



ESTUDO E DESENVOLVIMENTO DE UMA PLATAFORMA DE COMUNICAÇÃO SEM FIO PARA REDES ELÉTRICAS INTELIGENTES

Juliano Freitas Caldeira

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

Orientador: Mauricio Aredes

Rio de Janeiro
Novembro de 2012

ESTUDO E DESENVOLVIMENTO DE UMA PLATAFORMA DE
COMUNICAÇÃO SEM FIO PARA REDES ELÉTRICAS INTELIGENTES

Juliano Freitas Caldeira

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

Examinada por:

Prof. Mauricio Aredes, Dr.-Ing.

Prof. Jorge Lopes de Souza Leão, Dr. Ing.

Prof. Marcelo Gonçalves Rubinstein, D. Sc.

RIO DE JANEIRO, RJ - BRASIL

NOVEMBRO DE 2012

Caldeira, Juliano Freitas

Estudo e desenvolvimento de uma plataforma de comunicação sem fio para redes elétricas inteligentes / Juliano Freitas Caldeira. – Rio de Janeiro: UFRJ/COPPE, 2012.

XII, 150 p. il.; 29,7 cm.

Orientador: Mauricio Aredes

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Elétrica, 2012.

Referências Bibliográficas: p. 123-128.

1. Redes sem fio. 2. Protocolo de Comunicação. 3. *Smart grid*. I. Aredes, Mauricio. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Durante nossas vidas, às vezes não percebemos a verdadeira importância das pessoas que nos cercam. Só nos damos conta do real valor delas quando nos encontramos em uma situação de crise. É nesse momento que as pessoas de valor fazem a diferença e nos mostram a verdadeira essência do que é na verdade ser humano.

Em especial, agradeço aos meus pais, Viviane Habib Freitas e Mauricio Lacerda Caldeira, por todo o suporte e dedicação que sempre conferiram a mim; à minha namorada e, muito provavelmente, futura esposa Aline Guidry, por me apoiar e estar, mesmo quando distante, sempre ao meu lado; e, ao maior amor da minha vida, Rafaela, minha filha.

Agradeço também ao meu professor e orientador Mauricio Aredes, por sempre me motivar, escutar, aconselhar e ensinar, principalmente a seguir minha intuição, em todos os momentos; ao Prof. Jorge Lopes de Souza Leão, pelo grande aprendizado que tive na convivência diária e em suas aulas e conversas sempre cultas e inspiradoras; ao Professor José Rezende, por ser um grande motivador, desde os tempos da minha graduação e atravessando todo o mestrado, e que culminaram por basear toda a linha de trabalhos que desenvolvi desde então e, ao Prof. Carlos José Ribas D'ávila, por participar ativamente da minha formação como engenheiro, a minha eterna gratidão.

Por fim, agradeço aos meus amigos Paulo Gentil, Thiago Americano, Mauro Reis, Mamour Sop Ndiaye, Bruno França e a toda equipe LEMT por compartilhar comigo o pensamento de que um dia poderemos melhorar o mundo com nosso conhecimento tecnológico, esforço, dedicação e muito trabalho.

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

ESTUDO E DESENVOLVIMENTO DE UMA PLATAFORMA DE COMUNICAÇÃO SEM FIO PARA REDES ELÉTRICAS INTELIGENTES

Juliano Freitas Caldeira

Novembro/2012

Orientador: Mauricio Aredes

Programa: Engenharia Elétrica

Atualmente, o uso racional de energia elétrica se tornou um dos principais assuntos da agenda mundial, não apenas por questões ambientais e visível aquecimento global, mas também pela incapacidade de suprir a crescente demanda.

Diversos esforços vêm sendo criados em todo o mundo para o desenvolvimento de produtos e tecnologias capazes de auxiliar na redução de consumo, porém a indústria ainda não possui um padrão *de facto* para a plataforma de comunicação a ser utilizada e a maior parte dos consumidores de energia elétrica ainda não possui ferramentas capazes de informar em tempo real o quanto cada equipamento eletroeletrônico consome.

Assim, este trabalho se insere na busca de uma solução economicamente viável, destinada ao monitoramento e maior controle no consumo de energia elétrica, criando condições para a redução do desperdício de energia e os impactos ambientais causados por sua produção excessiva.

Desta forma, cria-se uma nova arquitetura para comunicação, especializada em redes elétricas inteligentes, capaz de transmitir dados de forma bidirecional entre dispositivos sensores e/ou atuadores em redes residenciais, oferecendo serviços à projetos de *smart grid*, visando baixo custo além de alta integração de componentes e serviços, facilitando a disseminação da informação e sua consequente disponibilização aos clientes de energia.

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

STUDY AND DEVELOPMENT OF A PLATFORM FOR WIRELESS
COMMUNICATION NETWORK IN SMART GRID ENVIRONMENT

Juliano Freitas Caldeira

November/2012

Advisor: Mauricio Aredes

Department: Electrical Engineering

Nowadays, the rational use of energy has become one of the main issues on the global agenda, not only by environmental issues and visible global warming, but also by the inability to meet the growing demand.

Several efforts have been established around the world to develop products and technologies that can assist to reduce consumption, but the industry still lacks a standard for the communication platform being used and, most electricity consumers does not have any tools to inform in real time how much each electrical and electronic equipment consume.

This study is inserted in the search for an economically viable solution, designed to monitor and control the consumption of electricity, creating conditions to reduce energy waste and environmental impacts caused by their excessive production.

Thus, it creates a new architecture for communication, specialized in smart grids, able to transmit data between sensors and / or actuators in home area networks, offering services to smart grid projects, aiming low cost plus high integration of components and services, facilitating the dissemination of information and its subsequent availability to energy customers.

Índice

1. Introdução.....	1
1.1. Motivação	1
1.2. Objetivos.....	4
1.3. Descrição dos capítulos do trabalho	5
2. Estado da arte	6
2.1. Arquiteturas avançadas de medição.....	6
2.2. IEEE 802.15.4.....	19
2.3. 6loWPAN.....	28
2.4. IEEE 802.11	33
2.5. PLC	37
2.6. Outros Sistemas	40
3. Projeto do sistema	43
3.1. Visão geral	43
3.2. Projeto da plataforma	45
3.2.1. <i>Hardware</i> de comunicação.....	45
3.2.2. <i>Software</i> de comunicação.....	60
3.3. Levantamento de tarefas e respostas.....	68
3.4. Procedimento de envio de mensagens	77
3.5. Validação por Redes de Petri.....	80
4. Simulação	83
4.1. Descrição do problema.....	83
4.2. Escolha do módulo de simulação	83
4.3. Simulador	85
4.4. Cenários.....	94
4.5. Metodologia de testes	96
4.6. Resultados obtidos.....	99
4.6.1. Transmissão de dados.....	99
4.6.2. Índice de Raj Jain	104
5. Resultados experimentais	106
5.1. Equipamento	106
5.2. Arranjo Experimental	108
5.3. Resultados obtidos.....	110

5.3.1. Capacidade de transmissão	111
5.3.2. Índice de Raj Jain.....	112
5.4. Comentários sobre os resultados	114
5.5. Medições reais	115
6. Conclusões	118
6.1. Epílogo	118
6.2. Trabalhos futuros.....	121
Referências Bibliográficas.....	123
Anexo A.....	129
Resultado da Rede de Petri.....	129

Lista de Figuras

Figura 1 – Oferta Interna de Energia Elétrica por Fonte – Brasil 2010 (EPE BEN, 2011).	2
Figura 2 – Evolução da Energia Elétrica no Brasil (BANCO MUNDIAL, 2012).....	2
Figura 3 – Modelo conceitual de <i>smart grid</i> descrito pelo IEEE (IEEE SG, 2012).....	6
Figura 4 – Potência de transmissão e recepção de diferentes protocolos de comunicação, adaptada de (FADLULLAH, et al., 2011).....	8
Figura 5 – Esquemático proposto como de maior escalabilidade para o <i>smart grid</i> , adaptada de (FADLULLAH, et al., 2011).....	8
Figura 6 – Arquitetura em três camadas proposta, adaptado de (ZHANG, et al., 2011).	10
Figura 7 – Algoritmo <i>Primal-Dual</i> proposto (KODIALAM, et al., 2005).	12
Figura 8 – Topologia proposta para a rede do sistema inteligente (JAMIAN, et al., 2011)	14
Figura 9 – Arquitetura de telecomunicações em camadas proposta para o <i>smart grid</i> (JAMIAN, et al., 2011).....	15
Figura 10 – Proposta de rede em malha para o <i>smart grid</i> , adaptado de (GHARAVI, et al., 2011).....	16
Figura 11 – Redes residenciais com suporte a conexão máquina a máquina (M2M) (STARSINIC, 2010).....	18
Figura 12 – Possíveis topologias de rede, adaptado de (SALMAN, et al., 2010).	20
Figura 13 – Estrutura do <i>superframe</i> , adaptado de (SALMAN, et al., 2010).	21
Figura 14 – Interface para celular do <i>Android</i> , adaptado de (TUDOSE, et al., 2011)...	24
Figura 15 – Modelo de uso da energia proposto, adaptado de (LANGHAMMER, et al., 2012).....	26
Figura 16 – Conversão bidirecional entre pacotes IPv6 e 6LoWPAN, adaptado de (HASBOLLAH, et al., 2010).....	29
Figura 17 – Intercomunicação entre clientes IPv6 e rede de sensores sem fio (WSN), adaptado de (DA SILVA CAMPOS, et al., 2011)	29
Figura 18 – Estrutura básica de camadas do 6LoWPAN, adaptado de (YIBO, et al., 2011).....	30
Figura 19 – Tempo de resposta obtido por número de saltos em uma rede 6LoWPAN, adaptado de (MUKHTAR, et al., 2008).	31
Figura 20 – Arquitetura proposta para redes de medição baseadas em IEEE802.11, adaptado de (JIAN, et al., 2011).....	35
Figura 21 – Utilização de link IEEE 802.11 em áreas de difícil acesso, adaptado de (TANAKA, et al., 2011).....	36

