMA (*Direct Memory Access*) move os dados entre os *buffers* MACA e a memória RAM, sem a necessidade de intervenção da CPU [106].

Dentre suas funções, o módulo MACA realiza a transmissão de pacotes, analisa os pacotes recebidos, implementa rotinas para acesso ao meio, controle de *link* e lida com os registros de controle/*status* mapeados na memória da CPU, independente do processador ARM.

Como as mensagens são processadas por seus estados, a função `process_radio_msg` deve ser chamada periodicamente para executar a máquina de estado das mensagens [105].

A função `process_radio_msg` verifica se há alguma mensagem presente na fila (RX, TX, ED ou TO) e, então, dependendo do tipo de mensagem, chama a respectiva função para processar a mensagem existente (`process_tx_msg`, `process_rx_msg`, `process_ed_msg` e `process_to_msg`).

O estado de máquina, presente em cada uma dessas funções de processamento, decide qual estado da mensagem é válido e, dependendo do estado, executa uma determinada função. Alguns estados gravam dados diretamente nos registradores MACA outros apenas verificam o campo *gRadioEvtFlags*, atualizado durante as interrupções MACA.

Basicamente, o código executa um ciclo permanente (`process_radio_msg` -> `process_xx_msg`) e é interrompido pelo módulo MACA, onde a *flag gRadioEvtFlags* é atualizada. Na função `process_xx_msg` esta *flag* é verificada e, se o ciclo da mensagem for concluído, a mensagem é removida da fila.

8.2.2 Restrições das Mensagens

Ao desenvolver uma aplicação no SMAC, devem ser consideradas as seguintes restrições das mensagens:

- Os dados da mensagem não devem ser alterados durante o período em que a mensagem estiver sendo colocada na fila, até que ela esteja em um estado final. O estado final é qualquer um dos seguintes estados: ação concluída, abortada ou quando um *time out* ocorrer.

- Uma vez que uma mensagem vai para um estado final, ela é removida da fila. Se a mensagem não vai para um estado final, a próxima mensagem na fila não é processada.

- O Módulo de Gerenciamento de Rádio pode ter um número variável de mensagens na fila. O tamanho da fila de mensagem é definido pela `MAX_NUM_MSG`, declarada pelo usuário no código fonte do SMAC.

- A função `process_radio_msg` deve ser chamada para processar essas mensagens. Se esta função não é apropriadamente chamada, o processamento das mensagens não funcionará conforme o esperado. A aplicação deve chamar a função `process_radio_msg` periodicamente pelo menos até a fila de mensagem ficar vazia.

8.2.3 Estrutura de Pacote `message_t`

O protocolo SMAC possui uma estrutura de pacote que pode ser manipulada pelas camadas superiores chamadas de `message_t`, a qual define a estrutura de armazenamento das informações da mensagem. Essa estrutura é composta pelos membros:

- `u8Status` - *Status* da mensagem. É formado por três bits para o tipo mensagem e cinco bits para o estado da mensagem.
  - `*pu8Buffer` - Estrutura que contém os dados a transmitir ou receber.
  - `u8BufSize` - Dependendo do tipo de mensagem, esse campo pode significar: tamanho do pacote transmitido ou número máximo de bytes a receber. No fim da recepção, este campo armazena o número de bytes recebidos e o canal onde o ED deve ser acessado.
-

- `cbDataIndication` - Indica uma função *callback*, ou função de retorno, que é chamada para notificar o progresso da comunicação.

```
typedef struct message_tag {  
    msg_status_t u8Status;  
    smac_pdu_t *pu8Buffer;  
    uint8_t u8BufSize;  
    callback_t cbDataIndication;  
}message_t;
```

Figura 27 - Estrutura de pacote `message_t`

8.3 Protocolos de Roteamento

Para aplicações específicas, como em redes em grande escala, em razão da necessidade de economia do consumo energético e de largura de banda, a implementação de algoritmos de roteamento geográfico na camada de rede é recomendada ao invés de algoritmos de roteamento sob demanda, como é o caso do padrão ZigBee.

Nos protocolos de roteamento por demanda baseados em “*flooding*”, tal como o AODV (*Ad hoc On Demand Distance Vector*) adotado no Zigbee, o processo de descoberta de rotas consiste em inundar a rede com a mensagem *broadcast* até que um nó, que conheça a rota ao destino da mensagem, ou o próprio nó de destino, seja encontrado. Tal procedimento possui custos altos em ocupação de banda e não se enquadram nos requisitos energéticos exigidos pelas redes de sensores sem fio em larga escala, caso em que se encontram as aplicações em áreas urbanas.

Já os protocolos de roteamento geográfico utilizam informações de localização para calcular uma rota eficiente para um determinado destino. Ainda, algoritmos de roteamento geográficos requerem baixo poder computacional e baixa sobrecarga de comunicação [107].

Nós utilizando estes algoritmos de roteamento não necessitam de uma grande quantidade de mensagens de controle para acompanhar o estado atual da topologia de rede, necessitando somente da informação da topologia a um salto de distância. A abordagem localizada utilizada para a tomada de decisão elimina a necessidade de manter tabelas de roteamento. Conseqüentemente, a sobrecarga de controle da rede é consideravelmente reduzida, favorecendo seu uso em redes de grande escala [108, 109].

Diante do exposto, tendo em vista que este trabalho irá contribuir com outro projeto envolvendo redes de sensores sem fio para controle e monitoramento da iluminação pública em grandes áreas, foram escolhidos, para utilização na camada de rede, algoritmos de roteamento geográfico. Além disso, tal algoritmo é beneficiado pelo fato de que as coordenadas geográficas, de cada ponto de iluminação pública, são conhecidas pelas concessionárias.

Esses algoritmos utilizam informações geográficas obtidas através de um sistema de GPS (*Global Positioning System*) ou através de um sistema local, válido somente para os sensores da rede. Assim, os algoritmos geográficos utilizam informações de posicionamento dos nós no roteamento dos dados.

Dentre os algoritmos que melhor se adaptam ao caso de roteamento baseado em posicionamento geográfico, destacam-se aqueles denominados de *Greedy Packet Forwarding* [64], que tratam do repasse criterioso das mensagens. Ao contrário de técnicas como *flooding*, esses tipos de algoritmos não se preocupam em definir uma rota antes de repassar as mensagens.

O protocolo de *roteamento Greedy* escolhido será baseado na distância Euclidiana ou distância Métrica, na qual a distância entre dois pontos pode ser calculada pela aplicação do teorema de Pitágoras.

Basicamente, o algoritmo funciona da seguinte maneira: na Figura 28, suponha que o nó S com raio de alcance R seja um nó intermediário que recebe uma mensagem endereçada ao nó D. A distância Euclidiana bidimensional será calculada para todos os nós da rede em relação ao nó de destino D. Ou seja:

$$d_{DP_n} = \sqrt{(x_D - x_{P_n})^2 + (y_D - y_{P_n})^2} \quad (3)$$

Figura 28 - Algoritmo *Greedy* baseado na distância Euclidiana

Desta forma, o nó S irá repassar o pacote para o nó P_n que estiver dentro do seu alcance R e que apresentar menor distância Euclidiana em relação ao nó de destino D. O

algoritmo é repetido até que a mensagem chegue ao nó de destino. A idéia é que, quando um nó intermediário receber uma mensagem, este deve repassá-la ao nó adjacente cujo algoritmo julgar estar mais próximo do destinatário final. Isto se deve ao fato de priorizar o menor número de saltos na rede, contabilizando um atraso mínimo na comunicação fim a fim.

Caso o nó intermediário S, responsável pelo repasse da mensagem, for o mais próximo do nó de destino D dentre todos os seus vizinhos, mas o nó D não pertencer a sua área de alcance de rádio, o *Critério Drop* será estabelecido e a mensagem será descartada para evitar que o algoritmo entre em *loop*. Tal situação é apresentada na Figura 29.

Figura 29 - Critério Drop

8.4 Protocolo Proposto

Conforme visto na Seção 6.1.2.1, o padrão IEEE 802.15.4 suporta topologias de rede em estrela, árvore e *mesh*. Entretanto, tal padrão não define como essas topologias podem ser obtidas e mantidas. Assim, a formação da rede e o roteamento por múltiplos saltos devem ser suportados pelas camadas superiores do protocolo de rede.

Nesta seção são abordadas as principais medidas adotadas com relação ao desenvolvimento do *software* baseado na implementação do protocolo de roteamento geográfico no SMAC, destinado à rede *mesh multi-hop*.

8.4.1 Quadro de mensagens

De acordo com a Figura 25, o protocolo SMAC disponibiliza 123 *bytes* para serem manipulados pelo usuário. Desta forma, para o protocolo aqui proposto, foram desenvolvidos três quadros de mensagens: um quadro *Hello*, um quadro de dados e um quadro de *acknowledgment* (quadro de confirmação).

A técnica utilizada, neste protocolo, para a descoberta de vizinhos é por meio do envio periódico de quadros *Hello*. Cada nó difunde, de tempos em tempos, quadros *Hello* para anunciar ou reiterar a sua presença aos nós vizinhos, mantendo assim sua conectividade com os vizinhos. Tal quadro permite disseminar informações locais a respeito da posição geográfica de cada nó na rede.

Caso nós vizinhos observem que o envio do quadro *Hello* de um determinado nó foi cessado, assume-se que ocorreu uma quebra de enlace e o nó está impossibilitado de se comunicar.

O quadro *Hello* possui 13 *bytes*. Este quadro contém não só a informação das coordenadas geográficas *x* e *y* do nó emissor do quadro *Hello*, mas também uma identificação do quadro *Hello* e o MAC ID deste nó emissor.

Ident. do quadro	Identificação do Nó	Coordenadas Geográficas	
Ident. Hello (1 byte)	MAC ID (4 bytes)	Posição X (4 bytes)	Posição Y (4 bytes)

Figura 30 - Quadro *Hello*

A estrutura do quadro de dados é composta por sete campos de informações totalizando 123 *bytes*, sendo 86 *bytes* restantes destinados para *payload*. O primeiro campo corresponde ao identificador do quadro de dados, o segundo campo contém as informações de endereçamento do fluxo dos dados, ou seja, o MAC ID do nó remetente de origem do fluxo e o MAC ID do nó destinatário final.

Pertencem ao terceiro campo informações de roteamento a cada salto da mensagem. Corresponde aos endereços: MAC ID do nó remetente intermediário e ao respectivo MAC ID do nó destinatário intermediário. Tais endereços modificam a cada salto. Quando a mensagem está a um salto de alcançar o nó destinatário final, o endereço do nó destinatário dos campos de endereçamento e roteamento são iguais.

São pertinentes ao quarto e quinto campos, respectivamente, as coordenadas geográficas *x* e *y* do nó emissor fonte e do nó destinatário final do fluxo na rede. Cada nó utiliza as informações deste último campo para rotear os dados pela rede. Tal roteamento de dados é realizado segundo o protocolo de roteamento geográfico baseado na distância Euclidiana (equação 3), apresentado na Seção 8.3.

O número do pacote para o controle do fluxo de dados é armazenado no campo seguinte, sendo o último campo responsável por conter os dados a serem transmitidos pela rede (*payload*).

Identificação do quadro	Identificação do Nó (Endereçamento)		Identificação do Nó (Roteamento)		Coordenadas Geográficas (Nó emissor fonte)		Coordenadas Geográficas (Nó destinatário final)		Identificação do Pacote	Payload (86 bytes)
Ident. Dados (1 byte)	MAC ID Nó Remetente (4 bytes)	MAC ID Nó Destinatário (4 bytes)	MAC ID Nó Remetente (4 bytes)	MAC ID Nó Destinatário (4 bytes)	Posição Xf (4 bytes)	Posição Yf (4 bytes)	Posição Xd (4 bytes)	Posição Yd (4 bytes)	Número do Pacote (4 bytes)	

Figura 31 - Quadro de dados

O quadro de *acknowledgment* é formado por 14 *bytes*. Existe um campo de identificação do quadro de *acknowledgment*, seguido por um campo representado pelo MAC ID do nó emissor anterior, responsável pelo envio do quadro de dados. Esta última informação é importante, uma vez que o nó emissor do quadro de dados também é o receptor do quadro de *acknowledgment*. Baseado nesta informação, um nó pode decidir se o quadro de *acknowledgment* é ou não destinado a ele.

O terceiro e quarto campos referem-se, respectivamente, ao MAC ID do nó emissor fonte do fluxo na rede e ao número do pacote de dados recebido. Tais informações permitem identificar os quadros de *acknowledgment* na rede. O último campo é responsável por armazenar o índice de qualidade do *link* (LQI¹⁶) dos dados recebidos.

Ident. do quadro	Ident. do Nó (Emissor Anterior)	Ident. do Nó (Emissor Fonte do Fluxo)	Identificação do Pacote Recebido	Qualidade do Link
Ident. ACK (1 byte)	MAC ID (4 bytes)	MAC ID (4 bytes)	Número do Pacote (4 bytes)	LQI (1 byte)

Figura 32 - Quadro de *acknowledgment*

Os quadros de *acknowledgment* são quadros de confirmação de recebimento enviados pelos nós da rede ao seu nó vizinho precedente, responsável pelo repasse do quadro de dados. Caso essa confirmação não seja recebida pelo nó vizinho, ele reenviará o quadro de dados ao nó adjacente, escolhido pelo algoritmo de roteamento geográfico, até que um quadro de *acknowledgment* seja recebido ou a mensagem seja descartada.

¹⁶ A medida LQI (*Link Quality Indication*) é uma caracterização da intensidade e/ou qualidade da mensagem recebida.

Em redes sem fio, a confirmação de recebimento de dados, a cada retransmissão de dado, é bem mais eficiente que a confirmação fim a fim. Isto porque a probabilidade de falha na transmissão entre dois nós é alta, por exemplo, devido à ocorrência de interferências.

O código a seguir cria uma estrutura para armazenar dados do tipo mensagem. Os membros desta estrutura definem os três tipos de quadros de mensagem abordados nesta seção: dados, *Hello* e *acknowledgment*.

```
/* *****  
 * DECLARAÇÃO DOS QUADROS DE MENSAGEM  
***** */  
  
struct mensagem_s {  
    enum { ACK = 'A', HELLO = 'H', MSGDADOS = 'D'  
    } tipo;  
    union {  
        struct {  
            uint32_t NoEmissorAnterior;  
            uint32_t OrigemFluxo;  
            uint32_t PacketNumber;  
            uint8_t Quality;  
        } ack;  
        struct {  
            uint32_t NoAtual;  
            int32_t posX;  
            int32_t posY;  
        } hello;  
        struct {  
            struct {  
                uint32_t remetente, destinatario;  
            } enderecamento;  
            struct {  
                uint32_t remetente, destinatario;  
            } roteamento;  
            int32_t posXf;  
            int32_t posYf;  
            int32_t posXd;  
            int32_t posYd;  
            uint32_t packetNumber;  
            uint32_t payload;  
        } msgdados;  
    } dados;  
};  
...
```

Figura 33 - Declaração dos quadros de mensagens

Para armazenamento e organização de todas as mensagens a serem enviadas pela rede é proposta a formação de uma tabela de mensagens. Tal tabela organiza as mensagens de *Hello*, dados e *acknowledgment* em uma fila, para que as mesmas sejam enviadas ordenadamente.

Essas filas são baseadas no princípio FIFO (*first in, first out*), em que os elementos que foram inseridos no início serão os primeiros a serem removidos. As mensagens são adicionadas e removidas da tabela segundo as funções básicas *enqueue*, que adiciona um elemento ao final da fila, e *dequeue*, que remove o elemento do início da fila. O tamanho máximo para esta fila de mensagens é determinado pelo parâmetro *tbMsgNmax*, estipulado em 50 no presente trabalho.

O princípio de funcionamento da tabela de mensagens pode ser visto na Figura 34. Inicialmente, são necessários três valores para operar com a tabela de mensagem, um tamanho fixo, definido por T ($T = tbMsgNmax$), um índice inicial para a primeira mensagem válida da tabela, I , e um índice para a posição onde a próxima mensagem será alocada, definido por F .