Figura 22 – Protocolo REMPLI como arquitetura de <i>smart grid</i> , adaptado de (SAUTER, et al., 2011).	38
Figura 23 – Padrões de consumo de energia para equipamentos eletroeletrônicos, adaptado de (SEUNGHYUN, et al., 2010).	40
Figura 24 – Interface gráfica do simulador, adaptado de (SEUNGHYUN, et al., 2010)	41
Figura 25 – Projeto de um sistema de comunicação capaz de habilitar o desenvolvimento de uma rede interna de medidores inteligentes.	43
Figura 26 – Esquema proposto para o dispositivo que irá suportar a pilha do protocolo proposto.	44
Figura 27 – Esquema básico de ações e funcionalidades que devem ser suportados pelo sistema.	45
Figura 28 – Sequência de pacotes do IEEE 802.15.4, adaptado de (LATR'E, et al., 2005).	51
Figura 29 – Estrutura do pacote de dados do IEEE 802.15.4, adaptado de (LATR'E, et al., 2005).	52
Figura 30 – Atraso da mensagem por tamanho do <i>TPmax</i> , adaptado de (LATR'E, et al., 2005).	53
Figura 31 – Eficiência da banda por tamanho do <i>TPmax</i> , adaptado de (LATR'E, et al., 2005).	54
Figura 32 – Esquema proposto para o levantamento da banda necessária por canal.	58
Figura 33 – Quadro de dados proposto para o protocolo.	61
Figura 34 – Modelo conceitual para comunicação entre dispositivos com múltiplos saltos.	62
Figura 35 – Diagrama de comunicação exemplificando áreas de influência e interferência entre as redes distintas A, B e C convivendo em um mesmo espaço físico.	64
Figura 36 – Fluxograma de validação das mensagens recebidas pelo protocolo.	66
Figura 37 – Fluxograma de recebimento e validação das mensagens pela camada de enlace.	67
Figura 38 – Sequência de troca de mensagens do tipo tarefa e resposta.	69
Figura 39 – Fluxograma de descobrimento de rotas.	76
Figura 40 – Fluxograma geral do procedimento de envio de mensagens.	77
Figura 41 – Fluxograma do procedimento de envio da camada de enlace do dispositivo.	78
Figura 42 – Primitivas de serviço para modelagem em Redes de Petri.	80
Figura 43 – Rede de Petri do protocolo para dois dispositivos – Emissor / Receptor. ...	81
Figura 44 – Esquema de classes para agentes do sistema.	85
Figura 45 – Diagrama de classes dos agentes do simulador.	86
Figura 46 – Diagrama de classes dos eventos do simulador.	87

Figura 47 - Diagrama de outras classes e estruturas de dados do simulador.....	88
Figura 48 – Probabilidade $P(d)$ para um alcance máximo de 37m.	90
Figura 49 – Relacionamento entre Motor e Eventos do simulador.	93
Figura 50 – Esquema proposto para o cenário de simulação com N sensores.	94
Figura 51- Capacidade de transmissão de dados do sistema, em Kbps, ao aumentar-se a quantidade de dispositivos da rede.	100
Figura 53 – Somatório do número total de colisões no receptor dos dispositivos.	101
Figura 52 – Taxa percentual de pacotes reenviados.	101
Figura 54 – Percentual dos pacotes enviados que não obtiveram resposta dos transmissores.	103
Figura 55 – Percentual de retransmissões por falta de reconhecimento (<i>no ACK</i>).	103
Figura 56 – Valores de <i>Fairness Index</i> quando se aumenta a quantidade de dispositivos.	105
Figura 57 – Microcontrolador MSP430 (A) e CC2500 (B) na placa desenvolvida para testes (<i>layout</i> cedido pela empresa <i>Powersave</i>).	106
Figura 58 – Placas prontas a serem embarcadas em tomadas, disjuntores e outros dispositivos eletroeletrônicos.	107
Figura 59 – Esquema experimental de comunicação.	108
Figura 60 – Comunicação pela porta USB a partir de um módulo TL.....	109
Figura 61 – Capacidade conjunta de transmissão de dados média (após três rodadas).111	
Figura 62 – Capacidade média individual de transmissão de dados.	112
Figura 63 - <i>Fairness Index</i> para múltiplos sensores.	113
Figura 64 – Comparação das medições do equipamento desenvolvido com um medidor SAGA2000 da <i>Landis+Gyr</i> para uma geladeira.	115
Figura 65 – Curva de consumo de energia para calibração do medidor <i>Powersave</i>	116
Figura 66 – Dados de medições reais coletadas pelo equipamento (em kW).	117

Lista de Tabelas

Tabela 1 - Variáveis para validação do CSMA/CA, adaptado de (SALMAN, et al., 2010; IEEE Std 802.15.4, 2003).....	22
Tabela 2 – Comparação da média de mensagens por 10 minutos de SNMP, adaptado de (CHOI, et al., 2009).....	32
Tabela 3 – Tabela sumário dos resultados em diferentes <i>chipsets</i> BPL adaptado de (LIU, et al., 2010).	39
Tabela 4 - Comparativo entre as alternativas avaliadas de <i>hardware</i> para o projeto.....	47
Tabela 5 - Pacote de rádio com dados úteis do protocolo proposto a partir dos dados do fabricante (TEXAS INSTRUMENTS, 2012).....	55
Tabela 6 - Tabela exemplo para configuração da Figura 34.	63
Tabela 7 – Primitivas de tarefas e respostas utilizadas pelo protocolo.	71
Tabela 8 – Resultados obtidos após a coleta de dados do experimento.	110

1. Introdução

1.1. Motivação

Atualmente o uso racional de energia elétrica se tornou um dos principais assuntos da agenda mundial, não apenas por questões ambientais e visível aquecimento global, mas também pela incapacidade de suprir a crescente demanda.

Diversos esforços vêm sendo criados em todo o mundo para o desenvolvimento de produtos e tecnologias capazes de auxiliar na redução de consumo, porém a indústria ainda não possui um padrão *de facto* a ser utilizado, e a maior parte dos consumidores de energia elétrica ainda não sabe realmente o quanto cada equipamento eletroeletrônico consome diariamente.

Esforços também vêm sendo realizados no desenvolvimento de novas tecnologias para a obtenção de energia elétrica através de fontes renováveis, tais como painéis fotovoltaicos e de aerogeradores. Porém, estas tecnologias ainda são mais caras do que as tradicionais formas de obtenção de energia a partir de combustíveis fósseis além de serem dependentes do clima (incidência de raios solares, ventos etc.) e região geográfica onde serão instalados, apresentando ainda uma baixa eficiência (em torno de 13% com máximo de 20,35% de acordo com pesquisa (SROECO SOLAR, 2012)). Desta forma, fica inviabilizada a sua adoção em massa como principal fonte de energia para o abastecimento de grandes cidades.

Através das informações do Ministério de Minas e Energia (EPE BEN, 2011), mostradas na Figura 1, percebe-se que o Brasil é um país onde grande parte da energia é gerada por hidrelétricas. Porém, no restante do mundo (grande maioria dos países), a abundância de recursos naturais renováveis não é a realidade, transformando o uso consciente e eficiente da energia elétrica um fator determinante para o crescimento de suas economias.

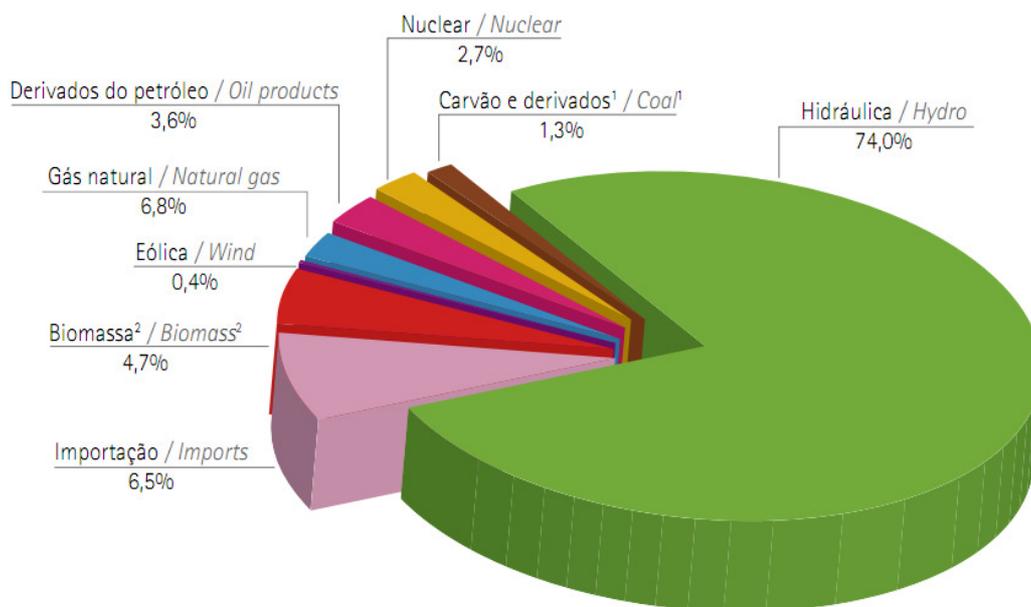


Figura 1 – Oferta Interna de Energia Elétrica por Fonte – Brasil 2010 (EPE BEN, 2011).

No Brasil o consumo de energia vem crescendo a cada ano como podemos observar na Figura 2 baseada nos dados obtidos para minas e energia (BANCO MUNDIAL, 2012). As previsões de crescimento no setor de eletricidade indicam que o consumo *per capita* brasileiro deverá dobrar ou até triplicar até 2030 (EPE BEN, 2011). Estas previsões não são apenas uma realidade para o cenário brasileiro, mas podem ser estendidas para diversos outros países.