O índice F deve ser incrementado a cada mensagem adicionada na fila. Conforme essas mensagens vão sendo removidas da fila, ou seja, enviadas, é estabelecido um critério para incrementar o índice I . Após o envio das mensagens de *Hello* e *acknowledgment*, o índice I deve ser incrementado imediatamente. No entanto, após o envio de uma mensagem de dados, é necessário aguardar o recebimento do quadro de confirmação para incrementar o índice I . Somente após o recebimento deste quadro a mensagem será removida da fila. Adicionalmente, caso o critério *drop* seja estabelecido, a mensagem de dados será descartada e o índice I também será incrementado.

Os índices I e F vão ser sempre menores do que T , tendo em vista que os índices válidos de um vetor de tamanho estão no intervalo entre 0 e $T-1$. Caso I ou F cheguem a T , depois de serem incrementados, devem voltar a ser zero.

Se a tabela de mensagens estiver vazia, ou seja, todas as N mensagens adicionadas à tabela já foram enviadas e removidas da fila, I será igual a F . Tal situação implica em um cuidado: a fila de mensagens não pode ser preenchida até que F atinja I ; isto é, condição em que F alcance T , volte a zero e cresça até atingir I . Isso porque não haveria como distinguir uma fila cheia de uma fila vazia.

Para dificultar a ocorrência deste caso, a tabela de mensagens deve ser acessada em intervalos de tempo muito pequenos (milissegundos), determinados pelo parâmetro

tpEnvioMsg, definido em 100ms neste protocolo. Isso permite verificar a existência de mensagens pendentes e enviá-las o quanto antes, reduzindo o atraso de fila.

O atraso de fila pode ser definido como o tempo que uma mensagem espera na fila para ser transmitida. Este atraso é influenciado pela quantidade de outras mensagens que chegaram antes na fila e que também estão esperando para serem transmitidas.

A carga de fluxos que passa pelo nó é fator determinante no atraso de fila. Com um tráfego muito intenso, o atraso de fila é maior. Por outro lado, se a fila estiver vazia, o tempo de atraso de fila praticamente não existe.

No entanto, ao acelerar o envio das mensagens presentes na tabela, o tempo de atraso de fila é reduzido e as mensagens têm um atraso menor na transmissão.

Para eliminar a existência de mensagens duplicadas na tabela, antes de adicionar uma nova mensagem a fila, é feita uma comparação da mesma com as demais mensagens já armazenadas na tabela. Caso a mensagem a ser adicionada já exista, ela será ignorada.

Figura 34 - Tabela de mensagens

O código abaixo apresenta a função responsável pela alocação dos quadros de mensagem *Hello*, dados e *acknowledgment* na tabela de mensagens.

```

/*****
* ADICIONA MENSAGEM Á TABELA DE MENSAGEM
*****/
struct mensagem_s TabelaMensagem[TBMSGNMAX];
int inicio = 0, fim = 0;
...

void preencher_tabela(struct mensagem_s msg)
{
    for (int i = inicio; i != fim; i = (i + 1) % TBMSGNMAX) {
        if (!memcmp(&TabelaMensagem[i], &msg, sizeof(msg))) {
            return;
        }
    }
    if (!((fim + 1) % TBMSGNMAX == inicio)) {
        memset(&TabelaMensagem[fim], 0, sizeof(msg));
        TabelaMensagem[fim++] = msg;
        fim %= TBMSGNMAX;
    }
    else {
        return;
    }
}

```

Figura 35 - Formação da tabela de mensagens

8.4.2 Formação da tabela de vizinhos

Para que um nó tenha conhecimento dos nós que estão dentro da sua área de alcance de rádio, é proposta a formação de uma tabela de vizinhos.

A principal função desta tabela é manter o nó informado sobre a existência de nós vizinhos em potencial para o repasse da mensagem, caso seja necessário realizar o roteamento da mensagem.

O protocolo *Greedy* necessita da posição física/geográfica dos nós da rede para que o endereçamento de cada um seja realizado. Uma vez definidos os endereços, cada nó realiza uma busca para conhecer seus nós vizinhos, armazenando seus respectivos endereços na tabela de vizinhos. Essa busca é realizada mediante o envio periódico do quadro *Hello* apresentado na Seção 8.4.1. Tal quadro é responsável pelo transporte de informações relevantes ao endereçamento de cada nó na rede.

O quadro *Hello* é enviado na forma de *broadcast* por todos os nós da rede, a cada intervalo de tempo pré-determinado pela constante *tpMsgHello*. Desta forma, a tabela de vizinhos é dinâmica e formada na medida em que um nó recebe um quadro

Hello de seus vizinhos. O número máximo de elementos que esta tabela pode conter é definido pelo parâmetro $tbVizNmax$.

A idéia básica para formação da tabela de vizinhos é bastante simples. Ao receber um quadro *Hello*, cada nó da rede registra em sua tabela de vizinhos o MAC ID do nó emissor do quadro e suas coordenadas geográficas x e y . Adicionalmente, para cada vizinho acrescentado na tabela, é dado um *status* (*TRUE*) para validar a posição do nó na tabela e uma constante denominada de δH .

Tal constante permite o controle da permanência de cada nó na tabela. Se um determinado nó ficar um período δH sem enviar o quadro *Hello*, seja por falta de bateria ou devido a um obstáculo físico, ele será automaticamente eliminado da tabela de seus vizinhos (*status* igual a *FALSE*). Essa medida evita incluir na tabela um nó falho e que poderá comprometer o repasse do pacote, caso ele seja escolhido.

A constante δH é iniciada em δH_{max} e decrementada a cada intervalo de tempo $tpMsgHello$. No protocolo presente, δH é iniciada em 3 e decrementada a cada intervalo de 20 segundos. O valor desta constante é restaurado para o valor inicial periodicamente, mediante o recebimento do quadro *Hello*. Ou seja, cada nó, ao reiterar sua presença aos nós vizinhos por meio do envio do quadro *Hello*, tem o parâmetro δH reiniciado para δH_{max} . Caso um nó fique 80 segundos sem enviar o quadro *Hello*, δH irá reduzir a zero e ele será eliminado da tabela de seus vizinhos.

Os parâmetros responsáveis pela dinâmica da tabela de vizinhos, $tbVizNmax$, $tpMsgHello$, δH e δH_{max} , são flexíveis e podem ser alterados para atender aplicações específicas. A Figura 36 apresenta o modelo da tabela de vizinhos. A primeira coluna da tabela é dedicada às informações do próprio nó local, e as demais colunas seguintes possuem as informações pertinentes aos nós vizinhos.

Figura 36 - Tabela de vizinhos

Para a formação da tabela de vizinhos mostrada anteriormente, foram implementadas as funções declaradas no código apresentado a seguir. Quando um determinado nó recebe um quadro *Hello*, as funções `posicao_valida()` e `AdicionaMsg_TabViz()` são chamadas para atualizar a tabela de vizinhos do nó.

```

/*****
 * FORMAÇÃO DA TABELA DE VIZINHOS
 *****/

struct elementos_s {
    bool_t valido;
    int Ident;
    int x;
    int y;
    int deltaH;
};

...
struct elementos_s tabelaViz[TBVIZNMAX];
int posicao_valida(int Ident)
{
    for (int i = 1; i < TBVIZNMAX; i++)
        if (tabelaViz[i].Ident == Ident && tabelaViz[i].valido == TRUE)
            return i;
    for (int i = 1; i < TBVIZNMAX; i++)
        if (tabelaViz[i].valido != TRUE)
            return i;
}

...
void AdicionaMsg_TabViz(int i, int Ident, uint32_t x, uint32_t y)
{
    tabelaViz[i].valido = TRUE;
    tabelaViz[i].Ident = Ident;
    tabelaViz[i].x = x;
    tabelaViz[i].y = y;
    tabelaViz[i].deltaH = DELTAHMAX;
}

```

Figura 37 - Formação da Tabela de Vizinhos

8.4.3 Critério de Repasse do Pacote

A fim de possibilitar a prática de roteamento de dados ao SMAC, seu código fonte deve ser modificado. Para tal é utilizada a biblioteca de rotinas de interface SMAC para introduzir uma camada de rede neste protocolo. O protocolo de rede adotado para este propósito é o protocolo de roteamento geográfico *Greedy* baseado na distância

Euclidiana, apresentado na Seção 8.3. Isto permitirá ao SMAC um caráter criterioso de repasse do pacote de dados, por meio de múltiplos saltos, e sua aplicação em redes sem fio com topologia *mesh*, antes limitado a estrela.

O pacote a ser roteado é estruturado de acordo com o quadro de dados da Seção 8.4.1, cujo *payload* é variável até 86 bytes. O critério estabelecido para o repasse de tal pacote é bastante simples: antes de transmitir uma mensagem, o nó responsável pelo repasse da mensagem deve analisar o próximo salto da mensagem, definindo, assim, qual nó da rede irá receber o pacote. A preferência é dada para o nó que estiver mais próximo do nó destinatário, priorizando assim, o menor número de saltos entre fonte e destino.

Para fornecer uma maior confiabilidade na transmissão de dados, o método adotado para o controle de tráfego neste trabalho parte do princípio da confirmação de recebimento de quadros. Basicamente, o nó destinatário de um pacote transmite uma mensagem de reconhecimento (quadro ACK) para cada pacote recebido. Esta mensagem informa à fonte o número sequencial do pacote recebido. O remetente aguarda o recebimento deste quadro de *acknowledgment* para transmitir o próximo quadro (*Hello*, dados ou *acknowledgment*) presente na fila de mensagens.

Se, após um período pré-definido por um temporizador, o quadro ACK para um determinado quadro de dados não tiver sido recebido, o protocolo assume que o pacote foi perdido e o retransmite. Essa técnica de reenvio automático de quadros de dados é definida por um número de tentativas de envio da mensagem determinada por NT . Este parâmetro é limitado entre zero e $maxNT$. Cada tentativa de reenvio da mensagem acontece após um intervalo de tempo determinado pelo temporizador. O valor deste temporizador é dado por:

$$TR = 4^{NT} \quad (4)$$

O número de tentativas é consultado a cada reenvio da mensagem. Enquanto o número de tentativas for diferente do valor máximo $maxNT$, o quadro de dados será enviado para o mesmo nó vizinho. Caso NT atinja o limite máximo, o nó remetente desiste de entregar o pacote para este nó e o exclui da sua tabela de vizinhos. O ciclo é, então, reiniciado e um segundo nó vizinho será escolhido para o repasse do pacote, baseado na distância Euclidiana. Caso contrário, o número de tentativas é incrementado e a tentativa de reenvio será executada novamente após o intervalo de tempo definido (TR).

O fluxograma abaixo apresenta essa técnica adotada para o repasse do pacote de dados segundo as diretrizes do protocolo de roteamento geográfico *Greedy*.

Figura 38 - Fluxograma lógico para repasse do pacote de dados

O critério adotado para escolha do próximo vizinho, apresentado no fluxograma acima, é executado pela função `proxSalto_rotGeog()`. Esta função está descrita no código abaixo

```

/*****
* FUNÇÃO: ROTEAMENTO GEOGRÁFICO
*****/
...
uint32_t proxSalto_rotGeog (int32_t posXd, int32_t posYd)
{
    int32_t temp1 = (tabelaViz[0].x - posXd) * (tabelaViz[0].x - posXd);
    int32_t temp2 = (tabelaViz[0].y - posYd) * (tabelaViz[0].y - posYd);
    float distancia = sqrt(temp1 + temp2);
    uint32_t macIdViz = ID_ME;
    for (int i = 1; i < TAMANHO_TABELA; i++) {
        if (tabelaViz[i].valido == TRUE) {
            temp1 = (tabelaViz[i].x - posXd) * (tabelaViz[i].x - posXd);
            temp2 = (tabelaViz[i].y - posYd) * (tabelaViz[i].y - posYd);
            float nova_distancia = sqrt(temp1 + temp2);
            if (nova_distancia < distancia) {
                macIdViz = tabelaViz[i].Ident;
                distancia = nova_distancia;
            }
        }
    }
    return macIdViz;
}
...

```

Figura 39 - Função de roteamento da mensagem

Uma vez que o pacote é entregue com sucesso, o quadro de dados é retirado da tabela de mensagens do nó remetente e o ciclo se repete para um próximo quadro de dados.

Adicionalmente, caso o critério *drop* seja estabelecido, o pacote será descartado e o ciclo é dado como terminado.

O número de tentativas NT estabelecido para este trabalho é igual a três. Desta forma, o intervalo de tempo a ser esperado a cada reenvio da mensagem é apresentado na Tabela 7.

Tabela 7 – Temporizador

Número de Tentativas	Temporizador (tempo até zerar)	AÇÃO
0	0	Tentativa inicial de envio para nó vizinho. - Recebeu ACK: Prossegue para Fim ou Reinicia ciclo. - Não recebeu ACK: Incrementa NT.
1	500ms	Reenvia mensagem para o mesmo nó vizinho. - Recebeu ACK: Prossegue para Fim ou Reinicia ciclo. - Não recebeu ACK: Incrementa NT.
2	1700ms	Reenvia mensagem para o mesmo nó vizinho. - Recebeu ACK: Prossegue para Fim ou Reinicia ciclo. - Não recebeu ACK: Incrementa NT.
3	6500ms	Remove nó vizinho da tabela de vizinhos. - Reinicia o ciclo. - Reenvia a mensagem por outro nó vizinho.

Os quadros de dados são considerados entregues somente quando o transmissor obtém uma confirmação do receptor, isto é desde o instante em que o quadro começa a ser transmitido até quando o reconhecimento chega ao transmissor.

O tempo total necessário para o envio de um pacote e recebimento de uma confirmação é dado pela soma das seguintes parcelas: tempo de transmissão do pacote (A_t); atraso de propagação (A_{prop}); tempo de processamento no receptor (A_{proc}); tempo de transmissão do ACK (A_{tAck}); atraso de propagação (A_{prop}); tempo de processamento no receptor (A_{proc}). Ou seja:

$$A = A_t + A_{prop} + A_{proc} + A_{tAck} + A_{prop} + A_{proc} \quad (5)$$

O tempo de transmissão do pacote é determinístico, pois está relacionado ao tempo gasto desde quando o primeiro bit do quadro começa a sair do transmissor, até quando o último bit termina de sair. Esse tempo depende da quantidade de bits transmitidos (F) e da taxa de bits nominal do meio (B), sendo dado por:

$$A_t = \frac{F}{B} \quad (6)$$

O atraso de propagação é tempo gasto pelo sinal para se propagar desde o transmissor até o receptor. Independe da quantidade de bits transmitidos, pois é uma característica física do meio. Tais características podem variar muito no tempo influenciando na propagação do sinal.

O atraso no processamento de recepção da mensagem está relacionado ao tempo decorrido para ler o cabeçalho e interrupção do sistema operacional. Este tempo varia dependendo do sistema operacional, portanto, é um tempo não determinístico.

Adicionalmente, existe o atraso de processamento de envio da mensagem em decorrência do tempo gasto para a montagem da mensagem, e cabeçalho. Este atraso é variável e não determinístico, pois o processo de envio concorre com outros processos é dependente do sistema operacional.

Além disso, existe o atraso da camada MAC que está diretamente relacionado ao estado do canal, ou seja, neste momento o nó está disputando com os demais nós um momento para enviar seu pacote.

Normalmente os valores mais significativos são A_t e A_{prop} e os demais termos podem ser desprezados. Para um cálculo aproximado, o tempo total necessário para o envio de um pacote pode ser dado por:

$$A = A_t + 2A_{prop} \quad (7)$$

A Figura 40 representa o tempo total aproximado para o envio de um pacote e o recebimento do quadro de confirmação ACK.

Figura 40 – Tempo de envio de um pacote

O tempo total observado em ensaios práticos para enviar um quadro de dados e receber um ACK é menor que *500ms*, ou seja, é menor que os intervalos de tempo estabelecido pelo temporizador. Isso permite considerar o tempo de recebimento de um ACK antes de tentar reenviar um quadro de dados.