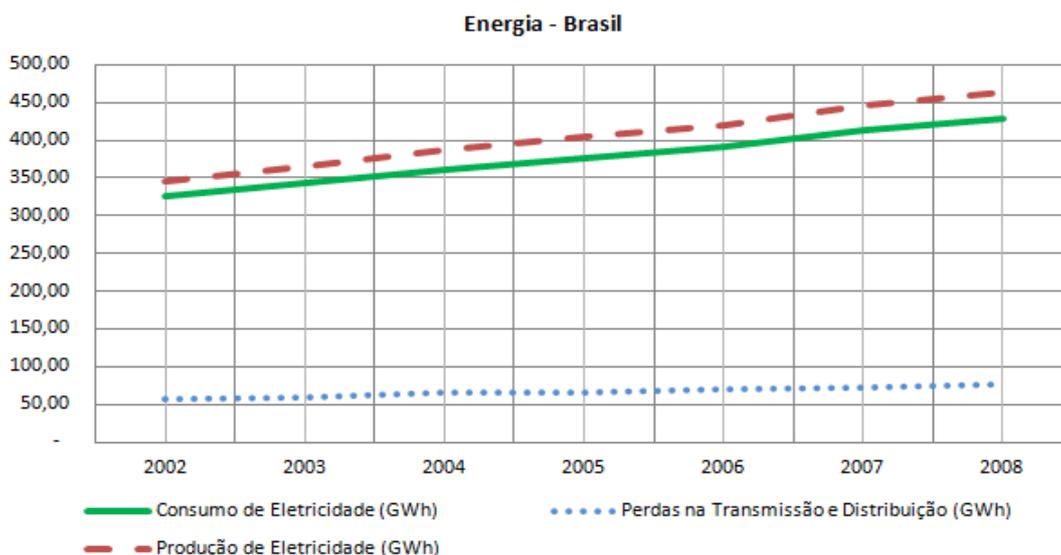


Figura 2 – Evolução da Energia Elétrica no Brasil (BANCO MUNDIAL, 2012).

Desta forma, a criação de tecnologias e produtos inovadores que propiciem o aumento da eficiência a partir do uso racional de energia elétrica vem sendo tratada cada vez mais em temas de publicações internacionais. A revista *The Economist* (THE ECONOMIST, 2008) cita a eficiência energética como o 5º tipo de combustível (“*the fifth fuel*” – carvão, petróleo, nuclear e energias alternativas são as outras citadas) onde diz que, além de evitar a emissão de gases que contribuem para o aquecimento global, reduz gastos desnecessários em contas de energia para os consumidores e novos gastos em infraestrutura para suprir a demanda.

A ABESCO e GTZ indicam que o nível de desperdício de energia elétrica no Brasil chega a uma cifra de R\$15 Bi (REVISTA FATOR, 2010). O mesmo estudo mostrou que o uso inteligente dos recursos proporciona a disponibilização de energia até 35% mais barata do que o custo de produção, além de todos os benefícios ambientais mencionados.

Assim, projetos relacionados ao tema do ‘uso de novas tecnologias para eficiência energética’ vêm atraindo a atenção de diversas empresas e do meio acadêmico para solucionar, com pesquisa e desenvolvimento, e produtos inovadores, os problemas e desafios intrínsecos ao tema (AGÊNCIA BRASIL, 2012).

Diversas propostas vêm sendo amplamente discutidas com diferentes abordagens sobre o assunto: fontes renováveis, *smart metering*, redes elétricas inteligentes, tecnologias eficientes e seguras para a transmissão bidirecional de dados, aumento da eficiência, entre outros.

Desta forma, a partir de uma pesquisa em artigos da comunidade científica foi observado que o desenvolvimento de novos protocolos de comunicação para o transporte de dados dos medidores eletrônicos de energia, peça fundamental para o início da instalação de uma rede *smart grid*, vem sendo propostos por grupos do IEEE (IEEE 802.11, 2012; IEEE 802.15.4, 2012; OPERA, 2004) e em protocolos comerciais, como o *Zigbee* (ZIGBEE ALLIANCE, 2012) e *Z-Wave* (Z-WAVE, 2012).

Além disto, pouco se fala no desenvolvimento de alternativas de baixo custo, com a otimização no uso de recursos computacionais especificamente para o mercado de energia, o que abre espaço para novos estudos e o desenvolvimento de novas plataformas e pilhas de protocolo que sejam capazes de aprimorar as alternativas comerciais existentes.

Neste contexto se insere este trabalho: no desenvolvimento de um dispositivo capaz de, utilizando um *hardware* com baixo poder computacional e de baixo custo, medir e transmitir os dados de consumo de energia de forma eficiente, com as especificidades e características da medição intervalar requeridas no *smart grid*, e se tornando uma plataforma viável para utilização e adoção em massa para as redes elétricas do futuro.

1.2. Objetivos

Abaixo estão apontados os principais objetivos deste trabalho:

- Avaliar as soluções em telecomunicações para *smart grid* existentes no mercado e artigos científicos, procurando analisar seus pontos fortes e fracos com base nos requisitos necessários para a medição intervalar proposta pelo mercado de energia elétrica, buscando soluções de maior eficiência para adoção em massa;
- Baseado nos pontos fortes dos sistemas existentes, avaliar e propor uma plataforma capaz de atender aos requisitos, visando baixo custo final de implantação com eficiência e alto valor agregado ao equipamento;
- A partir do desenvolvimento de uma arquitetura que suporte a simulação do sistema, validar a arquitetura proposta e suas limitações para futuras expansões, visando explorar a plataforma em suas limitações e capacidades;
- Implementar o algoritmo e a plataforma proposta em microcontrolador e *hardware* de comunicação básicos, de forma a explorar as capacidades do sistema através de ensaios e testes específicos com o objetivo de iniciar o processo de desenvolvimento;
- Validar sua implementação com parâmetros e métricas para tirar conclusões e novas possibilidades para trabalhos e outros desenvolvimentos futuros.

1.3. Descrição dos capítulos do trabalho

No Capítulo 2 encontra-se uma pesquisa bibliográfica dos diversos padrões e protocolos existentes relacionados com o tema *smart grid*, suas aplicações, arquiteturas e descrições sobre o seu funcionamento e desempenho em diferentes ambientes.

Após uma visualização geral do problema e das soluções atuais de protocolos e padrões existentes, no Capítulo 3 é feita a proposta de projeto de uma nova arquitetura, capaz de cumprir com os requisitos de projeto e possuir um diferencial competitivo com as atuais soluções disponíveis no mercado.

Com o intuito de validar o funcionamento do que foi proposto, antes da fase de implementação, faz-se necessário executar simulações da rede como um todo, de forma que possamos averiguar sua capacidade máxima. Nesta etapa serão descritos os requisitos, desenvolvimento e o funcionamento do simulador capaz de avaliar estes resultados. Esta etapa é descrita no Capítulo 4.

No Capítulo 5, será demonstrado o resultado prático após a codificação e implementação do sistema, além de suas comparações com o modelo teórico simulado. Desta forma será possível avaliar de forma objetiva detalhes do projeto e caso os resultados sejam divergentes em relação ao simulado, encontrar possíveis falhas de projeto e/ou modelos de simulação que estejam em desacordo.

Por fim, no Capítulo 6 serão demonstradas as conclusões do trabalho e possíveis trabalhos futuros que possam vir a ser desenvolvidos em um momento posterior para aprimoramentos e melhorias no atual estágio de desenvolvimento da plataforma de comunicação.

2. Estado da arte

2.1. Arquiteturas avançadas de medição

Antes de avaliarmos as tecnologias existentes, é preciso analisar a complexidade das arquiteturas envolvidas no *smart grid*. O IEEE disponibiliza um modelo conceitual (IEEE SG, 2012) dos elementos que estão presentes na cadeia produtiva da energia elétrica e que participam das redes elétricas inteligentes, conforme a Figura 3.

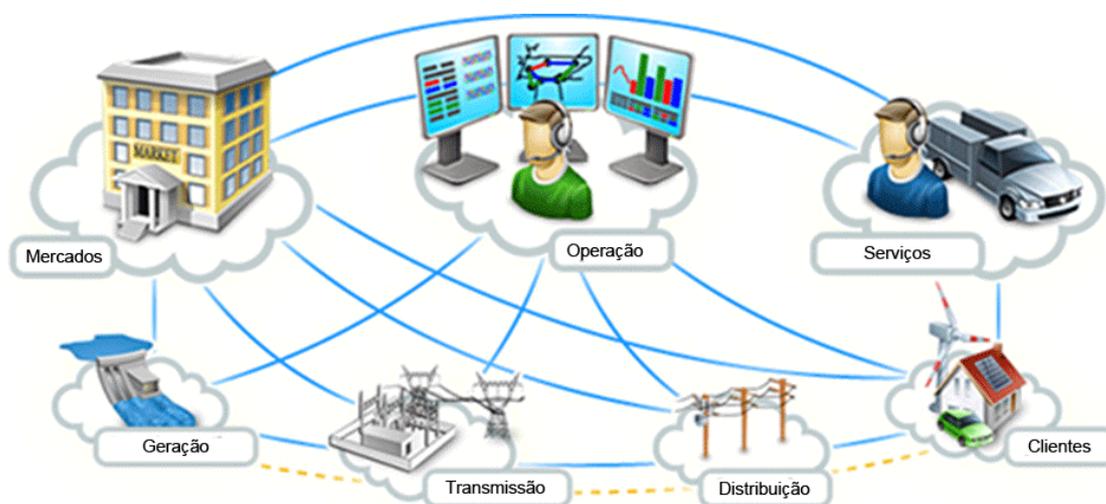


Figura 3 – Modelo conceitual de *smart grid* descrito pelo IEEE (IEEE SG, 2012)

O foco deste trabalho sobre redes elétricas inteligentes está voltado para o setor de distribuição e no segmento dos consumidores finais de energia (Clientes). Especificamente, será direcionado o foco para a área de telecomunicações para telemedição, automação e sensoamento das redes domésticas, assunto ainda muito comentado, mas que ainda não se trata de uma realidade para os consumidores, conforme descrito em pesquisa realizada pela IBM (IBM, 2009), onde é expressa a vontade dos consumidores por sistemas de auxílio à eficiência energética.

Uma interessante pesquisa descreve sobre tecnologias de comunicação existentes para redes inteligentes, visando diferentes utilizações e abordagens (FADLULLAH, et al., 2011). Neste estudo são apresentados:

- **IEEE 802.15.3a** – Conhecido como *UWB*, é demonstrado (FADLULLAH, et al., 2011) que para se alcançar tal banda (60GHz) é necessário um alto consumo de energia. Após diversos impasses para seu desenvolvimento, no ano de 2006, o grupo de trabalho foi extinto;
- **IEEE 802.11** – Também conhecido como *WiFi*, este protocolo foi desenvolvido para atingir altas taxas de transferência de dados. Possui, teoricamente, um alcance capaz de transmitir e receber dados por toda uma residência, e possui consumo de energia de alto a moderado, comparado a outras plataformas. Desta forma, o *WiFi* foi considerado impróprio para o uso de comunicação em massa por dispositivos do *smart grid*;
- **IEEE 802.15.1** – Presente na maioria dos dispositivos celulares móveis, o *Bluetooth* é extremamente popular entre as aplicações de transmissão de dados e áudio em curtas distâncias. O consumo de energia também é extremamente baixo, se comparado a outras tecnologias. Porém, o *Bluetooth* trabalha com o conceito de “*piconets*” que comportam apenas oito dispositivos conectados simultaneamente. Assim, seria necessário conectar diversas “*piconets*” o que aumentaria a latência de comunicação para sua adoção em massa;
- **IEEE 802.15.4** – Utilizando o IEEE 802.15.4 e um protocolo nas camadas superiores, tais como o Zigbee, é possível criar redes sem fio de baixo consumo, com diversas configurações (estrela, em malha etc.) com alta flexibilidade. O IEEE 802.15.4 foi criado para a comunicação entre dispositivos de uma rede doméstica. Assim, observa-se a escolha do autor (FADLULLAH, et al., 2011) pelo IEEE802.15.4 como melhor alternativa para redes elétricas inteligentes onde há comunicação entre equipamentos.

Além disto, é avaliada a potência média de transmissão para as diferentes tecnologias apresentadas, como observado na Figura 4.

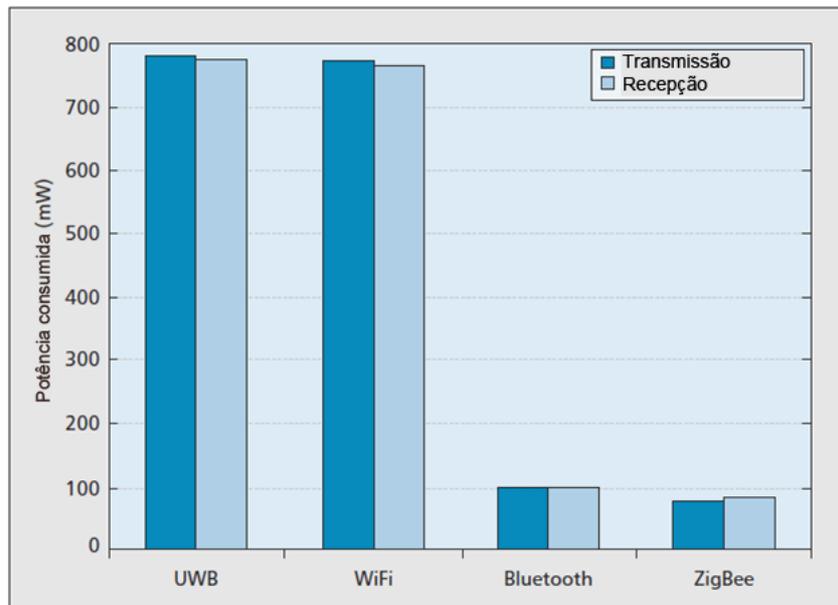


Figura 4 – Potência de transmissão e recepção de diferentes protocolos de comunicação, adaptada de (FADLULLAH, et al., 2011).

De forma a transmitir os dados até o centro de controle da concessionária de energia, é proposta a utilização do suporte das redes de telefonia celular 3G como o link de dados agregador de diversas redes domésticas (HAN) em redes prediais (BAN) e/ou de vizinhança (NAN) para que a informação seja levada até os sistemas de banco de dados e telemedição das empresas fornecedoras, como mostrado na Figura 5.

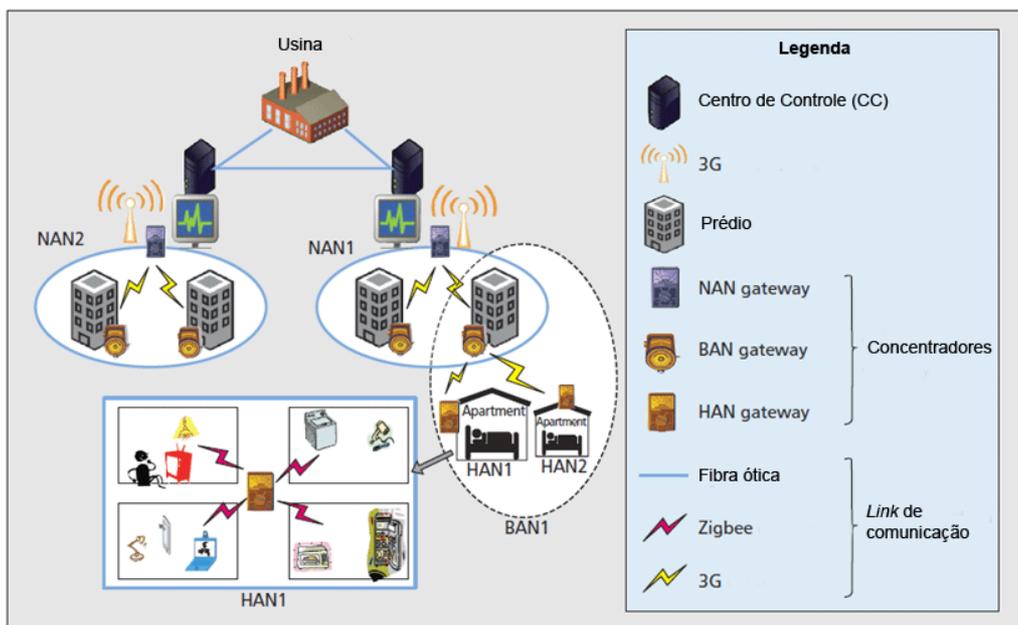


Figura 5 – Esquemático proposto como de maior escalabilidade para o *smart grid*, adaptada de (FADLULLAH, et al., 2011).

Além desta arquitetura, também foi observado (ZHANG, et al., 2011) que a utilização conjunta de IEEE 802.15.4 com *Zigbee* e IEEE 802.11 é uma alternativa viável, além da tecnologia IEEE 802.16 (*WiMAX*) e PLC aplicados aos diferentes níveis de redes residenciais (HAN), redes de vizinhanças (NAN) e redes de longa distância (WAN), propondo algoritmos de cooperação para evitar o congestionamento de um único *link* de dados.

Para as redes residenciais (HAN), são determinados (ZHANG, et al., 2011) três módulos distintos:

- **Módulo de serviço (SM)** – Responsável por prover o custo e valor da energia em tempo real aos consumidores;
- **Módulo de medição (MM)** – Responsável por medir e gravar dados de medições;
- **Sistema de controle de medição (MCS)** – Responsável por coletar os dados de MM e transmitir as informações para SM.

Na segunda camada da arquitetura (BAN e NAN), é proposta (ZHANG, et al., 2011) a interconexão entre diferentes MCS que estão geograficamente próximos. Assim, são incluídos dois novos agentes:

- **Central de controle de acesso (CAC)** – Interface de gerencia a comunicação entre as HANs e o provedor de energia;
- **Coletor de dados de medidores inteligentes (SMDC)** – Este dispositivo é um nó de uma rede sem fio que agrupa os dados de medição de toda a vizinhança e/ou comunidade;

Para a última camada proposta (o autor denominou de WAN), é previsto que ela seja a provedora de serviços de comunicações para as NANs, subestações, equipamentos do sistema de distribuição (EDS), remotas, e a própria concessionária de energia.

Esta camada irá agrupar os sistemas de controle, supervisão e aquisição de dados (SCADA) e de serviços corporativos e de controle de energia (E&SC), conforme mostrado na Figura 6.

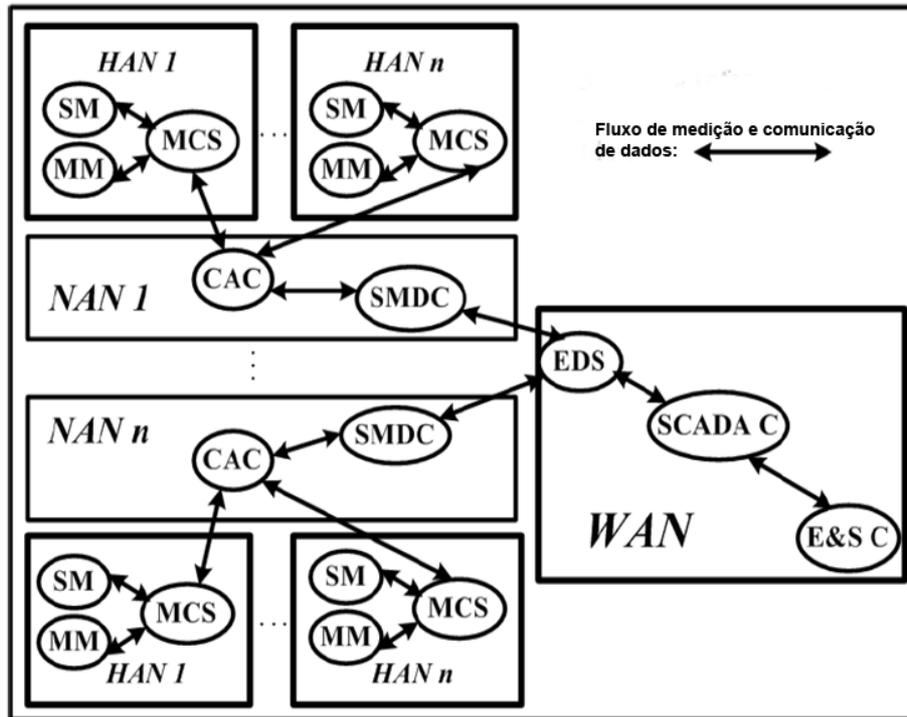


Figura 6 – Arquitetura em três camadas proposta, adaptado de (ZHANG, et al., 2011).

Para esta arquitetura, foi proposto que as redes sem fio (presentes na HAN e BAN/NAN) se utilizassem do recurso de redes em malha, por possuir benefícios e propriedades de caminhos múltiplos, compensando possíveis falhas através de redundância, possuindo alta escalabilidade, sendo uma rede altamente dinâmica e que pode contornar danos, mantendo a comunicação estável.