O código simplificado das funções `enviar_mensagemTx()` e `receber_mensagemRx()`, responsáveis, respectivamente, pela transmissão e recebimento de mensagens é apresentado nas Figuras 41 e 42.

Cabe ressaltar que o protocolo apresenta duas maneiras de remover um nó crítico da tabela de vizinhos.

A primeira é por falta de envio do quadro *Hello*. Tal quadro verifica a existência de nós falhos, porém, é independente de um determinado tráfego de dados, servindo apenas para manter a tabela de vizinhos atualizada.

A segunda maneira é por falta de comunicação durante o envio de um quadro de dados. Esta permite uma reorganização rápida da rede caso um nó pertencente a uma rota em vigência perca a comunicação com seus vizinhos, seja por falta de bateria, obstáculos ou interferências eletromagnéticas. Isso diminui o atraso do encaminhamento do pacote para o próximo vizinho.

```

/*****
* FUNÇÃO PARA ENVIAR MENSAGEM TX
*****/

void enviar_mensagemTx(void){
    ...
    if (!taProntoEnvio()) return; // Verifica estado de máquina da mensagem
    if (inicio == fim) return; // Tabela de mensagem está vazia
    // Se a mensagem for um quadro de dados verificar critério para repasse do pacote
    if (TabelaMensagem[inicio].tipo == MSGDADOS){
        if (TR-- != 0) return; // Enquanto temporizador não zerar retorna.
        if (NT==MAXNT){
            NT = 0;
        }
        // Calcula o vizinho para próximo salto baseado na distância Euclidiana
        TabelaMensagem[inicio].dados.msgdados roteamento.destinatario =
            proxSalto_rotGeog(TabelaMensagem[inicio].dados.msgdados.posXd,
                TabelaMensagem[inicio].dados.msgdados.posYd);
        // Incrementar fila caso critério drop seja estabelecido
        if (CRITERIODROP == TRUE){
            inicio++;
            inicio %= TBMSGNMAX;
            NT=0;
            TR=0;
            return;
        }
        // Chamar cálculo do temporizador
        NT++;
        TR = temporizador(NT); // Retorna quatro elevado a NT
        ...
        if (NT == MAXNT){ // Número de tentativas atingiu valor máximo maxNT
            for (int i = 1; i < TBVIZNMAX; i++){ // Remover nó da tabela de vizinhos
                if (tabelaViz[i].valido == TRUE)
                    if (tabelaViz[i].Ident == TabelaMensagem[inicio].dados.msgdados roteamento.destinatario){
                        tabelaViz[i].valido = FALSE; // Alterar status do nó afim de remover-lo da tabela
                        ...
                    }
            }
        }
    }
    ...
    //Chamar a função MCPSPDataRequest para enviar todos os tipos de mensagem
    TX_msg.u8BufSize = sizeof(struct mensagem_s);
    MCPSPDataRequest(&TX_msg);
    ...
    // Incrementar a fila de mensagem imediatamente após o envio de um quadro ACK e HELLO
    if (TabelaMensagem[inicio].tipo == ACK || TabelaMensagem[inicio].tipo == HELLO){
        inicio++;
        inicio %= TBMSGNMAX;
    }
}

```

Figura 41 - Função enviar mensagem TX


```

/*****
* FUNÇÃO PARA RECEBER MENSAGEM RX
*****/
void receber_mensagemRx(void)
{
    ...
    // Se a mensagem recebida é um quadro ACK
    if (msg.tipo == ACK){
        ...
        // Verificar Tabela de mensagem para remover o quadro de dados da fila
        if (TabelaMensagem[inicio].tipo == MSGDADOS){
            // Conferir se o quadro ACK recebido corresponde ao quadro de dados presente na fila
            if(msg.dados.ack.Origem == TabelaMensagem[inicio].dados.msgdados.enderecamento.remetente
                && msg.dados.ack.PacketNumber == TabelaMensagem[inicio].dados.msgdados.packetNumber){
                inicio++; // Retirar mensagem de dados da fila de mensagens
                inicio %= TBMSGNMAX;
                NT=0;
                TR=0;
            }
        }
    }
    ...
    // Se a mensagem recebida é um quadro HELLO atualizar tabela de vizinhos
    else if (msg.tipo == HELLO){
        uint32_t no_atual = msg.dados.hello.NoAtual;
        int index = posicao_valida(no_atual);
        AdicionaMsg_TabViz(index,msg.dados.hello.NoAtual,msg.dados.hello.posX,msg.dados.hello.posY);
    }
    // Se a mensagem recebida é um quadro de DADOS
    else if (msg.tipo == MSGDADOS){
        ...
        MLMELinkQuality(&u8Lqi); // Armazenar a qualidade do link do pacote recebido
        // Adicionar quadro de ACK correspondente ao quadro de dados recebido a fila de mensagens
        acrescenta_ack(msg.dados.msgdados.roteamento.remetente,msg.dados.msgdados.packetNumber,
            msg.dados.msgdados.enderecamento.remetente,u8Lqi); // chama a função preencher_tabela()

        // Se o próprio nó é o DESTINATÁRIO FINAL do fluxo de mensagens abrir payload
        if (ID_ME == msg.dados.msgdados.roteamento.destinatario && ID_ME ==
            msg.dados.msgdados.enderecamento.destinatario){
            ... // processar dados armazenados no payload
        }
        ...
        // Se o nó é intermediário, adicionar quadro de dados á fila de mensagem
        else if (ID_ME == msg.dados.msgdados.roteamento.destinatario){
            acrescenta_msgdados(msg); // chama a função preencher_tabela()
        }
    }
}

```

Figura 42 - Função receber mensagem RX

Mediante ao exposto nesta seção, a técnica abordada para o roteamento dos pacotes deve escolher o próximo nó a repassar o pacote, assegurando o caráter auto-curável da rede e priorizando o menor número de saltos na comunicação fim a fim.

8.4.4 Cenário de desenvolvimento

Com intuito de analisar o critério de repasse do pacote abordado na seção anterior, foram criados dois cenários para a avaliação do protocolo proposto.

O estudo de caso para o posicionamento dos nós chegou às configurações apresentadas na Figura 43 (Rede A) e na Figura 44 (Rede B). Trata-se de uma rede de sensores *mesh* formada pelos nós extremos NTC1 e LP1 e pelos nós intermediários SN1, SN2, LP2 e NTC2. Os nomes dados aos nós fazem referência as características de hardware do módulo de comunicação sem fio, abordado na Seção 8.1.

Os círculos e retângulos simbolizam obstáculos que interferem na comunicação da rede e, conseqüentemente, no alcance de rádio de cada nó.

Todos os dispositivos da rede são FFD e oferecem vias de comunicação redundantes para aumentar a confiabilidade da rede. Os cenários modelos garantem uma cobertura global da rede. Ou seja, todos os nós da rede estão conectados

Vale ressaltar que o posicionamento geográfico de cada nó adotado pelo roteamento geográfico *Greedy* foi baseado em um sistema de endereçamento local, onde as coordenadas x e y são válidas somente para os dispositivos de comunicação sem fio da rede.

Figura 43 - Representação gráfica do cenário de desenvolvimento – Rede A

Figura 44 - Representação gráfica do cenário de desenvolvimento – Rede B

9 MÉTODOS

Esta seção apresenta um resumo de como gerar um projeto a partir do protocolo SMAC. A descrição do método adotado para inserção do protocolo *Greedy* ao protocolo SMAC também será abordada e apresentada a seguir.

9.1 Requisitos Iniciais

Para o desenvolvimento do protocolo proposto na Seção 8, é necessária a instalação de alguns *softwares* adequados para trabalhar com a plataforma SMAC. Inicialmente, devem ser instalados o aplicativo *BeeKit* da *Freescale*, e os *drivers*¹⁷ necessários para correto funcionamento dos módulos de comunicação sem fio.

O *Beekit* [110] é uma aplicação de *software* que proporciona uma interface gráfica do usuário (GUI), onde o utilizador pode criar, modificar, guardar ou atualizar soluções de rede sem fio baseadas nas *stacks* protocolares da *Freescale*. Esta ferramenta permite ao usuário configurar vários parâmetros antes mesmo de criar um projeto. Isto reduz a necessidade de percorrer e preparar vários arquivos distintos e de configurar manualmente diversos parâmetros, como seleção de canal, seleção de *hardware*, potência de saída da antena etc.

Basicamente, o *Beekit* está relacionado à seleção do *codebase*, o qual corresponde à pilha de protocolo de rede utilizada em conjunto com o tipo de processador. No caso deste trabalho, o *codebase* utilizado corresponde ao protocolo SMAC em conjunto com o processador ARM7.

Figura 45 - Estrutura *BeeKit*
Fonte: [110]

¹⁷ *Softwares* utilizados pelos controladores do Sistema Operacional para abstrair o *hardware* da aplicação que o usa.

Adicionalmente, os projetos criados no *Beekit* devem ser exportados para o aplicativo IAR - *Embedded Workbench* [111]. Este aplicativo utiliza o arquivo estruturado XML (*eXtensible Markup Language*) do *BeeKit* para ter acesso aos dados do projeto. Trata-se de um compilador e *debugger* para “linkar” a pilha SMAC com os arquivos necessários para aplicação.

Em resumo o IAR-EW fornece uma IDE (*Integrated Development Environment*) para adicionar a aplicação do usuário aos módulos de comunicação sem fio. É feito *linking*¹⁸ entre a pilha SMAC e as aplicações do usuário, com um programa desenvolvido na linguagem C. Após essa “*linkagem*”, o *software* pode ser “*debugado*”, usando a porta JTAG das placas (através de USB), ou diretamente gravado na memória da placa, usando a mesma porta.

Existem cinco passos a serem executados para a criação de um projeto. Isto abrange desde a inicialização do *Beekit* até a fase final de importação do arquivo XML para o software IAR – EW. Cada um desses passos é ilustrado como segue:

1. Instalar e inicializar o aplicativo *Beekit*.

Figura 46 - Software *Beekit*

2. Selecionar o *codebase* SMAC em conjunto com o processador ARM7.

¹⁸ Refere-se à etapa de união dos programas, em suas partes comuns.

Figura 47 - Seleção do Codebase SMAC

3. Criar projeto. O usuário pode acrescentar um novo projeto ou utilizar um projeto *template* já existente no SMAC para adicionar sua aplicação.

Figura 48 - Seleção de projetos: (a) Template SMAC (b) Adição de um novo projeto

4. Customizar e validar o projeto. O processo de validação ajuda o usuário a encontrar possíveis inconsistências entre o *hardware* selecionado e as propriedades do projeto. Essas propriedades podem ser modificadas pelo usuário no *Beekit* e estão relacionadas a parâmetros como tipo de *hardware*, seleção do canal de rádio, inclusão de suporte para LCD, habilitação de segurança na transmissão dos dados, configuração da potência de saída, dentre outros.

Figura 49 - (a) Seleção de tipo de *hardware* (b) Validação do projeto

5. Exportar projeto do *BeeKit* para o IAR - *Embedded Workbench*.

Figura 50 - Exportando projeto do *BeeKit* para o IAR – EW.

Uma vez que o projeto é exportado para o IAR – EW, o processo de adição da aplicação do usuário ao código fonte do SMAC pode ser iniciado.

A aplicação pertinente a este trabalho é baseada na inserção de um protocolo de rede ao protocolo SMAC. A seção a seguir apresenta a descrição do método desenvolvido para tal propósito.

9.2 Método para implementação SMAC

A plataforma SMAC é bastante simples, formada por um conjunto limitado de regras, responsáveis por garantir a correta execução do protocolo. Uma vez que as

funções de primitivas, os tipos de mensagens e a estrutura de pacotes SMAC são conhecidos, é necessário que o usuário considere as instruções contidas em tais regras para o desenvolvimento de uma aplicação.

A importância da definição de um método baseado no conjunto de regras de protocolo SMAC é comprovada pelo surgimento de um maior controle sobre os recursos de execução do protocolo.

Diante do exposto, o método descrito nesta seção é simples e funcional, oferecendo os comandos necessários para o desenvolvimento de uma aplicação no SMAC.

Os seis passos fundamentais que configuram o método para implementação SMAC são apresentados a seguir:

- 1) Inicialmente devem ser declaradas as variáveis globais `message_t`, abordadas na Seção 8.2.4. A aplicação deve declarar, por exemplo, uma mensagem para transmissão, TX, e outra mensagem para recepção, RX.
- 2) Um *buffer* de dados precisa ser criado para cada mensagem usando a macro `smac_pdu_size`. Assim, caso a mensagem TX tenha 18 *bytes* de dados, o *buffer* pode ser declarado como dados `uint8_t data[smac_pdu_size(18)]`.
- 3) Como as mensagens são processadas por seus estados, cada mensagem deve ser iniciada usando a macro `MSG_INIT`. Tal macro é utilizada como segue:


```
MSG_INIT([mensagem],[ponteiro para o buffer de dados],  
[ponteiro callback]).
```
- 4) As funções do sistema precisam ser inicializadas, incluindo o rádio por meio da função de primitiva `MLMERadioInit()`. Adicionalmente, as primitivas de ajuste de potência de saída, `MLMEPAOutputAdjust`, e seleção do canal, `MLMESetChannelRequest`, também são utilizadas para inicialização de parâmetros do sistema.

- 5) Para a transmissão e recepção de mensagens a aplicação deve utilizar, respectivamente, as funções de primitivas `MCPSPDataRequest` e `MLMERXEnableRequest`.
- 6) Por fim, a função `process_radio_msg` precisa ser chamada periodicamente para processar as mensagens na fila.

As considerações descritas acima são abordadas no código da Figura 51. Trata-se de um código simplificado, em linguagem de programação C, apenas para orientação das regras de protocolo SMAC. No entanto, neste código não há capacidade de roteamento de mensagens até o destino. Esse trecho de código serve de apoio e pode ser utilizado como base para desenvolvimento das mais diversas aplicações.

Basicamente, a camada de aplicação chama as primitivas da camada de enlace para habilitar uma aplicação específica do usuário.

A camada de aplicação deve implementar as funções de chamada de retorno (`callback_t`), exigidas pelo SMAC. Tais funções permitem que as camadas mais baixas notifiquem a camada de aplicação que um evento ocorreu.

No código abaixo, a função de retorno `a_simple_rx_callback_fn()` é chamada quando uma nova mensagem é recebida.

```

/*****
* CÓDIGO SIMPLIFICADO: EXEMPLO SMAC SEM ROTEAMENTO DE MENSAGENS
*****/

void a_simple_rx_callback_fn (void)
{
    ...
}

...
message_t a_TX_msg;
message_t a_RX_msg;

uint8_t dataTX[smac_pdu_size(TX_SIZE)];
uint8_t dataRX[smac_pdu_size(RX_SIZE)];
...
uint8_t main()
{
    ...
    /* Inicia a mensagem TX */
    MSG_INIT(a_TX_msg, &dataTX, NULL);
    a_TX_msg.u8BufSize = TX_SIZE;

    /* Inicia a mensagem RX */
    MSG_INIT(a_RX_msg, &dataRX, a_simple_rx_callback_fn);
    a_RX_msg.u8BufSize = RX_SIZE;
    ...

    /* Inicia o Radio transceiver */
    MLMERadioInit();
    ...
    for(;;)
    {
        /* Processa as mensagens na fila */
        process_radio_msg();
        ...
        /* Põe na fila a mensagem a ser transmitida */
        if([Precisa transmitir uma mensagem]){
            MCPSPDataRequest(&a_TX_msg);
        }
        ...
        if([Precisa receber uma mensagem]){
            MLMERXEnableRequest(&a_RX_msg, TIME_OUT);
        }
    }
}

```

Figura 51 - Exemplo de código SMAC

9.3 Método para inserção do protocolo de rede no SMAC

Embora a plataforma SMAC possa ser empregada em redes de topologia *mesh*, ela não apresenta um mecanismo pronto de roteamento de mensagens. Tal mecanismo, então, deve ser desenvolvido pelo usuário da aplicação.