Para as HANs a tecnologia escolhida pelo autor (ZHANG, et al., 2011) foi IEEE 802.15.4 com *Zigbee*. Por possuir 16 canais (N_c) não sobrepostos em 2.4GHz com capacidade de $r = 250\text{kbps}$ cada, foi calculada a taxa máxima $C_{\max} = 4\text{Mbps}$ a partir da equação:

$$C_{\max} = r \times N_c \quad (2.1)$$

Dentro do conceito das redes de vizinhanças (NAN), foi escolhido o padrão IEEE 802.11n, que possui taxas de transmissão de até 300Mbps na banda de 2.4GHz, que é compartilhada com a camada física (PHY) do IEEE 802.15.4 com *Zigbee*.

Para a interconexão das WANs, como os pontos são geralmente fixos e fisicamente distantes uns dos outros foram utilizadas redes PLC (*powerline communication*) para a transmissão da grande maioria dos dados e todo o controle foi feito por redes IEEE802.16-2009 (*WiMAX*) que podem prover comunicação em distancias de até 3,2Km. Este tipo de protocolo possui taxas de transmissão de 70Mbps utilizando técnicas de transmissão OFDM (*orthogonal frequency division multiplexing*).

De forma a otimizar o fluxo de informações e manter os requisitos de capacidade do canal dentro dos limites físicos, foi descrito (ZHANG, et al., 2011) um algoritmo para garantir o fluxo de dados constante da rede, garantindo a sua estabilidade.

Supondo que $x_i^t(e)$ é a variável de programação que indica no canal i do *link e* durante o intervalo de tempo t . Se $x_i^t(e) = 1$ temos que o canal está transmitindo. Desta forma, podemos afirmar que o número de canais ativos, $C(e)$, durante um intervalo de tempo t é:

$$\sum_{i \in C} x_i^t(e) \leq C(e) \quad e \in N \quad (2.2)$$

Além disto, pode ser observado que existem λ pares ($S_1, S_2, \dots, S_\lambda$) de canais e *links* entre os diversos pares de nós da rede que podem comunicar-se diretamente uns com os outros. Cada par S_j restringe o canal i ao *link e*. A cada par é atribuída uma constante β (S_j) de acordo com a seguinte restrição:

$$\frac{1}{\beta(S_j)} \sum_{(e,i) \in S_j} \frac{f_i(e)}{c_i(e)} \leq 1 \quad , \quad j \in \{1, 2, \dots, \lambda\} \quad (2.3)$$

Onde $f_i(e)$ representa o fluxo de informações transmitidos no canal i e link e .

A comunicação entre os nós do sistema deve existir de forma consistente para M pares (origem, destino). Para cada par, a capacidade do i -ésimo canal é dada por $p_i^m(e)$ e o fluxo total, $r(m)$, transmitido entre os pares pode ser calculado pela equação 2.3 como:

$$r(m) = \sum_{e=s(m)} \sum_{i \in C} p_i^m(e) \quad (2.4)$$

Assim, foi proposto um algoritmo (KODIALAM, et al., 2005), demonstrado na Figura 7, capaz de minimizar o custo de comunicação, calculando o caminho e fluxo ótimos para a comunicação entre os dispositivos.

```

w(j) = δ, j ∈ {1,2,..., λ}
b = 0
While ∑j=1 w(j) < 1
  For m = 1,2,...,M
    r = r(m)
    While r > 0
      Set Δw(j) =  $\frac{\sum_{j \in S_i} \alpha(j)}{c_i(e)}$ 
      Pmin(s(m),d(m)) = min ∑m ∈ M Δw(j)
      μ = min(f(Pmin))
      σ = min{r, μ}
      r = r - σ
      fi(e) = fi(e) + σ
      w(j) = w(j) * (1 +  $\frac{\epsilon \sigma}{f(P_{min})}$ )
    endwhile
  endfor
  b = b + 1
endwhile
g = max  $\left( \sum_{j \in S} \frac{f_i(e)}{c_i(e)} \right)$ 
SF =  $\frac{b}{g}$ 

```

Figura 7 – Algoritmo *Primal-Dual* proposto (KODIALAM, et al., 2005).

*Onde SF é o máximo fator de escala, que demonstra a capacidade restante necessária pela rede.

No que diz respeito ao mercado de distribuição de energia, foi avaliada uma publicação (JAMIAN, et al., 2011) para arquiteturas voltadas para o gerenciamento de cargas em *smart grid*, onde foi proposta uma plataforma que ajusta o consumo de energia dos usuários de acordo com a demanda da rede. Desta forma, uma das premissas para a estabilidade do sistema é manter um balanço em tempo real entre a potência ativa fornecida pelo sistema de geração e a demanda, ou seja, um mecanismo de equilíbrio do sistema que evita o colapso da rede, e suas consequências, tais como, um possível “apagão” por sobrecarga da rede de distribuição.

Estudos demonstram que a estabilidade dos geradores, a capacidade das linhas de transmissão e perdas são os grandes limitantes do sistema elétrico e que, de modo a evitar o colapso, são sugeridas que algumas grandes cargas devam ser removidas da rede desenergizando-as completamente do sistema (VERMA, et al., 2010; RAJAGOPAL, et al., 2011) nos horários de ponta.

Como uma forma de evitar a desenergização direta das grandes cargas, observa-se uma nova proposta (JAMIAN, et al., 2011) para controlar o sistema através do monitoramento e controle do consumo de todo o sistema, onde não há desenergização de grandes grupos de cargas, mas sim, o ajuste dinâmico do controle da demanda, sempre que necessário, para manter a estabilidade do sistema. Esta técnica começou a ganhar o interesse dos pesquisadores com a introdução dos veículos elétricos (EV) e energia renovável (painéis fotovoltaicos).

Para que esta linha de pesquisa seja aplicada, o uso de sensores, atuadores e tecnologias de comunicação tornou-se uma necessidade. Desta forma, a proposta (JAMIAN, et al., 2011) inclui um sistema híbrido, de tecnologias distintas, composto pelas entidades ilustradas na Figura 8:

- Equipamentos Inteligentes (*Smart Appliances - SA*);
- Medidores Inteligentes (*Smart Meters - SM*);
- Servidores Inteligentes (*Smart Servers – SS*);
- Servidores Regionais Inteligentes (*Smart Regional Servers – SRS*).

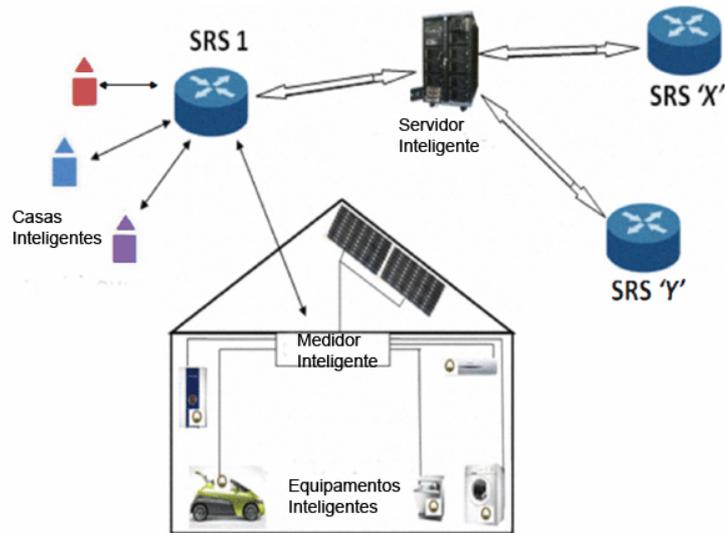


Figura 8 – Topologia proposta para a rede do sistema inteligente (JAMIAN, et al., 2011)

Foram definidos também três tipos de aplicações para as cargas inteligentes:

- **Cargas baseadas em temperatura** – São descritos por este tipo de modelo os aquecedores de água, geladeiras, freezers que podem ser ligados e desligados por certo período de tempo sem afetar a temperatura média de funcionamento dos mesmos;
- **Cargas baseadas em modo de operação** – Para cargas que podem operar em múltiplos modos de operação. Como exemplo, podemos citar os aparelhos de ar condicionado. O usuário poderá ajustar os modos de operação que variam de 19°C a 25°C e o sistema controlador irá analisar e decidir qual a temperatura que será ajustada de acordo com o contexto de utilização macro do sistema de energia;
- **Cargas baseadas em tempo de operação** – Máquinas de lavar roupas e louças são exemplos de cargas que podem adiar suas operações sem que seja sentido um grande impacto sentido pelo consumidor. Este tipo de carga deverá possuir uma menor prioridade no acesso à energia por não serem críticos aos usuários do sistema.

A proposta de arquitetura de comunicação entre as entidades propostas (JAMIAN, et al., 2011) e o Centro de Controle da Concessionária (CS) é apresentada na Figura 9:

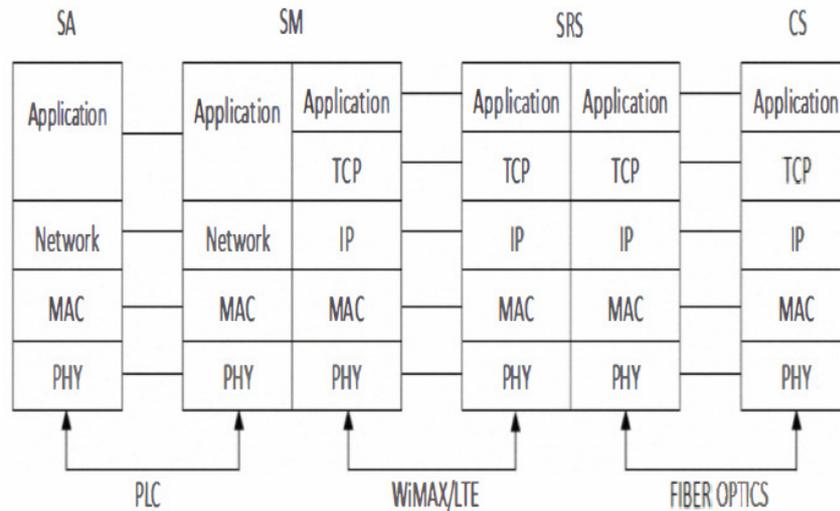


Figura 9 – Arquitetura de telecomunicações em camadas proposta para o *smart grid* (JAMIAN, et al., 2011)

Podemos observar que a comunicação entre os equipamentos/eletrodomésticos inteligentes (SA) e os medidores inteligentes (SM) é implementada através da tecnologia PLC, utilizando um protocolo de 1975, chamado X10 (SMART HOME USA, 2012). Para a comunicação entre os medidores inteligentes (SM) e os servidores (SRS) a proposta é que seja utilizada a comunicação *WiMAX* ou LTE (4G). Estas informações são então levadas até o centro de controle da concessionária por um *link* de fibra ótica, aonde serão analisados e processados os dados para que uma nova iteração seja executada pelo algoritmo.