Para dar início à descrição da metodologia para inserção do protocolo de rede, o projeto *Generic Application* deverá ser exportado do *Beekit*. Trata-se de um projeto de aplicação genérica e dará suporte às modificações necessárias para o desenvolvimento do protocolo proposto.

Vale ressaltar que os códigos de roteamento de dados utilizados em simuladores de rede, como o NS-2, escritos em C e C++, também poderão ser adequados e portados para a plataforma SMAC.

O SMAC utiliza uma biblioteca de rotinas para interface com as aplicações. Tal biblioteca contém um conjunto de primitivas necessárias para a cooperação entre tarefas de uma aplicação. Mediante a manipulação de tais primitivas SMAC, é possível elaborar um protocolo para redes *mesh multi-hop*.

O código fonte `generic_app.c` é utilizado como código base para o desenvolvimento do projeto. Este código está presente na raiz de pastas do projeto *Generic Application* dentro da pasta de arquivo Fonte da aplicação.

Figura 52 - Código fonte `generic_app.c`

Inicia-se aqui o processo de reestruturação do arquivo fonte `generic_app.c` para a criação da camada de rede no protocolo SMAC. Fazem parte desta criação o protocolo de rede baseado no algoritmo geográfico, bem como os quadros de mensagens *Hello*, *ACK* e de dados, responsáveis pelo funcionamento do protocolo e troca de informações na rede.

O principal processo de comunicação no protocolo SMAC funciona em torno das funções de primitivas `MCPSPDataRequest` e `MLMERXEnableRequest`. Para habilitar a recepção de mensagens, o usuário deve chamar a função `MLMERXEnableRequest` e, para solicitar a transmissão de uma mensagem, deve chamar a função `MCPSPDataRequest`.

A Seção 8 apresentou a arquitetura desenvolvida para o funcionamento do protocolo SMAC em redes *mesh multi-hop*. Para incluir a capacidade de roteamento de

mensagens, a aplicação deve adicionar ao código fonte do SMAC as rotinas com o princípio de funcionamento do protocolo da camada de rede. Tal fato inclui a ativação das rotinas para a transmissão e recebimento de mensagens em conjunto com a chamada, de forma adequada, das funções de primitivas SMAC responsáveis pelo envio e recepção de mensagens (`MCPSPDataRequest` e `MLMERXEnableRequest`).

O processo de recepção de mensagens está vinculado à chamada periódica da função de primitiva `MLMERXEnableRequest`. Esta função habilita o rádio para o recebimento de pacotes. As mensagens recebidas são armazenadas em um *buffer* e aguardam para serem processadas. O processo é bastante simples: quando uma mensagem é recebida, a *flag* de indicação de recebimento de mensagens (`gRadioEvtFlags.Bits.data_indication_flag`) é setada para `TRUE` e a função de retorno da mensagem RX (`application_rx_cb()`) atualiza a *flag* de ativação da função de recebimento (`dataIndicationRX`). Esta *flag* é uma confirmação de que a mensagem foi recebida e copiada para o *buffer* com sucesso.

Após receber uma mensagem, o sistema operacional do nó precisa entregar a mensagem ao processo que irá consumi-la. A aplicação deve chamar a função de recebimento de pacotes para identificar o tipo de mensagem recebida e posterior processamento das mesmas. Para cada tipo de mensagem recebida (`ACK`, *Hello* ou dados) há um processamento específico. O comportamento da função de recebimento para cada tipo de mensagem pode ser avaliado na descrição da função `receber_mensagemRx()`, presente na Figura 42.

Para satisfazer a propriedade de ordenação, os nós enviam as mensagens na ordem que estão armazenadas na tabela de mensagens, descrita na Seção 8.4.1. Antes de enviar uma mensagem, o nó consulta o algoritmo de roteamento geográfico para decidir o próximo salto da mensagem, função `proxSalto_rotGeog()`. Uma vez definido o nó vizinho, a função `MCPSPDataRequest` deve ser chamada para realizar a transmissão da mensagem. O princípio de envio das mensagens pode ser visto na função `enviar_mensagemTx()` da Figura 41.

Por fim, cabe ressaltar que, antes das mensagens serem recebidas ou enviadas, é necessário aguardar o estado de máquina das mensagens. Uma vez que as mensagens estão com seu estado completo, estas podem ser processadas.

Consoante a tais diretrizes, o código fonte `generic_app.c` foi modificado para atender os requisitos do protocolo aqui proposto. Procurando esquematizar as principais modificações, as Figuras 53, 54, 55, 56 e 57 apresentam uma simplificação

do código `generic_app.c` alterado. Os comandos necessários para dar suporte ao protocolo de rede são apontados e obedecem as diretivas de implementação do protocolo SMAC abordadas na Seção 9.2.

```
/* *****  
 * CÓDIGO generic_app.c ALTERADO - PARTE 1  
 * ***** */  
...  
// (A) Criar a estrutura de quadro de mensagens responsáveis pela troca de informação  
// na rede.  
struct mensagem_s{  
    ... // declarado aqui os quadros Hello, ACK e de dados vide Figura 33.  
};  
...  
// (B) Declaração das mensagens RX e TX e criação do buffer de dados.  
message_t TX_msg;  
message_t RX_msg;  
uint8_t dataTX[smac_pdu_size(sizeof(struct mensagem_s))];  
uint8_t dataRX[smac_pdu_size(sizeof(struct mensagem_s)) + 1];  
...  
// (C) Declaração das funções da máquina de estado das mensagens TX e RX.  
bool_t taProntoEnvio(void){  
    switch (TX_msg.u8Status.msg_state) {  
        case MSG_TX_ACTION_COMPLETE_SUCCESS:  
        case MSG_TX_ACTION_COMPLETE_FAIL:  
        case MSG_TX_ABORTED:  
        case MSG_TX_ACTION_COMPLETE_CHANN_BUSY:  
            return TRUE;  
        default:  
            return FALSE;  
    }  
}  
bool_t taProntoRecebimento(void){  
    switch (RX_msg.u8Status.msg_state) {  
        case MSG_RX_ACTION_COMPLETE_SUCCESS:  
        case MSG_RX_ACTION_COMPLETE_FAIL:  
        case MSG_RX_ABORTED:  
        case MSG_RX_TIMEOUT_FAIL:  
            return TRUE;  
        default:  
            return FALSE;  
    }  
}  
...
```

Figura 53 - Código `generic_app.c` alterado – Parte 1

```
/* *****  
* CONTINUAÇÃO: CÓDIGO generic_app.c ALTERADO - PARTE 2  
* ***** */  
...  
  
// (D) Declarar a função que habilita recebimento do próximo pacote.  
void hab_recebimento_prox_pacote(void)  
{  
    RX_msg.u8BufSize = sizeof(struct mensagem_s);  
    MLMERXEnableRequest(&RX_msg, 0);  
}  
...  
  
/* (E) Declarar as função de envio das mensagens em conjunto com a função de  
primitiva MCPSPDataRequest. Antes de enviar as mensagens armazenadas na tabela de  
mensagens consultar algoritmo de roteamento para definir próximo salto da mensagem.  
Função definida na Figura 41. */  
  
void enviar_mensagemTx(void){  
    if (!taProntoEnvio()) return;  
    ...  
    TX_msg.u8BufSize = sizeof(struct mensagem_s);  
    MCPSPDataRequest(&TX_msg);  
    ...  
}  
...  
  
/* (F) Declarar as função de recebimento de mensagem. Para cada tipo de mensagem  
recebida há um processamento específico, conforme função definida na Figura 42. */  
  
void receber_mensagemRx(void){  
    ...  
    if (msg.tipo == ACK){  
        ... // Processa ACK recebido  
    }  
    else if (msg.tipo == HELLO){  
        ... // Processa HELLO recebido e atualiza tabela de vizinhos do nó  
    }  
    else if (msg.tipo == MSGDADOS){  
        ... /* Processa mensagem de dados recebida, se for um nó intermediário  
encaminha o pacote para ser reenviado segundo algoritmo de roteamento geográfico.*/  
    }  
}  
...  

```

Figura 54 - Código generic_app.c alterado – Parte 2

```

/*****
* CONTINUAÇÃO: CÓDIGO generic_app.c ALTERADO - Parte 3
*****/
...

// (G) Declarar a função principal MAIN.
void Main(void){
    ...
    // (H) Inicializar as mensagem TX e RX.
    MSG_INIT(TX_msg, &dataTX, NULL);
    MSG_INIT(RX_msg, &dataRX, (callback_t) (application_rx_cb));
    ...

    // Chamar função de inicialização de hardware e periféricos do MC1322X.
    generic_app_init();

    /* Chamar função de retorno do timer SMAC (TMR0_SMAC), acessada a cada intervalo
    de tempo definido por tpEnviotbMsg. O temporizador (T) de reenvio das mensagens e dados
    e o timer das mensagens Hello são vinculados aos ciclos de estouro deste timer. */

    Timer_Setup(TMR0_SMAC);
    ...

    for (;;) {

        // (I) Chamar a função process_radio_msg periodicamente.
        (void)process_radio_msg();

        // (J) INICIAR PROCESSO DE ENVIO DAS MENSAGENS
        // Enviar mensagens de Hello periodicamente
        /* A cada intervalo de tempo tpMsgHello o timer HELLO estoura e adiciona
        um quadro Hello a tabela de mensagens a serem enviadas.*/
        if (TMR0_SMAC_HELLO == TRUE){
            acrescenta_hello(); // chama a função preencher_tabela()
            ...
        }

        // Acessar tabela de mensagens a serem enviadas periodicamente.
        /* A cada tpEnviotbMsg (100ms) a tabela de mensagens é acessada para
        enviar as mensagens presentes na fila */
        if (TMR0_SMAC_TABTX == TRUE) {
            (void)enviar_mensagemTx();
            ...
        }
        ...
    }
}
```

Figura 55 - Código generic_app.c alterado – Parte 3


```

/*****
* CONTINUAÇÃO: CÓDIGO generic_app.c ALTERADO - Parte 4
*****/

...
// (K) INICIAR PROCESSO DE RECEBIMENTO DAS MENSAGENS
//Aguardar máquina estado da mensagem
if (!taProntoRecebimento()) continue;
...
if (dataIndicationRX == TRUE){ // Flag de recebimento de mensagem
    if (RX_msg.u8Status.msg_state == MSG_RX_ACTION_COMPLETE_SUCCESS){
        (void)receber_mensagemRx();// Processa mensagem recebida
    }
    ...
}

// (L) Chamar primitiva MLMERXEnableRequest para habilitar o rádio
(void)hab_recebimento_prox_pacote();
} // FIM do for.
} // FIM da função main.

...

// (M) Definição da chamada da função de retorno quando uma mensagem RX é recebida
void application_rx_cb(void)
{
    if (FALSE == gRadioEvtFlags.Bits.data_indication_flag) return;
    gRadioEvtFlags.Bits.data_indication_flag = FALSE;
    dataIndicationRX = TRUE; // habilita função de recebimento da mensagem
}

...

// (N) Definição da função de inicialização do sistema.
static void generic_app_init(void)
{
    u8AppChann = CHANNEL_NUMBER;
    u8AppPower = OUTPUT_POWER;
    ...
    ITC_EnableInterrupt(gTmrInt_c);
    CRM_RegisterISR(gCrmKB4WuEvent_c, Switch_1_isr);
    CRM_RegisterISR(gCrmKB5WuEvent_c, Switch_2_isr);
    CRM_RegisterISR(gCrmKB6WuEvent_c, Switch_3_isr);
    CRM_RegisterISR(gCrmKB7WuEvent_c, Switch_4_isr);
    IntDisableAll();
    ...
}

```

Figura 56 - Código generic_app.c alterado – Parte 4

```

/*****
* CONTINUAÇÃO: CÓDIGO generic_app.c ALTERADO - Parte 5
*****/

...
ResetMaca();
MLMERadioInit();// Primitiva de inicialização do rádio
MLMSetChannelRequest(u8AppChann); // Primitiva de seleção do canal de rádio
IntEnableAll();
Uart_Init();
LED_Init();
KbGpioInit();

...
#if SW_INT_ENABLE == TRUE
MLMSetWakeupSource(gExtWuKBI4En_c | gExtWuKBI5En_c | gExtWuKBI6En_c |
    gExtWuKBI7En_c, 0x00, 0x0F);
#endif

...
(void)MLMEPAOutputAdjust(u8AppPower); // Primitiva de ajuste da potência de
saída

...
// Habilita Módulo de segurança SMAC - criptografia AES.
#if SMAC_FEATURE_SECURITY == TRUE
(void)CipherEngineInit();
(void)CipherConfigure(AES_DEFAULT_MODE, &CTR_Key, &Ctr_Value);
#endif

...
}
...

```

Figura 57 - Código generic_app.c alterado – Parte 5

A partir das considerações metodológicas abordadas acima, a capacidade de roteamento de dados é adicionada ao protocolo SMAC. Tal capacidade é apontada pelo estudo mediante a inserção de um protocolo de rede às rotinas de interface do protocolo SMAC.

Uma vez que o *link* entre a pilha SMAC e a aplicação do usuário está completa, o código em C é migrado para o código de *firmware* dos módulos de comunicação sem fio. Esses módulos são conduzidos para o cenário de desenvolvimento e submetidos a testes experimentais para avaliação do protocolo desenvolvido.

10 AVALIAÇÃO DA PROPOSTA

A presente seção tem por foco a avaliação do critério de repasse de pacotes proposto e implementado no protocolo SMAC. Para tal, os nós de comunicação sem fio presentes nos cenários de desenvolvimento da Seção 8.4.4 foram submetidos a ensaios experimentais para coleta de dados.

Esses cenários possuem uma rede com topologia *mesh* onde o principal desafio é o roteamento de pacotes por meio dos nós da rede. Para tal, são necessários múltiplos saltos, através de nós intermediários, até que o pacote alcance o destino desejado.

O maior objetivo dos testes realizados é verificar a eficiência do protocolo SMAC com o roteamento de pacotes proposto, na escolha do próximo nó a repassar o pacote. O protocolo deve minimizar o número de saltos entre o nó fonte e o nó destino, assegurando a característica auto-curável da rede.

O *sniffer* USB MC1322X foi utilizado para captura de pacotes em tempo real, permitindo, assim, o rastreamento das mensagens e depuração do protocolo SMAC.

A Figura 58 apresenta a tela da ferramenta de *software* utilizada para registrar o tráfego de pacotes interceptado pelo *sniffer*. Os dados armazenados são posteriormente exportados para uma ferramenta de análise de dados, onde é possível gerar gráficos com o desempenho do protocolo SMAC.

Figura 58 - Analisador de rede

Inicialmente, foi realizada uma avaliação da distância máxima e mínima entre os nós em função dos limites extremos da faixa de potência de saída dos dispositivos. Esta análise é apresentada na Tabela 8.

Tabela 8 – Distância entre nós em função da potência de saída

POTÊNCIA	ALCANCE
2,8mW / 4,5dBm	280m
1μW / -30dBm	5m

Em detrimento da limitação de dispositivos para testes, o cenário de desenvolvimento se restringiu a redes com cinco nós estáticos de comunicação sem fio. Além disso, a utilização de *sniffers* para a captura de mensagens em trânsito “no ar” é limitada pela sensibilidade de sua antena.

Ensaio práticos demonstraram a ineficiência do uso do *sniffer* para extensões superiores a seis metros. Assim, em razão da atuação localizada do *sniffer*, a distância estabelecida entre os nós das redes é restringida a poucos metros.

Os cenários de avaliação propostos são formados por duas redes distintas (Rede A e Rede B). Isso porque o posicionamento geográfico dos nós e as restrições de cobertura impostas por obstáculos são diferentes. Tal ocorrência afeta a formação da tabela de vizinhos de cada nó e, conseqüentemente, os caminhos estabelecidos para roteamento dos dados entre a fonte e o destino.

Desta forma, visando uma análise característica a cada cenário de desenvolvimento, os resultados obtidos nos ensaios experimentais das redes A e B serão apresentados separadamente. Tais resultados são apresentados a seguir.

10.1 Rede de sensores sem fio - A

A Figura 59 apresenta as coordenadas geográficas e o alcance do rádio dos nós para a rede A. Adicionalmente, é exibida uma demonstração da formação da tabela de vizinhos (TV) particular a cada um dos nós.

Trata-se de uma rede formada pelos nós extremos NTC1 e LP1 (*Network Coordinator 1* e *Low Power 1*) e pelos nós intermediários SN1, SN2, NTC2 e LP2 (*Sensor Node 1*, *Sensor Node 2*, *Network Coordinator 2* e *Low Power 2*).

Figura 59 – Rede de sensores sem fio - A

Para identificação dos nós na rede, cada nó possui um MAC ID resumido, constando apenas os três últimos termos do endereço indicado na placa impressa de cada um deles. O endereço de cada nó é apresentado na Tabela 9.

Tabela 9 – Identificação dos nós da rede

Nó	MAC ID resumido
NTC1	0X983
NTC2	0X7FD
SN1	0X9F2
SN2	0X872
LP1	0X793
LP2	0X7F8

Uma vez definido o posicionamento geográfico dos nós na rede, é necessário estabelecer dois nós para serem origem e destino do fluxo de mensagens. Tal fluxo é estabelecido entre o nó de origem NTC1 e o destinatário LP1.

O tráfego de mensagens na rede é formado pelo envio periódico dos quadros de mensagens *Hello* e dados. Adicionalmente, quadros de confirmação (ACK) são enviados automaticamente em resposta ao recebimento de um quadro de dados.

Os intervalos de tempo que cada quadro de mensagem é enviado na rede e os tipos de transmissão estão definidos na Tabela 10.

Tabela 10 - Tráfego de mensagens

Quadro de mensagem	Intervalo de tempo (segundos)	Tipo de Transmissão	Descrição do Fluxo
HELLO	20	<i>Broadcast</i>	Enviados por todos os nós da rede para formação e manutenção da tabela de vizinhos.
Dados	30	<i>Unicast</i>	Enviados periodicamente do nó NTC1 ao nó LP1.
ACK	Automático	<i>Unicast</i>	Enviados automaticamente pelos nós da rede para cada quadro de dados recebido.

Para uma análise efetiva do desempenho do protocolo proposto, alguns nós da rede são forçados a condições críticas de funcionamento. Isto permitirá verificar a capacidade de tolerância a falhas, bem como o caráter auto-organizável da rede.

As condições críticas de funcionamento do nó simulam as possíveis falhas de um nó sensor da rede, seja por motivos como falta de energia, falta de visibilidade para outro nó da rede, interferências ou até mesmo algum dano físico. Tais condições são instituídas à rede mediante o desligamento do rádio do nó durante uma determinada rota.

Em uma rede de sensores sem fio, falhas são possíveis e aceitáveis. No entanto, a rede deve corrigi-las de maneira automática. No presente trabalho, a falha de um nó que faz parte da rota de um pacote não deve impedir que o pacote chegue ao seu destino. Isso porque a topologia da rede oferece caminhos alternativos para que os pacotes alcancem o seu destino.

Para garantir essa disponibilidade de rotas alternativas para os pacotes, o protocolo de roteamento proposto reorganiza a rede e traça novas rotas em substituição àquelas interrompidas pela falha de um nó.

Caso um nó pertencente a uma rota perca a comunicação com seu nó vizinho antecessor, haverá um critério de tentativas para o reenvio do pacote. O nó antecessor tentará enviar o pacote três vezes para este nó falho. Caso não receba um ACK, o nó falho será removido da sua tabela de vizinhos.

A Tabela 11 apresenta uma descrição das falhas introduzidas na rede A. Os rádios dos nós SN1 e SN2 são desligados durante a coleta de dados para avaliar o desempenho do protocolo de roteamento implementado no SMAC.

Tabela 11 – Falhas introduzidas na Rede A

Falha introduzida	Nó	Tempo de Ausência do Nó
Rádio desligado	SN2	337.749s
Rádio desligado	SN1	195.318s

A Tabela 12 apresenta os fluxos de mensagens quando os nós da rede estão com potência máxima, a uma distância de poucos metros um dos outros. São estabelecidas três rotas pela rede: uma rota ótima e duas rotas alternativas.

Tabela 12 – Fluxo de mensagens na Rede A

Posição do nó na rede	Fluxo ÓTIMO Rota 1	Fluxo Alternativo Rota 2	Fluxo Alternativo Rota 3
Origem	NTC1	NTC1	NTC1
Intermediário	SN1	SN1*	LP2
Intermediário	SN2*	NTC2	NTC2
Destino	LP1	LP1	LP1

* Nós sujeitos à falha

Inicialmente, a rota 1 é traçada entre o nó fonte NTC1 e o nó de destino LP1. Esse percurso é a rota ótima segundo o critério de repasse de pacote *Greedy* e fornece a menor distância entre os nós fonte e destino.

Posteriormente, com o intuito de verificar o caráter auto-curável da rede, o rádio do nó SN2 é desligado e a rota ótima é interrompida. Ao detectar a perda de comunicação, o nó SN1, antecessor ao nó SN2, reenvia a mensagem para este nó de acordo com o critério de tentativas de reenvio de pacotes. Caso não receba um ACK, o nó falho SN2 é removido da tabela de vizinhos do nó SN1 e uma segunda rota, a rota 2, é formada até o destino.

Após um determinado tempo, uma segunda falha é introduzida à rede e o funcionamento do rádio do nó SN1 é interrompido. Do mesmo modo indicado anteriormente, o nó NTC1 tenta restabelecer a comunicação com o nó SN1 através do reenvio de mensagens. Caso a comunicação não seja efetivada, uma terceira rota é estabelecida entre nó NTC1 e LP1: a rota 3.

As representações gráficas do desempenho do protocolo durante o intervalo de tempo em que as condições de falhas são introduzidas na rede são exibidas a seguir.

A Figura 60 apresenta uma análise quantitativa das mensagens enviadas na rede. São enviadas, ao todo, 22 mensagens de dados entre o nó NTC1 e LP1. No entanto, como são necessários três saltos até o destino, o número completo de mensagens de dados enviadas é 66, em condições de pleno funcionamento de todos os nós.

Em detrimento das falhas introduzidas na comunicação da rede, são enviadas 6 mensagens de dados adicionais, contabilizando, ao todo, 72 mensagens de dados. Tais mensagens são necessárias para reorganização da rede e restabelecimento do fluxo de mensagens até o destino.

Figura 60 – Análise quantitativa das mensagens enviadas – Rede A

Em relação aos quadros de mensagem *Hello*, a maioria dos nós possui um mesmo número de mensagens enviadas. Somente os nós SN2 e SN1 transmitem um número inferior de mensagens devido à desativação do rádio dos mesmos.

De acordo com a Tabela 11, os nós SN2 e SN1 permanecem respectivamente, 337.749 e 195.318 segundos ausentes da rede. Em razão desta perda de comunicação, os nós SN2 e SN1 deixam de enviar, individualmente, em torno de 17 e 10 quadros *Hello*.

A quantidade de mensagens ACK enviadas e recebidas é igual, ou seja, nenhuma mensagem enviada foi perdida. Como são enviadas 22 mensagens de dados, o nó fonte NTC1 (0x983) recebe 22 mensagens de confirmação e o nó de destino LP1 (0x793) envia também, 22 mensagens ACK.

É observado ainda, que a quantidade total de mensagens ACK enviadas na rede (66) é três vezes maior que o número de mensagens de dados enviados pelo nó fonte (22). Isso porque são necessários três saltos para que a mensagem chegue ao destino. Desta forma, a cada salto da mensagem de dados, no mínimo uma mensagem ACK será recebida pelo emissor da mensagem.

Uma visão geral das mensagens enviadas na rede é exibida na Figura 61. Os quadros de mensagem de dados enviados na rede possuem 41 *bytes*.

É observado que, para cada mensagem de dados enviada, uma mensagem ACK é recebida. Ainda, cada pacote necessita, individualmente, de três saltos para atingir o nó de destino.

Figura 61 – Visão geral das mensagens enviadas

Para uma análise mais detalhada, o tempo de envio das mensagens de dados e de resposta de confirmação (ACK) para o pacote 0 é exibido na Figura 62. O tempo entre o envio de um quadro de dados e o envio de um quadro ACK, para cada salto, foi respectivamente: 0.076, 0.1 e 0.052 segundos.

Figura 62 – Tempo entre mensagens para o pacote 0

Para o envio de pacotes de 41 *bytes* a uma taxa de transferência de 250kbps, o tempo de transmissão do pacote, segundo a equação 6, é de 1.312ms.

O tempo total observado em ensaios práticos para o nó transmissor enviar um quadro de dados e o nó receptor enviar um quadro ACK é variável entre os limites de 0.019 e 0.101 segundos. Este tempo é flexível em decorrência dos atrasos não determinísticos de cada nó, abordados na Seção 8.4.3. Tais atrasos variam inesperadamente de um pacote para outro e são responsáveis pela imprecisão do tempo entre saltos.

A Figura 63 exibe os resultados coletados para o menor e o maior tempo entre a transmissão de um quadro de dados e o envio de uma confirmação, durante a transferência de um pacote, em um único salto, do nó transmissor para o nó receptor. Os valores entre os tempos t_2 e t_4 são registrados pelo analisador de rede que intercepta as mensagens que trafegam na rede.

É observado que a comunicação da rede é afetada pelos atrasos introduzidos por nó, a cada salto da mensagem. Quando o envio de um quadro de ACK atrasa, por exemplo, faz com que ocorra um *timeout* de espera por confirmação e conseqüente retransmissão.

Para o cálculo do tempo de propagação, é necessário relacionar a velocidade de propagação de uma onda eletromagnética (3×10^8 m/s) com o espaço percorrido pelo

signal. Uma vez que a distância entre os nós da rede é pequena (poucos metros), o tempo de propagação do sinal na rede é mínimo.

As diferenças entre os tempos de envio do quadro de dados e posterior ACK são em decorrência, principalmente, do tempo de processamento de envio e de recepção da mensagem, relacionadas a cada nó da rede.

Figura 63 – Tempo entre o envio de um quadro de dados e de ACK

Mediante ao exposto, o atraso de roteamento dos pacotes na rede relaciona-se com o tempo total que um pacote leva para deixar o nó fonte e chegar, em múltiplos saltos, ao nó de destino.

Esse tempo é obtido somando-se todos os atrasos e todos os tempos de processamento dos nós intermediários (isto é, que repassaram o pacote) até o momento que este atingiu o nó de destino.

Quanto menor o tempo entre t_2 e t_4 na Figura 63, menor será o atraso introduzido por salto na comunicação da rede. Isto porque, antes de retransmitir um pacote, o nó intermediário primeiramente envia um quadro de confirmação para o nó antecessor do pacote.

A figura a seguir ilustra a transmissão para o pacote 0 presente na Figura 62. Os tempos em vermelho são registrados pelo analisador da rede. São necessários três saltos até que o pacote oriundo do nó NTC1 chegue ao nó de destino LP1. Assim, o número de atraso introduzido por salto será maior, e o pacote irá gastar mais tempo para chegar ao nó de destino.

Figura 64 – Análise de transmissão do pacote 0

O histograma de densidade para todas as mensagens *Hello* enviadas na rede é apresentado na Figura 65. A maioria das mensagens é enviada em torno de 20 segundos, conforme pré-definido.

Adicionalmente, são exibidos, no eixo horizontal secundário, os intervalos de tempo de envio dos pacotes oriundos do nó fonte NTC1. As pequenas variações encontradas para os intervalos de tempo definidos para as mensagens *Hello* (20

segundos) e de dados (30 segundos) são em decorrência do tempo de espera das mensagens na fila de envio.

Figura 65 – Histograma de densidade das mensagens *Hello* – Rede A

Para uma análise da capacidade de auto-organização da rede, o gráfico da Figura 66 mostra os fluxos estabelecidos pela rede durante o período de coleta de dados vinculado às falhas dos nós SN2 e SN1. São exibidas as três rotas estabelecidas na rede em função do tempo.

Para tal, é mostrado o rastreamento da mensagem de dados por nó da rede durante a execução de cada rota. Ou seja, para cada pacote de mensagem difundido na rede são registrados os nós por onde a mensagem passou até chegar ao destino.

É observado que, após 550 segundos, a rede é restabelecida. Assim os nós SN2 e SN1 voltam a fazer parte da comunicação da rede e são inseridos novamente na tabela de seus vizinhos.

A partir de então, o fluxo de dados pela rota ótima é formado novamente, uma vez que todos os nós pertencentes a essa rota foram restituídos à rede.

Figura 66 – Fluxo de mensagens de dados enviadas – Rede A

Em detrimento da falha de comunicação dos nós SN2 e SN1, os fluxos dos pacotes 6 e 11 são interrompidos e a rede deve adotar um mecanismo para restabelecer o fluxo desses pacotes até o nó LP1. Tal mecanismo é centrado no método de tentativas de reenvio do pacote.

A avaliação do número de tentativas de reenvio da mensagem em detrimento da perda de comunicação dos nós SN2 e SN1 na rede, durante uma rota, é mostrada nas Figuras 67 e 68.

O critério de reenvio da mensagem ao nó falho deve satisfazer ao fluxograma lógico apresentado na Figura 38. Ainda, cada tentativa de reenvio da mensagem deve acontecer após um intervalo de tempo determinado pelo temporizador, apresentado na Tabela 7.

O tempo capturado para as tentativas de retransmissão dos pacotes 6 e 11 é abordado, individualmente, em cada um dos gráficos. O estudo da diferença entre esses intervalos de tempo é apresentado na Tabela 13.

É observado que o tempo entre cada tentativa de reenvio está de acordo com o estipulado pelo temporizador definido no protocolo.

Figura 67 – Análise do reenvio de mensagens – pacote 6 – Rede A

Figura 68 - Análise do reenvio de mensagens – pacote 11 – Rede A

Tabela 13 – Tempo entre tentativas de reenvio – Rede A

Nó Falho	Nó Emissor	Nó Receptor	Nº Tentativa	Tempo	Δt entre tentativas
SN2	0X9F2	0X872	----	245,057s	----
SN2	0X9F2	0X872	1	245,560s	0,503s
SN2	0X9F2	0X872	2	247,263s	1,703s
SN2	0X9F2	0X872	3	253,763s	6,5s
SN1	0X983	0X9F2	----	394,912s	----
SN1	0X983	0X9F2	1	395,412s	0,5s
SN1	0X983	0X9F2	2	397,112s	1,7s
SN1	0X983	0X9F2	3	403,613s	6,493s

O rastreamento das mensagens ACK enviadas por nó da rede em resposta às mensagens de dados enviadas na Figura 66, é exibido a seguir. Para cada mensagem de dados recebida pelos nós, uma mensagem de confirmação é enviada.

Figura 69 – Fluxo de mensagens ACK enviadas – Rede A

O tempo estimado para que cada mensagem alcance o destino é abordado na Figura 70. A análise mostra que o atraso para as mensagens com números de pacotes 6 e 11 é superior aos demais. Isso se deve ao tempo adicional necessário para reorganização da rede devido à perda de comunicação com os nós SN2 e SN1.

A Tabela 14 aborda um resumo do comportamento do protocolo para a rede A.

Tabela 14 - Resumo do comportamento do protocolo SMAC – Rede A

Fluxo	Nº Saltos	Nº Msg de dados sem falhas	Nº Msg de dados com falhas	Δt até destino sem falha (T_{SF})	Δt até destino com falha (T_{CF})	Nº Tentativa reenvio	Diferença ($T_{CF} - T_{SF}$)
Rota 1	3	11	14	0,368s	9,124s	3 (0x9F2 → 0x872)	8,756s
Rota 2	3	6	9	0,423s	9,123s	3 (0x983 → 0x9F2)	8,70s
Rota 3	3	5	5	0,405s	---	---	---

É observado na Tabela 14 que a diferença entre o tempo de envio da mensagem até o destino com falha e sem falha na rede é, praticamente, o tempo que a rede necessita para se auto-organizar (8,7 segundos).