Em uma abordagem mais sofisticada, envolvendo problemas de comunicação e possíveis falhas na transferência de dados e controles das redes de *smart grid*, é proposto (GHARAVI, et al., 2011) um sistema com múltiplos pontos de acesso (*gateways*) em uma rede em malha sem fio para a comunicação entre os equipamentos eletroeletrônicos e os medidores inteligentes. Desta forma, utilizando recursos avançados de rede IEEE802.11s (IEEE 802.11, 2012) é possível expandir o número de rotas, utilizando múltiplos *gateways*, tráfego de dados em tempo real e alta confiabilidade de redes em malha sem fio, como pode ser observado na Figura 10.

Os dispositivos deverão ser programados para se beneficiar desta arquitetura de múltiplos gateways e assim conseguir executar de forma efetiva o balanceamento de cargas de dados na rede em termos de máxima performance em vazão, confiabilidade e reconfiguração/recuperação em caso de danos e/ou panes em equipamentos da rede.

Com a utilização do protocolo IEEE802.11s passa a ser permitido para o sistema o reencaminhamento de mensagens e seleção de caminhos na camada 2 (enlace), além de apenas na camada 3 do modelo OSI (TANENBAUM, 2003). Assim, em toda estratégia de roteamento são utilizados endereços MAC, que podem suportar mensagens em *unicast*, *multicast* e *broadcast*. Mesmo os dispositivos que, porventura, não suportem o protocolo de redes em malha sem fio, tais como o IEEE 802.11b, podem receber uma atualização no pacote de serviços estendidos (ESS MESH, 2006). Desta forma, são classificados como nós escravos e enviam dados através de redes em malha através de seus pontos de acesso.

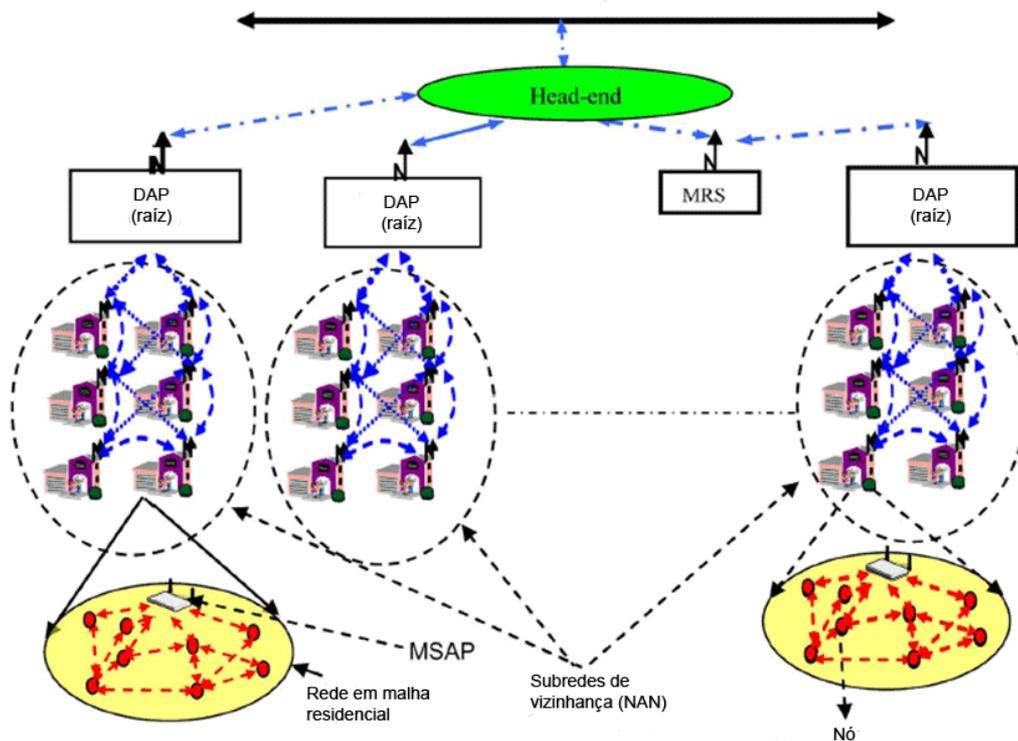


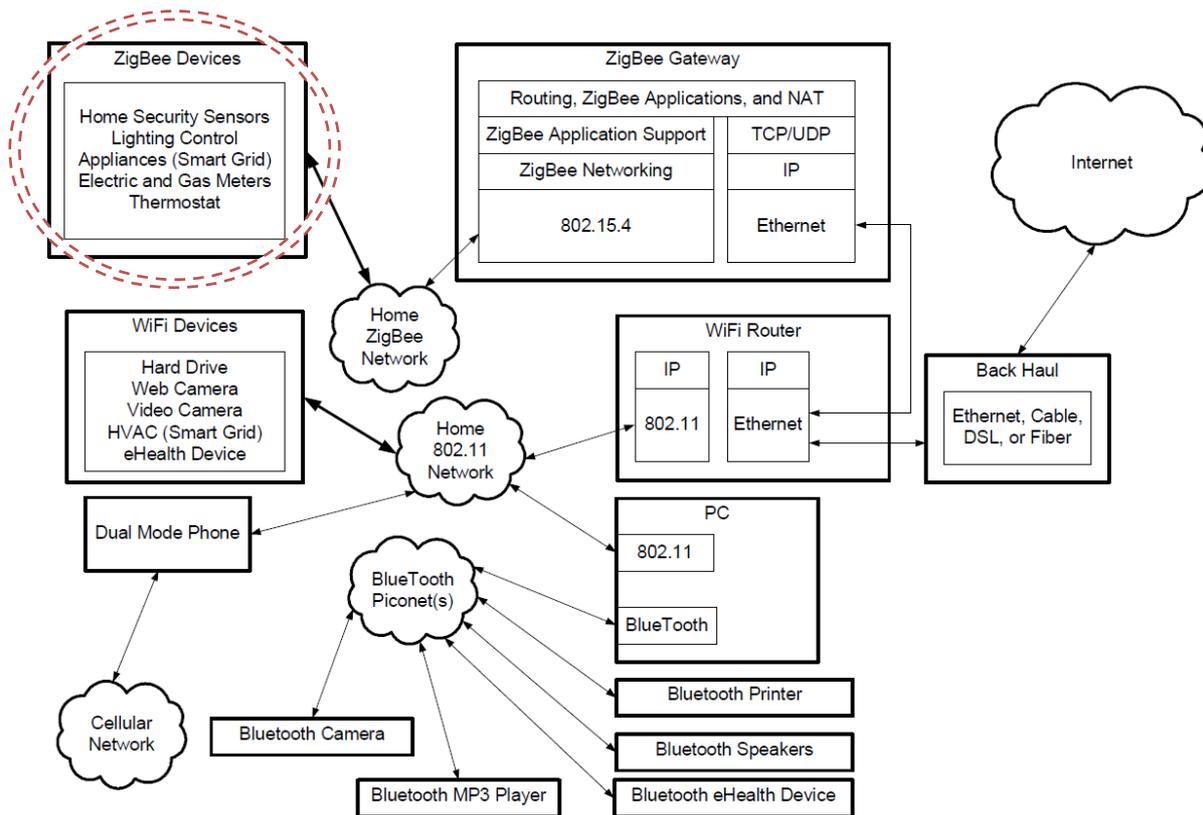
Figura 10 – Proposta de rede em malha para o *smart grid*, adaptado de (GHARAVI, et al., 2011).

Podemos observar na Figura 10 os conceitos de:

- **Estação de retransmissão em malha** (MRS – *Mesh relay station*): Representa uma unidade que apenas retransmite os pacotes de dados na rede em malha, redirecionando-os aos destinatários;
- **Estação de ponto de acesso em malha** (MSAP – *Mesh-station with access point*): Representa o ponto do medidor residencial, funcionando como um *gateway* entre a rede doméstica e os pontos de agregação de dados (DAP – *Data aggregation points*).
- **Estações de agregação de dados** (DAP – *Data aggregation point station*): Representam os *gateways* agregadores das redes de vizinhanças. Estas estações são consideradas as raízes das árvores em cada sub-rede do sistema.
- **Estação portal mestre** (MGS – *Master gateway station*): Representam o ponto de interconexão da interface de medição avançada. É o ponto físico e lógico onde é feita a interconexão com o *backbone* da concessionária para a validação dos dados.

A partir desta plataforma são propostas diversas arquiteturas híbridas (GHARAVI, et al., 2011) de roteamento baseados em uma combinação de protocolos reativos (sob demanda) e proativos, tendo como função manter o estado das rotas, ao mesmo tempo em que se reduzem os impactos de mudanças na topologia da rede. Como exemplo, pode-se observar o protocolo de roteamento HWMP descrito pelo IEEE 802.11s *Task Group* (ESS MESH, 2006).

Após uma análise criteriosa dos padrões existentes em redes sem fio para comunicação máquina a máquina (M2M), foram avaliadas (STARSINIC, 2010) quais tecnologias seriam ideais para cada tipo de aplicação, novamente sugerindo que as redes IEEE802.15.4 fossem utilizadas quando o assunto é controle e medição de energia elétrica, conforme avaliação total descrita na Figura 11.



ZigBee	6LoWPAN	Wi-Fi	Bluetooth
Applications	Applications	Applications	Applications
ZigBee Application Support Layer	UDP Transport	UDP/TCP Transport	HID TCP/UDP RFCOMM
ZigBee Network Layer	IPv6 Network Layer 6LoWPAN Adaption	IPv6 / IPv4 Network Layer	SDP IP
802.15.4 MAC Layer	802.15.4 MAC Layer	802.11 MAC Layer	L2CAP / Link Mgr. / Link Controller
802.15.4 Phy Layer	802.15.4 Phy Layer	802.11 Phy Layer	Bluetooth PHY

Figura 11 – Redes residenciais com suporte a conexão máquina a máquina (M2M)
(STARSINIC, 2010)

Desta forma, uma análise mais elaborada sobre os padrões citados como principais pela literatura deverá ser executada para que sejam avaliados cenários e características intrínsecas destes sistemas e plataformas que garantem diferenciais em inovação frente a outras tecnologias existentes.

2.2. IEEE 802.15.4

O objetivo desta seção é demonstrar, a partir de artigos científicos publicados, a tecnologia da padronização IEEE802.15.4 (IEEE Std 802.15.4, 2003) e sua implementação na camada de aplicação, executada pela *Zigbee Alliance* (ZIGBEE ALLIANCE, 2012). Também será abordado o tema sobre algoritmos de roteamento sob demanda (reativos) e redes IP que funcionam baseadas em redes de sensores sem fio com protocolo IETF 6LoWPAN (IETF 6lowpan, 2012) procurando obter uma maior interoperabilidade entre os diferentes sistemas existentes para sensores e compatibilidade com o atual padrão de redes IP.

No ano de 2003, foi criado o padrão IEEE802.15.4. Após a aprovação da documentação original 803.15.4-2003 foram criados quatro grupos de trabalho (TG4a, TG4b, TG4c e TG4d) (SALMAN, et al., 2010) responsáveis por adicionar e criar alternativas para a camada física, segurança, localização, que foram sendo posteriormente adicionadas aos padrões (IEEE 802.15.4a-2007, IEEE802.15.4b-2006, IEEE802.15.4c-2009 e IEEE802.15.4d-2009).

Mesmo com diversas alterações do padrão ao longo dos anos, as topologias definidas na versão original do padrão de 2003 foram mantidas. Assim, este padrão (IEEE Std 802.15.4, 2003) define dois tipos distintos de dispositivos:

- **FFD** (*Full-Function devices*) – Capaz de executar os papéis de coordenador da rede local (PAN *coordinator*) ou simplesmente um dispositivo (*device*);
- **RFD** (*Reduced-Function device*) – Capaz de executar aplicações bem simples. Possui recursos computacionais reduzidos e pouca memória.

De acordo com o padrão (IEEE Std 802.15.4, 2003), uma rede PAN (*personal area network*) pode operar em topologia estrela. Redes mais complexas tais como redes em malha, não são previstas pelo padrão original (SALMAN, et al., 2010). Assim, a sua execução e implementação fica para as camadas superiores à camada de enlace (camada 2 do modelo OSI (TANENBAUM, 2003)), em tecnologias como o *Zigbee*.

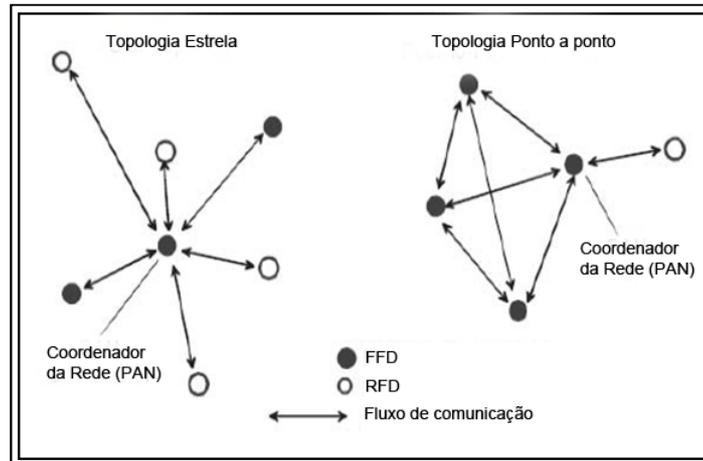


Figura 12 – Possíveis topologias de rede, adaptado de (SALMAN, et al., 2010).

A **camada física** do IEEE 802.15.4-2003 prevê o funcionamento em três faixas/frequências de operação:

- 2,4 GHz – Banda ISM;
- 868 MHz – Europa;
- 915 MHz – América.

O projeto, a arquitetura física, e a simulação de um *transceiver* IEEE802.15.4 podem ser avaliados em (LEE, et al., 2012). A banda de 2,4GHz prevê dezesseis canais com 5MHz de guarda entre os canais, indo de 2400MHz a 2483,5MHz.

Como padrão de espalhamento de espectro é utilizado o DSSS (*direct sequence spread spectrum*) com uma sequência de 32 *chirps* (sequências de pseudorruídos com duração muito menor do que um *bit* de informação) para a especificação de um símbolo, modulado em O-QPSK (*offset quadrature phase-shift keying*). Desta forma, quando o receptor utiliza a mesma sequência de *chirps* produzidos pelo transmissor, a informação original é reconstruída e pode ser entendida no receptor. A taxa máxima de transmissão por canal é de 250Kbps (ZHANG, et al., 2011).

Na banda de 868MHz está previsto apenas um canal, com 20Kbps e, na banda de 915Mhz, dez canais, com taxa de transferência máxima de 40Kbps e 2MHz de guarda (espaçamento para evitar interferências) entre cada canal. A modulação para a faixa abaixo de 1GHz é a BPSK (*binary phase-shift keying*) com uma sequência de 15 *chirps* para o espalhamento de espectro em DSSS.

Os padrões subsequentes ao de 2003 (IEEE 802.15.4b-2006, IEEE 802.15.4a-2007, IEEE 802.15.4c-2009 e IEEE 802.15.4d-2009) adicionaram à camada física uma maior taxa de dados, novos canais nas bandas e até a utilização de UWB (*Ultra Wide Band*) (SALMAN, et al., 2010), funcionando na faixa de 3.1GHz a 4.8GHz e de 6.0GHz a 10.6GHz já aprovados pelo FCC (*Federal Communications Commission*), além das bandas de 249.6MHz a 749.6MHz, 780MHz e 950MHz com diferentes padrões de modulação.

A **camada de enlace** do IEEE 802.15.4-2003 apresenta diversas semelhanças à do padrão IEEE 802.11 (*Wi-Fi*) no que diz respeito às estratégias e controle de acesso ao meio e estruturas de quadros de dados.

A camada MAC propõe suporte para dois modos distintos:

- **Com sinalização** (*beacon-enabled*) – Neste modo são enviados periodicamente sinais do coordenador para sincronização e troca de informações da rede entre os dispositivos. Estes “*beacons*” formam o primeiro *slot* de tempo de uma estrutura chamada de *superframe*. A estrutura e formato de um *superframe* são definidos pelo coordenador da rede, que possui dois períodos distintos conforme mostrado na Figura 13.

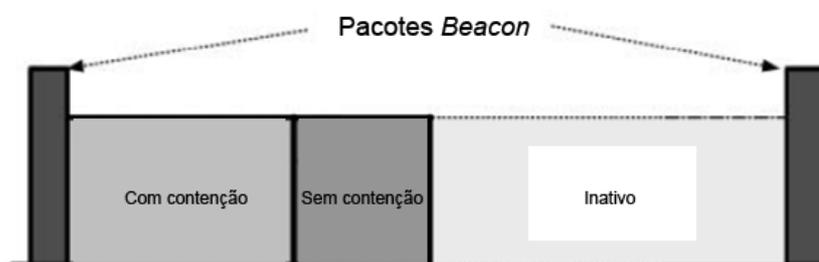


Figura 13 – Estrutura do *superframe*, adaptado de (SALMAN, et al., 2010).

Após o recebimento do *beacon*, inicia-se o período de acesso com contenção (CAP – *Contention Access Period*). Ou seja, neste período, se um dispositivo quiser se comunicar, ele deverá disputar o controle de acesso ao meio via CSMA/CA (*Carrier sense multiple access with collision avoidance*) (FEDERAL STANDARD 1037C, 1996) dividido em *slots* de tempo. Uma abordagem mais aprofundada do mecanismo de CSMA/CA aplicado às redes IEEE802.15.4 pode ser avaliada em (WANG, et al., 2011).

Terminado o período de contenção, caso existam dispositivos que necessitem de baixa latência na comunicação, o sistema entra no período livre de contenção com até sete *slots* de tempo garantidos para dispositivos.

- **Sem sinalização** (*non beacon-enabled*) – Neste modo de operação não há a utilização de *superframes*. Todo o acesso ao meio é feito com disputado via o mecanismo de CSMA/CA.

Em qualquer um dos casos, é escolhido um intervalo de tempo padrão (atualmente em 320 μ s) para sincronização entre os dispositivos nas unidades de tempo, chamado de *backoff*. O algoritmo é iniciado definindo-se NB em 0 (número de tentativas de retransmissão quando o canal estiver ocupado), CW (inicialmente em 2 – janela de contenção que representa o número de períodos de *backoff* a serem contados antes da transmissão, se o canal estiver disponível) e BE (número de repetições de *backoff* a serem executadas antes da transmissão, ocorrendo a cada iteração). Há ainda quatro atributos definidos para o funcionamento da rede, de acordo com a Tabela 1.

Tabela 1 - Variáveis para validação do CSMA/CA, adaptado de (SALMAN, et al., 2010; IEEE Std 802.15.4, 2003).

Nome do atributo	Descrição	Valor Padrão
<i>macBattLifeExt</i>	Indica se há um extensor de bateria habilitado pelo coordenador da rede.	Falso
<i>aMaxBE</i>	O valor máximo que o expoente de <i>backoff</i> pode ter no algoritmo CSMA/CA.	5

<i>macMinBE</i>	O valor mínimo que o expoente de <i>backoff</i> pode assumir em CSMA/CA. Se este valor for 0, o algoritmo de prevenção de colisões é desabilitado na primeira iteração.	3
<i>macMaxCSMABackoffs</i>	O número máximo de <i>backoffs</i> que o algoritmo vai tentar antes de assumir a falha do canal.	4

De posse dos dados da Tabela 1, é proposto (SALMAN, et al., 2010) que o valor de BE identifique o atributo *macBattLifExt*; e caso seja verdadeiro, BE assume um valor menor que *macMinBE*.