Figura 70 – Tempo das mensagens até o destino – Rede A

A medição da qualidade do *link* foi executada para cada mensagem de dados recebida. Os valores mínimo e máximo LQI (-100dbm e -15dbm) são associados com a menor e a maior qualidade do *link*, respectivamente. A caracterização da intensidade das mensagens recebidas pode ser vista na Figura 71.

A análise do desempenho da qualidade do *link* mostrou que os níveis máximos e mínimos LQI, durante a execução de cada rota, tenderam a uma variação muito pequena. Ou seja, os valores LQI capturados para uma mesma rota são recorrentes. Isso porque cada nó da rede mantém com um determinado nó vizinho um nível de qualidade do *link* praticamente constante. Tal fato implica em uma tendência comportamental cíclica da qualidade do *link* medida durante as rotas 1, 2 e 3, conforme visto na Figura 71.

Ainda, o nível da qualidade do *link* apresentado é considerado entre o patamar bom (-60dbm) e muito bom (-40dbm), de acordo com as recomendações da *Freescle* [106]. Isto porque o cenário de desenvolvimento não está submetido às interferências que por ventura poderiam prejudicar a comunicação na rede.

No entanto, em zonas relacionadas a parques industriais e similares, o nível de emissão de interferências é muito grande. Isto pode gerar perdas na qualidade do *link*, bem como prejudicar a taxa de entrega de pacotes ao destino.

Figura 71 – Qualidade de *link* – Rede A

10.1.1 Verificação do critério *Drop*

Para a verificação do critério *Drop*, descrito na Seção 8.3, a rede A foi submetida a novos ensaios experimentais. O princípio adotado para avaliar este critério é descrito na Tabela 15. O rádio dos nós SN1 e NTC2 é desligado visando deixar o nó LP2 sem vizinhos em potencial para o repasse do pacote.

Tabela 15 – Falhas para verificação do critério *Drop*

Falha introduzida	Nó	MAC ID
Rádio desligado	SN1	0X9F2
Rádio desligado	NTC2	0X7FD

A Tabela 16 apresenta os fluxos das mensagens para cada rota estabelecida para verificação do critério *Drop*. Inicialmente, a rota 1 (ótima) é estabelecida entre os nós

fonte NTC1 e LP1. Posteriormente, o nó SN1 tem seu rádio desligado e uma segunda rota é estabelecida, a rota 2.

Por último, o nó NTC2 é removido da rede e o critério *Drop* é estabelecido. Isso porque o nó intermediário LP2, responsável pelo repasse do pacote, é o nó mais próximo do nó de destino dentre todos os seus vizinhos. No entanto, o nó de destino não pertence à sua área de alcance de rádio. Desta forma, as mensagens de dados recebidas pelo nó LP2 durante a ausência de seus vizinhos SN1 e NTC2, serão descartadas. Essa situação é abordada na Figura 72.

Tabela 16 – Fluxo de mensagens na rede A - Critério *Drop*

Posição do nó na rede	Fluxo ÓTIMO Rota 1	Fluxo Alternativo Rota 2
Origem	NTC1	NTC1
Intermediário	SN1	LP2
Intermediário	SN2	NTC2
Destino	LP1	LP1

Figura 72 – Verificação do critério *Drop*

Os pacotes 7, 8, 9 e 10 enviados pelo nó fonte NTC1(0x983) ao nó LP2 (0x7F8) são perdidos em decorrência do nó LP2 ser um máximo local, situação abordada na

Seção 5.5.1.3. Após 360 segundos, a rede é restabelecida e o fluxo de mensagens por meio da rota ótima é restituído.

Adicionalmente, o rastreamento das mensagens ACK enviadas em resposta às mensagens de dados enviadas na Figura 72 é apresentado a seguir.

Figura 73 – Fluxo de mensagens ACK enviadas – Critério Drop

10.1.2 Redes A com três fluxos

Para uma análise do desempenho do protocolo SMAC com roteamento em situações com mais de um fluxo de dados, a rede A foi submetida a três fluxos com diferentes origens para o mesmo destino (LP1). A Tabela 17 apresenta a origem e o destino para cada um dos fluxos com os respectivos nós intermediários ótimos.

Tabela 17 – Definição dos fluxos de mensagens até o destino

Posição do nó na rede	Fluxo 1	Fluxo 2	Fluxo 3
Origem	NTC1	NTC2	LP2
Intermediário	SN1	---	NTC2
Intermediário	SN2	---	---
Destino	LP1	LP1	LP1

Tal medida visa proporcionar um tráfego maior de pacotes na rede e avaliar o comportamento do protocolo na entrega de mensagens ao destino, bem como na capacidade de auto-organização da rede mediante a uma falha.

A transmissão de mensagens de dados para o destino, em cada fluxo, é periódica. Os intervalos de tempos de repetição para cada um dos fluxos determinados acima estão descritos na Tabela 18.

Tabela 18 – Ciclo de repetição de cada Fluxo

Fluxo	Intervalo de tempo	Número de saltos até o destino
1	6s	3
2	1s	1
3	5s	2

A Figura 74 apresenta o rastreamento das mensagens de dados enviadas na rede. Apesar do aumento no tráfego de mensagens, o protocolo proposto mostrou-se eficiente na entrega de pacotes ao destino para cada fluxo e preservou a capacidade de manutenção da rede mediante a quebra de um *link*.

Tal capacidade é verificada por meio da interrupção da rota ótima estabelecida pelo Fluxo 1. Após 120 segundos, o nó SN2 (0X872) é removido da rede e o Fluxo 1 é interrompido. O nó NTC2 (0X7FD) é o segundo nó vizinho ótimo ao nó SN1 (0X9F2) para este fluxo, e é o nó escolhido para substituir o nó falho SN2. Desta forma, a rede se auto recupera e o Fluxo 1 é restabelecido.

Figura 74 – Fluxo de mensagens de dados enviadas por fluxo – Rede A

O rastreamento das mensagens de confirmação (ACK) para cada mensagem de dados enviadas na Figura 74 é exibido a seguir.

Figura 75 – Fluxo de mensagens ACK enviadas por fluxo – Rede A

A Figura 76 apresenta uma análise do tempo gasto, por fluxo, para que as mensagens sejam entregues ao nó destinatário LP1.

Figura 76 – Tempo das mensagens até o destino por fluxo – Rede A

Quanto maior o número de saltos até o destino, maior o atraso na comunicação fim a fim. Para uma melhor visualização, a figura a seguir exibe a estimativa do atraso na comunicação fim a fim para os últimos pacotes da Figura 76. Os Fluxos 1 e 2 possuem respectivamente, o maior e o menor atraso na comunicação fim a fim.

Figura 77 – Visão detalhada dos atrasos – Rede A

A qualidade do *link* das mensagens de dados recebidas pelos nós da rede, para cada fluxo, é mostrada na Figura 78. O estudo mostra que, durante a execução de uma mesma rota, a qualidade do *link* estabelecida entre dois nós apresentou pouca variação.

Figura 78 – Qualidade de *link* por fluxo – Rede A

10.2 Rede de sensores sem fio - B

A segunda rede proposta neste trabalho é apresentada na Figura 79. Trata-se de uma rede também formada pelos nós extremos NTC1 e LP1 e pelos nós intermediários SN1, SN2, NTC2 e LP2.

Figura 79 – Rede de sensores sem fio - B

Semelhante à rede A, o fluxo de mensagens foi estabelecido entre o nó de origem NTC1 e o destinatário LP1. O tráfego de mensagens nesta rede é formado pelo envio periódico dos quadros de mensagens *Hello* e dados; e correspondem aos intervalos de tempo pré-definidos na Tabela 10.

O que individualiza a rede A da rede B é o número de saltos da mensagem até o destino. A rota ótima, estabelecida entre o nó NTC1 e LP1 para a rede B, possui apenas dois saltos até o destino, enquanto a rota ótima para rede A possui três saltos.

A Tabela 19 apresenta uma descrição das falhas introduzidas na rede B. O rádio dos nós SN1 e LP2 é desligado durante a coleta de dados para avaliar o desempenho do protocolo de roteamento geográfico inserido no SMAC.

Tabela 19 - Falhas introduzidas na Rede B

Falha introduzida	Nó	Tempo de Ausência do Nó
Rádio desligado	SN1	336.258s
Rádio desligado	LP2	173.087s

Os fluxos de mensagens entre os nós de origem NTC1 e de destino LP1 são exibidos na Tabela 20. São estabelecidas três rotas pela rede.

Tabela 20 - Fluxo de mensagens na Rede B

Posição do nó na rede	Fluxo ÓTIMO Rota 1	Fluxo Alternativo Rota 2	Fluxo Alternativo Rota 3
Origem	NTC1	NTC1	NTC1
Intermediário	SN1 *	LP2 *	NTC2
Intermediário	---	SN2	SN2
Destino	LP1	LP1	LP1

* Nós sujeitos à falha

Inicialmente, a rota 1, rota ótima, é traçada entre o nó fonte NTC1 e o nó de destino LP1. Posteriormente, o rádio do nó SN1 é desligado e a rota ótima é interrompida e uma segunda rota é traçada, rota 2. Por último, uma segunda falha é introduzida na rede e o funcionamento do rádio do nó LP2 é interrompido. Assim, uma terceira rota alternativa é estabelecida entre nó NTC1 e LP1, a rota 3.

As representações gráficas do desempenho do protocolo, durante o intervalo de tempo em que as condições de falhas são introduzidas na rede, são exibidas a seguir.

A Figura 80 apresenta uma análise quantitativa das mensagens enviadas na rede. São enviadas ao todo 22 mensagens de dados entre o nó NTC1 e LP1. No entanto, devido ao número de saltos necessários até o destino (2 ou 3 saltos), o número completo de mensagens de dados enviadas é 55, em condições de pleno funcionamento dos nós.

Uma vez que a rede é submetida a duas falhas, é necessário o reenvio de mensagens para reorganização da rede e restabelecimento do fluxo de mensagens até o destino. Assim, é necessário o reenvio de 6 mensagens, no mínimo, para a reorganização da rede, devido às falhas introduzidas, contabilizando, ao todo, 61 mensagens de dados.

O número de mensagens *Hello* enviadas pelos nós SN1 e LP2 é menor devido ao instante de perda de comunicação, registrado na Tabela 19.

Figura 80 - Análise quantitativa das mensagens enviadas – Rede B

O histograma de densidade das mensagens *Hello* enviadas na rede em conjunto com os pacotes transmitidos pelo nó fonte NTC1 são exibidos na figura a seguir. As mensagens de dados e de *Hello* emitidas na rede estão em torno, respectivamente, de 30 e 20 segundos, conforme esperado.

Figura 81 - Histograma de densidade das mensagens *Hello* – Rede B

A Figura 82 mostra os fluxos estabelecidos pela rede durante o período de coleta de dados vinculado às falhas dos nós SN1 e LP2. Para tal, é mostrado o rastreamento

das mensagens de dados por nó da rede durante a execução de cada rota. Após 550 segundos, a rede é restabelecida e a rota com menor número de saltos até o destino é constituída novamente.

Figura 82 - Fluxo de mensagens de dados enviadas– Rede B

Analisando o gráfico da Figura 82 os fluxos dos pacotes 6 e 11 são interrompidos em torno de 250 e 400 segundos, respectivamente. Desta forma, o processo para reestruturação dos fluxos é executado pelo protocolo visando encontrar uma rota alternativa até o nó de destino. Esse processo é realizado por meio de tentativas de reenvio da mensagem ao nó falho.

É observado que após número de tentativas de restabelecimento da comunicação com os nós SN1 (0x9F2) e LP2 (0x7F8) ter excedido o valor máximo (3), estes nós são substituídos na rede para garantir a entrega dos pacotes 6 e 11.

Inicialmente o nó SN1 é substituído pelo nó LP2, e posteriormente, o nó LP2 é substituído pelo nó NTC2 (0x7FD). Essas novas rotas possuem um maior número de saltos até o destino.

A avaliação do número de tentativas de reenvio da mensagem para os pacotes 6 e 11 é mostrada nas Figuras 83 e 84.

Figura 83 - Análise do reenvio de mensagens – pacote 6 – Rede B

Figura 84 - Análise do reenvio de mensagens – pacote 11 – Rede B

A diferença entre cada tentativa de retransmissão da mensagem é abordada na Tabela 21. Os tempos coletados entre cada tentativa de reenvio da mensagem estão de acordo com os tempos pré-definidos pelo temporizador implementado no protocolo proposto.

Tabela 21 – Tempo entre tentativas de reenvio – Rede B

Nó Falho	Nó Emissor	Nó Receptor	Nº Tentativa	Tempo	Δt entre tentativas
SN1	0X983	0X9F2	----	250,035s	----
SN1	0X983	0X9F2	1	250,536s	0,501s
SN1	0X983	0X9F2	2	252,238s	1,702s
SN1	0X983	0X9F2	3	258,736s	6,498s
LP2	0X983	0X7F8	----	400,053s	----
LP2	0X983	0X7F8	1	400,553s	0,5s
LP2	0X983	0X7F8	2	402,255s	1,702s
LP2	0X983	0X7F8	3	408,761s	6,506s

O rastreamento das mensagens ACK enviadas por nó da rede, durante a execução de cada rota, é exibido na figura a seguir.

Figura 85 - Fluxo de mensagens ACK enviadas – Rede B

A Tabela 22 aborda um resumo do comportamento do protocolo para a rede B. É observado que o número de saltos exigidos pela rota 1 até o destino é menor que as demais rotas. Logo, a velocidade de transmissão do pacote até o destino por esta rota será maior.

Novamente, a diferença entre os tempos da mensagem até o destino com falha e sem falha na rede, é aproximadamente o tempo estipulado para que a rede contorne o nó falho e estabeleça uma nova rota.

Tabela 22 - Resumo do comportamento do protocolo SMAC – Rede B

Fluxo	Nº Saltos	Nº Msg de dados sem falhas	Nº Msg de dados com falhas	Δt até destino sem falha (T_{SF})	Δt até destino com falha (T_{CF})	Nº Tentativa reenvio	Diferença ($T_{CF} - T_{SF}$)
Rota 1	2	11	14	0,153s	9,155s	3 (0x983 → 0x9F2)	9,002s
Rota 2	3	6	9	0,433s	9,058s	3 (0x983 → 0x7F8)	8,625s
Rota 3	3	5	5	0,356	---	---	---

Na Figura 86 é apresentada uma estimativa do tempo das mensagens de dados até o destino. São observados os picos de tempo para os pacotes de número 6 e 11. Esses picos são em detrimento das falhas dos nós SN1 e LP2.

Figura 86 - Tempo das mensagens até o destino – Rede B

A análise da medição da qualidade do *link* das mensagens recebidas pode ser vista na Figura 87. Os valores retornados para esta indicação são considerados apropriados. Isso porque os valores situados entre -60dbm e -40dbm, segundo indicação *Freescale* [116], correspondem a uma faixa de qualidade de *link* entre boa e muito boa.

Semelhante a rede A, os níveis da intensidade LQI capturada entre dois nós para uma mesma rota são muito próximos, ou seja, sofreram pouca variação. Isso exhibe um caráter cíclico das medidas LQI para cada uma das rotas efetuadas na rede.

Figura 87 – Qualidade de link – Rede B

Para finalizar a coleta de dados na rede B, um último teste foi realizado para avaliar o comportamento do protocolo mediante a falha de dois nós consecutivos em uma mesma rota. Essa situação é apresentada na Figura 88.

Os nós SN1 (0x9F2) e o nó LP2 (0x7F8) são removidos da rede praticamente juntos instantes antes de uma mensagem de dados ser enviada pelo nó NTC1 (0X983). Devido ao intervalo de tempo de reorganização da rede ser inferior ao tempo necessário para remover um nó por falta de *Hello*, os nós SN1 e LP2 ainda fazem parte da tabela de vizinhos do nó NTC1.

Desta forma, o nó NTC1 inicialmente tenta enviar a mensagem (pacote 3) para o nó SN1 (rota ótima). Como tal nó está ausente da comunicação da rede, o nó NTC1 remove o nó SN1 de sua tabela de vizinhos e aponta um segundo vizinho alternativo para esta rota.

O segundo nó escolhido é o nó LP2 (rota 2) que também está ausente da rede, mas ainda permanece na tabela de vizinhos do nó NTC1. Devido à falta de comunicação, o nó LP2 também é removido da tabela de vizinhos do nó NTC1.

Finalmente, um terceiro vizinho, NTC2 (0x7FD), é selecionado pelo protocolo de roteamento geográfico, e a rota 3 é estabelecida para manutenção do fluxo entre o nó fonte NTC1 e o nó de destino LP1.

É observado que após 225 segundos, a comunicação na rede é restabelecida e nós SN1 e LP2 voltam a transmitir mensagens de *Hello* para seus vizinhos. Assim, quando o nó fonte NTC1 enviar o próximo pacote para o nó LP1, a rota ótima será restituída.

Figura 88 – Falhas consecutivas - Rede B

Os quadros de confirmação enviados para cada uma das mensagens de dados enviadas na rede é exibido a seguir.

Figura 89 - Fluxo de mensagens ACK enviadas: Falhas consecutivas - Rede B

Em casos de falhas consecutivas, o tempo de reorganização da rede é acumulativo. Isso aumenta o tempo gasto para que a mensagem chegue até o destino.

O tempo de reorganização da rede em detrimento das falhas consecutivas dos nós SN1 e LP2 foi de 17.403 segundos. Este tempo de reorganização da rede é o dobro do tempo gasto para restabelecer a rede quando uma única falha acontece isoladamente.

Na Figura 90 é apresentada uma estimativa do tempo das mensagens de dados até o destino para este cenário.

Figura 90 - Tempo das mensagens até o destino: Falhas consecutivas – Rede B

É importante ressaltar que a remoção de um nó falho da tabela de vizinhos por falta de comunicação durante uma rota, garante uma agilidade na reestruturação da rede. O mesmo não aconteceria se, este nó falho, só fosse removido após a expiração do período de ausência de recebimento do quadro *Hello*.

Uma consideração a ser feita é que a natureza cíclica da comunicação na rede, orientada a períodos de envios, possui temporização flexível. Essa flexibilidade é possível por meio de ajustes dos parâmetros que configuram os intervalos de tempos para funcionamento do protocolo SMAC com roteamento geográfico.

O passo da fila da tabela de mensagens a serem enviadas, por exemplo, pode ser ajustado para valores menores do que o determinado neste trabalho (100ms). Isso permitirá um aumento da velocidade no processo de envio das mensagens, e conseqüente redução no atraso da comunicação fim a fim.

Porém, ainda não foi feito um estudo para valores inferiores a 100ms, em razão da dificuldade do *sniffer* em capturar e registrar o tráfego de mensagens nessas circunstâncias.

Em geral, os atrasos na comunicação fim a fim obtidos nos ensaios experimentais para as redes A e B, são considerados aceitáveis para o tempo da aplicação em sistemas de gerenciamento e monitoramento da iluminação pública.

Para sistemas que necessitam de um tempo de resposta menor, os parâmetros do protocolo proposto podem ser reajustados visando atender às exigências de tempo de execução de tarefas da aplicação.

11 CONCLUSÕES

O principal objetivo deste trabalho foi contribuir com o desenvolvimento de um protocolo para a camada de rede, baseado no algoritmo de roteamento geográfico para redes sem fio *mesh*. O protocolo SMAC foi utilizado para o desenvolvimento do trabalho.

O protocolo SMAC é bastante simples, se comparada a sistemas operacionais mais complexos, como o TinyOS. Além disso, ela possibilita a comunicação sem fio efetiva entre os nós da rede.

A inserção do protocolo de roteamento geográfico no SMAC, implementado por meio da menor distância Euclidiana, definiu o melhor caminho entre o nó fonte e nó destino, priorizando o menor número de saltos em uma comunicação fim a fim. Como o número de atrasos introduzidos por salto é reduzido, diminuindo a latência, a velocidade de transmissão das mensagens é maior.

Com relação à tolerância às falhas, o protocolo proposto demonstrou ser capaz de encontrar rotas alternativas na presença de nós roteadores falhos, assim, a rede se auto-organiza para manter seu pleno funcionamento e otimizar o tráfego de dados.

O critério adotado para confirmação de recebimento de dados a cada retransmissão de mensagem é mais eficiente do que as técnicas de confirmação fim a fim, uma vez que diminui a probabilidade de falha na transmissão entre dois nós.

A implementação de um protocolo da camada de rede permitiu ao SMAC a utilização da topologia de rede *mesh*, uma topologia de rede mais complexa do que a topologia estrela.

Devido à limitação de dispositivos para testes, a rede foi restringida a seis nós estáticos de comunicação sem fio. Para redes com maior número de nós, alguns pontos podem ser analisados para aumentar a confiabilidade da rede, como por exemplo: a restrição da quantidade de nós vizinhos por dispositivo e a otimização da quantidade de retransmissões, para uso mínimo de bateria.

Por fim, este trabalho irá colaborar com o projeto de pesquisa “Redes de Sensores Sem Fio de Aplicação Urbana”, do Laboratório de Automação Industrial da EESC-USP. Essa colaboração pode ser comprovada através do estudo da pilha protocolar SMAC e do método abordado para o desenvolvimento de uma camada de rede na mesma.

Além disso, o protocolo de rede implementado, baseado no algoritmo de roteamento geográfico, é altamente indicado para aplicações de sistemas de controle e monitoramento de variáveis, relacionadas à iluminação pública em áreas metropolitanas.

Como proposta de melhoria para este trabalho, a técnica aqui descrita pode ser integrada com algoritmos de roteamento mais complexos já avaliados em simulação [112]. Tais algoritmos eliminam o critério *drop* estabelecido neste trabalho. Caso o encaminhamento do pacote atinja um nó que seja máximo local, o pacote não será descartado. Para tanto, modificações no protocolo GPSR [68] são empregadas para garantir o fluxo de pacotes até o destino.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SILVA, I. D. **Sistema de Comunicação Móvel Domótico**. 2007. 63f. Dissertação (Bacharelado em Ciência da Computação) -Universidade Federal de Pelotas, Pelotas, 2007.
- [2] VISWANATHAN, M. (2006). *World wireless sensor markets*. In: International Congress for Wireless Sensors and Networks (ICWSN), Paris, France. Frost & Sullivan. *apud. Energy-Efficient Self-Organization for Wireless Sensor Networks*. Thomas Watteyne, PhD thesis, INSA Lyon, November 2008.
- [3] HART COMMUNICATION FOUNDATION. **WirelessHart Simple Reliable Secure**. Disponível em: <http://www.hartcomm2.org/hart_protocol/wireless_hart/wirelesshart_brochure.pdf>. Acesso em: 10 de jun. 2010.
- [4] ZIGBEE ALLIANCE. **ZigBee Specification**. Disponível em:<<http://zigbee.org/>>. Acesso em: 29 jun. 2009.
- [5] IEEE Std. 802.15.4. **Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (LR-WPANS)**. 2006.
- [6] RFC 5548 (2009). Routing Requirements for U-LLNs. Disponível em: <<http://datatracker.ietf.org/doc/rfc5548/>>. Acesso em: 15 out. 2010.
- [7] Draft-ietf-roll-rpl-11 (2010). RPL: IPv6 Routing Protocol for Low power and Lossy Networks. Disponível em: <<http://datatracker.ietf.org/doc/draft-ietf-roll-rpl/>>. Acesso em: 5 nov. 2010.
- [8] J-SIM. Disponível em: <<http://www.j-sim.org/>>. Acesso em: 3 março 2009.
- [9] OMNET++. Disponível em: <<http://freshmeat.net/projects/omnetpp/>> Acesso em: 8 maio 2009.
- [10] NS2. Disponível em: <www.isi.edu/nsnam/ns/>. Acesso em: 14 de março de 2009.
- [11] T S. DEVELOPERS. **Shox - a scalable ad hoc network simulator**. Disponível em: <<http://shox.sourceforge.net>>. Acesso em: 13 maio 2009.
- [12] SIVALINGAM, K. M.; SRIVASTAVA, M. B.; AGRAWAL, P. Low Power Link and Access Protocols for Wireless Multimedia Networks. In: PROCEEDINGS IEEE VEHICULAR TECHNOLOGY CONFERENCE VTC'97, 47, 1997, Phoenix.
- [13] STEMM, M.; GAUTHIER, P.; HARADA, D.; KATZ, R. Reducing Power Consumption of Network Interfaces in Hand-Held Devices. In: PROCEEDINGS 3RD INTL. WORKSHOP ON MOBILE MULTIMEDIA COMMUNICATIONS, 1996, Princeton.
- [14] MANJESHWAR, A.; AGRAWAL, D. **TEEN: A routing protocol for enhanced efficiency in wireless sensor networks**. California, 2001.

- [15] LINDSEY, S.; RAGHAVENDRA, C. R. **PEGASIS: Power-Efficient Gathering in Sensor Information Systems**. Los Angeles, 2000.
- [16] HEINZELMAN, W. B.; CHANDRASKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless networks. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 33., 2000, Cambridge.
- [17] SEMICONDUCTOR, FREESCALE^{TM}. **Simple Media Access Controller (SMAC) - user's Guide, Freescale^{TM} Semiconductor**. Document Number: SMACRM, Rev. 1.5, 2008.
- [18] WATTEYNE, T.; MEHTA, A.; PISTER, K. Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense. Sixth ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), Tenerife, Canary Islands, Spain, 2009.
- [19] FAÇANHA, T. S. **Rede de sensores sem fio: uma abordagem para detecção de falhas em sistemas elétricos**. 2007. 62f. Dissertação (Monografia para Título de Tecnólogo em Telemática) - Centro Federal de Educação Tecnológica do Ceará, Ceará, 2007.
- [20] SILVA, R. C. **Redes de sensores sem fio**. 2008. 52f. Dissertação (Conclusão de curso de Sistemas de Informação) - Universidade Estadual de Montes Claros, Montes Claros, 2008.
- [21] AKYILDIZ, I. F.; WANG, X.; WANG, W. Wireless mesh networks: a survey, Computer Networks. v. 47, p. 445-487, March, 2005.
- [22] VIEIRA, M. A. M.; COELHO JÚNIOR., C. N.; SILVA JÚNIOR., D. C.; MATA, J. M. **Survey on wireless sensor network devices**. Belo Horizonte, 2003.
- [23] MOREIRA, M. D. D. **Introdução às Redes de Sensores Sem Fio**. Disponível em: <http://www.gta.ufrj.br/grad/06_1/rssf/>. Acesso em: 10 set. 2009.
- [24] ULMER, C. **Wireless Sensor Network**. Disponível em: <http://users.ece.gatech.edu/~grimace/research/sensorsimii/wireless_sensor_network.ppt>. Acesso em: 15 out. 2009.
- [25] AKYILDIZ, I. F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. A survey on sensor networks. **IEEE Communications Magazine**, v. 40, p. 102-114, August, 2002.
- [26] LOPES, A. M. A. **Processamento adaptativo de consultas em redes de sensores sem fio**. 2006. 135 f. Dissertação (Mestrado em Informática aplicada) - Universidade de Fortaleza, Fortaleza, 2006.
- [27] PISTER, K.; DOHERTY, L. TSMP: Time Synchronized Mesh Protocol. In: PARALLEL AND DISTRIBUTED COMPUTING AND SYSTEMS (PDCS). Orlando, Florida, USA, November, 2008.

- [28] VANNINEN, T., TUOMIVAARA, H., HUOVINEN, J. A demonstration of frequency hopping ad hoc and sensor network synchronization method on warp boards. In: PROCEEDINGS OF THE THIRD ACM INTERNATIONAL WORKSHOP ON WIRELESS NETWORK TESTBEDS, EXPERIMENTAL EVALUATION AND CHARACTERIZATION. San Francisco, California, USA, September, 2008.
- [29] JIANG, X., TANEJA, J., ORTIZ, J., TAVAKOLI, A., DUTTA, P., JEONG, J., CULLER, D., LEVIS, P., and SHENKER, S. An architecture for energy management in wireless sensor networks. SIGBED Rev. 4, 3, July, 2007.
- [30] RUIZ, L. B.; NOGUEIRA, J. M.; LOUREIRO, A. **Sensor network management**. Handbook of sensor networks: compact wireless and wired sensing systems. Edited by Mohammad Ilyas and Imad Mahgoub, Washington: CRC Press, 2004, p. 57 – 84.
- [31] MADDEN, S.; FRANKLIN, M.; HELLERSTEIN, J. TinyDB: an acquisitional query processing system for sensor networks. **ACM Transactions on Database Systems - TODS**, v. 30, n.1, 2005, p. 122-173.
- [32] MADDEN, S.; FRANKLIN, M. J.; HELLERSTEIN, J.; HONG, W. TAG: a Tiny AGgregation service for ad-hoc sensor networks. In: SYMPOSIUM ON OPERATING SYSTEM DESIGN AND IMPLEMENTATION – OSDI, 5., 2002, Boston, p. 171–182.
- [33] CONSIDINE, J.; LI, F., KOLLIOS, G.; BYERS, J. Approximate aggregation techniques for sensor databases. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING – ICDE, 2004, Boston, p. 449-460.
- [34] RODRIGUES, R. O que é uma Rede de sensores sem Fio. **Revista Controle e Instrumentação**, nº144, p. 98-99, 2009.
- [35] TAVARES, P. L. **Redes de sensores sem fio**. Disponível em: <http://www.gta.ufrj.br/grad/02_2/Redes%20de%20sensores/Redes%20de%20Sensores%20Sem-fio.htm>. Acesso em: 20 out. 2008
- [36] SCHWARTZ, S. M. **Frequency Hopping Spread Spectrum (FHSS) vs. Direct Sequence Spread Spectrum (DSSS)**. Disponível em: <<http://www.sorin-schwartz.com/white.htm>>. Acesso em: 15 nov. 2009.
- [37] RODRIGUES, V. J. S. **Computação Móvel e Ubíqua**. Goiás, 2006. p. 1- 4.
- [38] COZZO, R. Tecnologia Industrial Wireless, Conceitos Básicos. **Revista Mecatrônica Atual**, n. 38, p.18-25, 2008.
- [39] FARIAS, P. C. B. **Redes-básico**. Disponível em: <<http://www.juliobattisti.com.br/tutoriais/paulocfarias/redesbasico006.asp>>. Acesso em: 13 agosto 2009.
- [40] OLIVEIRA, F. T. X.; FILHO, H. B. **Rádio Spread Spectrum**. Disponível em: <<http://www.planetarium.com.br/planetarium/noticias/2004/11/1099337171>>. Acesso em: 6 out. 2009.

- [41] ALBUQUERQUE, P. U. B.; ALEXANDRIA, A. R. **Redes industriais**: aplicações em sistemas digitais de controle distribuído: protocolos industriais, aplicações SCADA. 2ed. São Paulo: Ensino Profissional, 2009, 258 p.
- [42] SILVA, I. M.; OLIVEIRA, L. A. H. G. Aplicações de redes de sensores sem fio em monitoramento de poços petrolíferos terrestres. In: RIO OIL & GAS EXPO AND CONFERENCE. 2008, Rio de Janeiro, p. 1-8.
- [43] TANENBAUM, A. S. **Redes de Computadores**. 4 ed. Rio de Janeiro: Elsevier, 2003, 945 p.
- [44] ROHDE, G. F.; BEZERRA, E. A. **Consumo eficiente de energia em RSSF utilizando um protocolo MAC**. RS, 2004, p. 1-14.
- [45] HAYKIN, S.; MOHER, M. **Sistemas Modernos de Comunicação Wireless**. Porto Alegre: Bookman, 2008, 580 p.
- [46] WOO, A.; CULLER, D. (2001). A transmission control scheme for media access in sensor networks. In: THE SEVENTH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING. 2001, Rome, p. 221-235.
- [47] SHIH, E.; CHO, S. H.; ICKES, N.; MIN, R.; SINHA, A.; WANG, A.; CHANDRAKASAN, A. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In: THE SEVENTH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING. 2001, Rome.
- [48] FROST, C. S.; FRIEDMAN, E. M. **Computer simulation model for a CSMA/CD local area network utilizing a multi – rate voice coding technique for load control**. Kansas, 1985, 211 p.
- [49] INTERNET ENGINEERING TASK FORCE (IETF). **Mobile ad-hoc networks (MANET) working group**. Disponível em: <<http://www.ietf.org/html.charters/manet-charter.html>>. Acesso em: 10 out. 2009.
- [50] SOHRABI, K.; GAO, J.; AILAWADHI, V.; POTTIE, G. J. Protocols for self-organization of a wireless sensor network. **IEEE Personal Communications**. p. 16-27, 2000.
- [51] LEE, A.; WARD, P. A. S. **A study algorithms in wireless mesh networks**. Waterloo, 2004.
- [52] CARNEIRO, D.; PRUDÊNCIO, P.; SOUSA, D.; MARROCOS, T.; PINHEIRO, P.; CAVALCANTE, D.; BRAQUEHAIS, R.; FERNANDEZ, M. Um estudo de caso para economia de energia em ambientes wireless através de implementações em hardware. **Revista Controle e Instrumentação**, n. 144, p. 48-55, 2009.
- [53] CUGNASCA, C. E. Redes de sensores sem fio. In: CONGRESSO BRASILEIRO DE AGRICULTURA DE PRECISÃO. 2008, Piracicaba.

- [54] JOHNSON, D.; MALTZ, D. Dynamic Source Routing in Ad Hoc Wireless Networks. In: COMPUTER COMMUNICATION REVIEW – PROCEEDINGS OF SIGCOMM, p. 153-181, 1996.
- [55] PERKINS, C.; ROYER, E. Ad-hoc On-Demand Distance Vector Routing. In: IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS. 1999, New Orleans, p. 90-100.
- [56] DRAVES, R.; PADHYE, P. AND ZILL, B. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. In: SIGCOMM, August 2004.
- [57] PERKINS, C.; BHAGWAT, P. Higly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for Mobile Computers. **ACM SIGCOMM**. v. 24, n. 4, p. 234-244, 1994.
- [58] CLAUSEN, T.; JACQUET, P. **Optimized Link State Routing Protocol (OLSR)**. Disponível em: <<http://www.ietf.org/rfc/rfc3626.txt>>. Acesso em: 25 set. 2009.
- [59] MURTHY, S.; GACIA-LUNA-ACEVES, J.J. An efficient routing protocol for wireless networks. **Mobile networks and Applications, Special Issue Routing in Mobile Communication Networks**. v. 1, n. 2, p.183- 197, 1996.
- [60] WATTEYNE, T.; BARTHEL, D.; DOHLER, M. and AUGÉ-BLUM, I. Using Virtual Coordinates for Wireless Sensor Networks: Proof-of-Concept Experimentation. In: 2nd WORKSHOP ON EXPERIMENTAL EVALUATION AND DEPLOYMENT EXPERIENCES ON VEHICULAR NETWORKS (WEEDEV), Washington D.C., USA, April, 2009.
- [61] WATTEYNE, T.; AUGÉ-BLUM, I.; DOHLER, M.; UBEDA, S. and BARTHEL, D. Centroid Virtual Coordinates - A Novel Near-Shortest Path Routing Paradigm. In: INTERNATIONAL JOURNAL OF COMPUTER AND TELECOMMUNICATIONS NETWORKING, Elsevier, Special Issue on Autonomic and Self-Organising Systems, v. 53, p. 1697-1711, July, 2009.
- [62] Draft-ietf-roll-routing-metrics-08 (2010). Routing Metrics used for Path Calculation in Low Power and Lossy Networks. Disponível em: <<http://datatracker.ietf.org/doc/draft-ietf-roll-routing-metrics/>>. Acesso em: 30 agost. 2010.
- [63] HUI, J. W. and CULLER, D. E. Extending IP to Low-Power, Wireless Personal Area Networks. IEEE Internet Computing 12, p. 37-45, July, 2008.
- [64] MAUVE, M.; WIDMER, J. A Survey on Position-Based Routing in Mobile Ad-Hoc Networks. **Network, IEEE**, v. 15, p. 30-39, nov./dez, 2001.
- [65] SILVA, D. R. T. **Estudo de Algoritmos de Roteamento em “Ad-Hoc Multi-Hop Mesh Networks” para Monitoramento de Sistema de Iluminação Pública**. 2008. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade de São Paulo. São Carlos, 2008.

- [66] ELHAFSI, E. H.; E SIMPLOT-RYL, D. **Flattening the gap between source-destination paths in energy greedy georouting in wireless sensor network**. France, 2007. p. 56-65.
- [67] KRANAKIS, E.; SINGH, H.; URRUTIA, J. Compass Routing on geometric networks. In: PROC CANADIAN CONFERENCE ON COMPUTATIONAL GEOMETRY, 11., 1999, Vancouver, p. 51-54.
- [68] KARP, B.; KUNG, H. T. GPSR: Greedy perimeter stateless routing for wireless networks. In: ACM INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING (MOBICOM), 2000, Boston, p. 243-254.
- [69] DENARDIN, W. G.; BARRIQUELO, C. H.; CAMPOS, A.; PRADO, R. N. Impacto das camadas inferiores para o desenvolvimento de protocolos de roteamento para redes de sensores sem fio em grande escala. XII Congresso Brasileiro de Automática, Bonito, setembro, 2010.
- [70] MENDES, D. R. **Redes de Computadores: Teoria e Prática**. São Paulo: Novatec, 2007, 384 p.
- [71] SANKARASUBRAMANIAM, Y.; ÖZGÜR, B. A.; AKYILDIZ, I. F. Esrt: event-to-sink reliable transport in wireless sensor networks. In: PROCEEDINGS OF THE 4TH ACM INTERNATIONAL SYMPOSIUM ON MOBILE AD HOC NETWORKING AND COMPUTING. 2003, Annapolis, p. 177-188.
- [72] STANN, F.; HEIDEMANN, J. Rmst: Reliable data transport in sensor networks. In: PROCEEDINGS OF THE FIRST INTERNATIONAL WORKSHOP ON SENSOR NET PROTOCOLS AND APPLICATIONS. 2003, Alaska, p. 102-112.
- [73] VUNDA, F. J. Protocolo de Comunicação IEEE 802.15.4 e ZigBee. Porto Alegre, 2005. 87 p.
- [74] IEEE STANDARD 802.2. **Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements -Part 2: Logical Link Control**. 1998.
- [75] GUTIERREZ, J. On the use of IEEE 802.15.4 to enable wireless sensor networks in building automation. In: IEEE INTERNATIONAL SYMPOSIUM ON PERSONAL, INDOOR AND MOBILE RADIO COMMUNICATIONS. Barcelona, 2004. p.1865-1869.
- [76] HOWITT, I.; GUTIERREZ, J. A. IEEE 802.15.4 Low Rate - Wireless Personal Area Network Coexistence Issues. **Wireless Communications and Networking**. v.3, p. 1481-1486, 2003.
- [77] CALLAWAY, E.; GORDAY, P.; HESTER, L.; GUTIERREZ, J. A.; NAEVE, M.; HEILE, B.; BAHL, V. Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate Wireless Personal Area Networks. **IEEE Communications**. v. 40, n.8. p. 70-77, 2002.

- [78] SILVA, I. M. **Análise de desempenho de sistemas de comunicação sem fio para monitoramento de unidade de produção de poços petrolíferos terrestres**. 2008. 70f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio Grande do Norte, Natal, 2008.
- [79] SANTOS, S. T. **Redes de Sensores Sem fio em Monitoramento e Controle**. 2007. 87f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.
- [80] JENNIC TECHNOLOGY FOR A CHANGING WORLD. **Calculating 802.15.4 Data Rates**. p.1-8, 2006.
- [81] ZHENG, J., LEE, M. J. **A comprehensive performance study of IEEE 802.15.4**. New York, 2004. p.1-14.
- [82] IEEE STANDARD FOR PART 15.4. **Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANS)**. 2003.
- [83] KOUBAA, A.; ALVES, M.; NEFZI, B.; SONG, Y. **Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks**. In: PROC. WORKSHOP OF REAL-TIME NETWORKS, 2006, France, 6 p.
- [84] CAMPOS, C. F. P. S. **Estudo e especificação de um sistema de instrumentação para unidades de elevação de petróleo utilizando tecnologia sem fio**. 2006. 77f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio Grande do Norte, Natal, 2006.
- [85] NOVA, F. S. V. **Redes de sensores em malha sem fio: um estudo de caso de automação residencial usando a tecnologia ZIGBEE/IEEE 802.15.4**. 2009. 96f. Dissertação (Bacharelado em Ciência da Computação) - Universidade Federal do Pará, Belém, 2009.
- [86] PISTER, K. S. J.; DOHERTY, L. **TSMP: TIME SYNCHRONIZED MESH PROTOCOL**. Proceedings of the IASTED International Symposium Distributed Sensor Networks (DNS 2008). Orlando, USA, November, 2008.
- [87] IEEE 802.15^{TM} WPAN. **4e Grupo de Trabalho (TG4e)**. Disponível em: <<http://translate.google.com.br/translate?hl=pt-BR&langpair=en%7Cpt&u=http://www.ieee802.org/15/pub/TG4e.html>> Acesso em: 15 nov. 2010.
- [88] IEEE 802.15^{TM} WPAN. **4g Grupo de Trabalho (TG4g)**. Disponível em: <<http://translate.google.com.br/translate?hl=pt-BR&langpair=en%7Cpt&u=http://www.ieee802.org/15/pub/TG4g.html>> Acesso em: 10 nov. 2010.
- [89] KOJIMA, F.; HARADA, H. **Study on Multipath Characteristics for IEEE 802.15.4g SUN Applications in the Frequency Band Used in Japan**. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS WORKSHOPS (ICC), Japan, May, 2010.

[90] ISA- THE INSTRUMENTATION, SYSTEMS, AND AUTOMATION SOCIETY. **ISA100.11A RELEASE 1: An Update on the First Wireless Standard Emerging from the Industry for the Industry.** 2007.

[91] BARRIQUELO, C. H.; GARCIA, J. M.; CORRÊA, C.; MENEZES, C. V.; CAMPOS, A.; PRADO, R. N. **Sistemas inteligentes baseados em ZigBee para iluminação pública com lâmpadas LEDS.** Santa Maria, 2008.

[92] SILVA, H. T. M. **Projeto e implementação de um sensor magnético sem fio aplicado ao método Plunger Lift.** 2009. 49f. Dissertação (Relatório de Estágio Supervisionado para conclusão de curso em engenharia da computação) - Universidade Federal do Rio Grande do Norte, Natal, 2009.

[93] JUNIOR, L. A. C. WirelessHart - Tecnologia Wireless aplicada a instrumentos de campo. **Revista Controle & Instrumentação**, n.144, p. 74 -78, 2009.

[94] LENNVALL, T.; SVENSSON, S.; HEKLAND, F. A comparison of WirelessHart and ZigBee for Industrial Applications. In: IEEE INTERNATIONAL WORKSHOP ON FACTORY COMMUNICATION SYSTEMS. 2008, Germany. p. 85-88.

[95] KIM, A. N.; HEKLAND, F.; PETERSEN, S.; DOYLE, P. When Hart goes Wireless: Understanding and Implementing the WirelessHart Standard. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION. 2008, Germany, p. 899-907.

[96] BIASI, M.; SNICKARS, C.; LANDERNAS, K.; ISAKSSON, A. J. (2008). Simulation of Process Control with WirelessHart Networks Subject to Packet Losses. In: IEEE INTERNATIONAL CONFERENCE AUTOMATION SCIENCE AND ENGINEERING. 2008, Bangalore, p. 548-553.

[97] PEKHTERYEV, G.; SAHINOGLU, Z.; ORLIK, P.; BHATTI, G. Image transmission over IEEE 802.15.4 and zigbee networks. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS – ISCAS, 2005, Japan, p. 3539- 3542

[98] BOUGHANMI, N.; SONG, Y. **Improvement of zigbee routing protocol including energy and delay constraints.** Loria, 2006, 4 p.

[99] NEFZI, B.; SONG, Y. Performance analysis and improvement of zigbee routing protocol. In: INTERNATIONAL CONFERENCE ON FIELD BUSES & NETWORKS IN INDUSTRIAL & EMBEDDED SYSTEMS, 7., 2007, France.

[100] ALMEIDA, F. B. (2009). **Padronização da comunicação sem fio em Ambientes industriais – ISA SP100.** 2009. 55f. Dissertação (Trabalho de Conclusão de Curso em especialização em automação industrial com ênfase em informática industrial, instrumentação, controle e otimização de processos contínuos) - Universidade Federal da Bahia, Salvador, 2009.

[101] ZHANG; X.; WEI, M.; WANG, P.; KIM, Y. Research and implementation of security mechanism in ISA100.11a networks. In: 9TH INTERNATIONAL CONFERENCE ON ELECTRONIC MEASUREMENT & INSTRUMENTS, China, August, 2009.

- [102] CHEN, F.; GERMAN, R. and DRESSLER, F. Towards IEEE 802.15.4e: A Study of Performance Aspects. Dept. of Computer Science, University of Erlangen, Germany, 2010.
- [103] SEMICONDUCTOR, FREESCALE™. **MC13224V Advanced ZigBee™ Compliant Platform-in-Package (PiP) for the 2.4 GHz IEEE® 802.15.4 Standard - user'sGuide, Freescale™ Semiconductor**. Document Number: MC1322x, Rev. 1.2., 2009.
- [104] INSTRUMENTS, TEXAS. **A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee™**. Document Number: SWRS036A , Rev. 1.03., 2009.
- [105] SEMICONDUCTOR, FREESCALE™. **MC1322x Simple Media Access Controller (SMAC) - user'sGuide, Freescale™ Semiconductor**. Document Number: 22XSMACRM, Rev. 1.4., 2009.
- [106] SEMICONDUCTOR, FREESCALE™. **MC1322x Advanced ZigBee™-Compliant SoC Platform for the 2.4 GHz IEEE® 802.15.4 Standard Reference Manual**. Document Number: MC1322xRM, Rev. 1.3., 2010.
- [107] SOHRABY, K; MONOLY, D.; ZNATI, T. Wireless Sensor Networks: Technology, Protocols and Applications, John Wiley & Sons, Inc. 2007.
- [108] STOJMENOVIC, I. Position-Based Routing in Ad Hoc Networks. **IEEE Communications Magazine**, v. 40, p. 128 - 134, 2002.
- [109] CHEN, H.; QIAN, D.; WU, W.; FANG, W. Link-Aware Geographic Routing in Wireless Sensor Networks. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND ENGINEERING. v.2, p. 937 - 942, 2009.
- [110] SEMICONDUCTOR, FREESCALE™. **Beekit Wireless Connectivity Toolkit - user's Guide, Freescale™ Semiconductor**. Document Number: BKWCTKUG, Rev. 2.1., 2009.
- [111] SYSTEMS, IAR. **IAR Embedded Workbench® for ARM**. Document Number: DS-EWARM-540, 2009
- [112] PANTONI, R. P.; BRANDAO, D. Analysis and Proposal of a Geocast Routing Algorithm Intended For Street Lighting System Based On Wireless Sensor Networks. In: INTERNATIONAL JOURNAL OF COMPUTER AND NETWORK SECURITY, v. 2, p. 65-70, 2010.