Em um segundo passo, o algoritmo aguarda por um tempo randômico entre $[0, 2^{BE} - 1]$. Quando este período expira, é então executado o procedimento de CCA (*clear channel assessment*) para verificar se há algum outro dispositivo transmitindo dados no meio. Caso o canal esteja livre, o dado não é transmitido imediatamente, e aguarda até que seja executado CCA por CW vezes, decrementando-o, até que CW seja zero, quando o pacote é então enviado. No caso do canal apresentar-se ocupado, o algoritmo é repetido com o incremento (+1) da variável BE, sem que ele exceda o valor de *aMaxBE*. Estas interações são executadas NB vezes, e, caso o canal se encontre ocupado para transmissão, a operação falha.

Diversas tentativas de implementação (HAN, et al., 2010; TUDOSE, et al., 2011; LUAN, et al., 2010) colocam sensores *Zigbee* e 6loWPAN em diversas redes de sensoreamento, gerando testes de conectividade, interfaces de controle, inclusive com iniciativas de conectividade via celulares com interface gráfica para o sistema operacional *Android* do *Google* (Figura 14).

Baseado nos resultados iniciais obtidos (CAO, et al., 2008) e nas topologias apresentadas pelo padrão (IEEE Std 802.15.4, 2003) é possível inclusive calcular o gasto de energia dos nós, via protocolo de roteamento LEACH (*low energy adaptative clustering hieranrchy*) (HEINZELMAN, et al., 2002), agrupando as redes de sensores

em *clusters*, cada um com seu nó pai, e levando todas as informações até o coordenador e *gateway* da rede, chamado de *sink*.

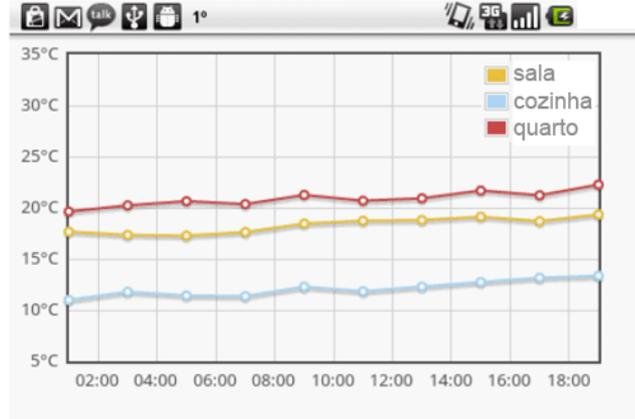


Figura 14 – Interface para celular do *Android*, adaptado de (TUDOSE, et al., 2011).

Utilizando LEACH, foi observado (CAO, et al., 2008) que a rede irá consumir a menor quantidade de energia para transmitir informações dos sensores na rede para seus nós pai, que redirecionarão as mensagens aos nós coordenadores, de acordo com as equações 2.5, 2.6 e 2.7.

Seja:

$$E_{pai} = k(E_{estático} + \epsilon_{amp}d_{pai_sink}^2) + \left(\frac{T}{N} - 1\right)kE_{estático} + \left(\frac{T}{N} - 1\right)k(E_{estático} + \epsilon_{amp}d_{receptor}^2) + k(E_{estático} + \epsilon_{amp}d_{receptor}^2) - 1 \quad (2.5)$$

Sendo ϵ_{amp} o ganho do sinal dado pelo amplificador, $d_{receptor}$ a maior distância que o nó pai pode cobrir até alcançar os sensores filho, T o número de nós, N o número de clusters formados, e k o número de *bytes* de dados transmitidos. $E_{estático}$ é a energia gasta pelos circuitos de transmissão e d_{pai_sink} é a distância do nó pai ao coordenador da rede e *gateway* (*sink*).

Analogamente, é demonstrado que a energia gasta por um nó que não é pai:

$$E_{n\grave{a}o_pai} = Nk(E_{est\acute{a}tico}) + k(E_{est\acute{a}tico} + \varepsilon_{amp}d_{receptor_pai}^2) \quad (2.6)$$

Com as premissas descritas (CAO, et al., 2008), e as equações 2.5 e 2.6, temos que:

- $E \left[d_{receptor_pai}^2 \right] = \frac{M^2}{2\pi N}$
- $\frac{T}{N} - 1 \approx \frac{T}{N}$ (na exaustão de energia)
- $E_{cluster} = E_{pai} + \frac{T}{N}E_{n\grave{a}o_pai}$
- $E_{total} = N \cdot E_{cluster}$

Desta forma, determina-se que o valor de N ótimo para o tamanho do *cluster* pode ser obtido através da relação:

$$N = \sqrt{\frac{T\varepsilon_{amp}M^2}{2\pi(T\varepsilon_{est\acute{a}tico} + \varepsilon_{amp}d_{broadcast}^2)}} \quad (2.7)$$

Assim, asseguramos que para qualquer valor $n \in \mathbb{N}$ assumido por N, teremos o menor consumo de energia para a rede de sensores e atuadores que utiliza esta arquitetura de roteamento.

Em outra abordagem, o autor (LANGHAMMER, et al., 2012) busca comparar o consumo de energia e o alcance *indoor* de uma rede IEEE802.15.4 com *chips* amplamente utilizados da empresa *Texas Instruments* na faixa de 868MHz, o CC1101 (CC1101, 2009), e um protocolo comercial europeu denominado *Wireless Metering-Bus* (wM-Bus).

Utilizando o modelo de consumo de energia da Figura 15 e os modelo EIRP (*equivalent isotropically radiated power*) de propagação de sinal e perdas, é

representada a relação sinal/ruído de um receptor de acordo com a equação 2.8 (LANGHAMMER, et al., 2012).

$$SNR = P_S|_{dBm} + G_S - L_B + G_R - P_T - F - I \quad (2.8)$$

Onde P_S é a potência irradiada de transmissão (em dBm), G_S e G_R representam os ganhos das antenas de transmissão e recepção, L_B é a atenuação do sinal causada pelas perdas do caminho (distância, paredes, pisos etc.), F é o ruído e I representa as perdas no receptor.

O ruído térmico P_T pode ser calculado pela expressão 2.9, onde B_{eff} representa a banda efetiva, o parâmetro T a temperatura ambiente (em Kelvin), e k é a constante de Boltzmann ($1.3806503 \times 10^{-23} \text{ m}^2 \text{ kg s}^{-2} \text{ K}^{-1}$).

$$P_T = 10 \cdot \log_{10}(k \cdot T \cdot B_{eff}) \quad (2.9)$$

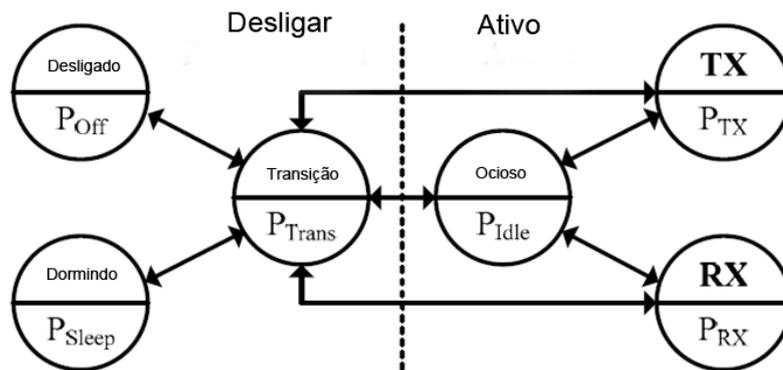


Figura 15 – Modelo de uso da energia proposto, adaptado de (LANGHAMMER, et al., 2012).

Utilizando pacotes com tamanho de 32 bytes, foi feita uma comparação em relação ao consumo de energia e alcance entre as duas tecnologias. Foi observado (LANGHAMMER, et al., 2012) que, o IEEE 802.15.4 consegue aceitar pacotes com

uma atenuação de até 9.2dB maior que os recebidos pelo *wM-Bus*, devido à funcionalidade de ARQ (*automatic repeat request*), presente na camada de enlace, que envia o pacote por até três vezes consecutivas, até que seja recebido um pacote do tipo ACK (*acknowledgement*) confirmando o recebimento, além das melhorias na camada física e modulação do sinal, características que garantem uma maior sensibilidade e imunidade à ruído ao conjunto transmissor e receptor.

Porém, de forma prática, o módulo CC1101 (CC1101, 2009) se mostrou viável para atenuações de até 80.5dB (suficientes para utilização em residências), com eficiência de energia até três vezes maior, que o módulo IEEE 802.15.4 com *Zigbee*, que recebeu dados com taxas de atenuação de até 93dB.

2.3. 6loWPAN

Como uma tentativa de reduzir problemas de interoperabilidade entre diferentes sistemas de comunicação, o IETF criou um grupo de trabalho (IETF 6lowpan, 2012) com o objetivo de habilitar em uma rede de sensores o padrão IPv6 (IPV6 CISCO, 2008), denominada 6loWPAN.

Em sua grande maioria, as redes de sensores sem fio possuem sérias limitações no que diz respeito às taxas de transmissão e tamanho máximo do pacote (MTU - *Maximum Transmission Unit*). Para manter a interoperabilidade entre sistemas padrão IP, torna-se necessária a criação de novos métodos para que sejam executadas conversões nos cabeçalhos do IP, criando seus correlatos reduzidos em tamanho, para que eles possam ser direcionados a uma sub-rede interna de sensores sem gerar incompatibilidades ou mudanças na padronização dos sistemas existentes.

Alguns trabalhos na literatura (DA SILVA CAMPOS, et al., 2011; HASBOLLAH, et al., 2010) buscam desenvolver *gateways* de interoperabilidade para comunicação, que exercem a função de tradutores dos pacotes IPv6 para o padrão 6loWPAN, na tentativa de, utilizando uma rede IPv6 comum, acessar sensores e atuadores baseados nestas tecnologias.

Isto implica em executar uma conversão bidirecional entre a rede interna de dispositivos e a rede externa que deseja acessá-los. Um dos métodos propostos para este tipo de conversão intitula-se HC1 (*Header Compression 1*) (HASBOLLAH, et al., 2010). Utilizando esta técnica, é possível transformar de 40 a 60 *bytes* do *overhead* IPv6 em 1 a 4 *bytes*, utilizando um compressor de dados, localizado na entrada de uma rede funcionando sobre a plataforma IEEE 802.15.4, e um descompactador na saída do *gateway*, conforme ilustrado na Figura 16.

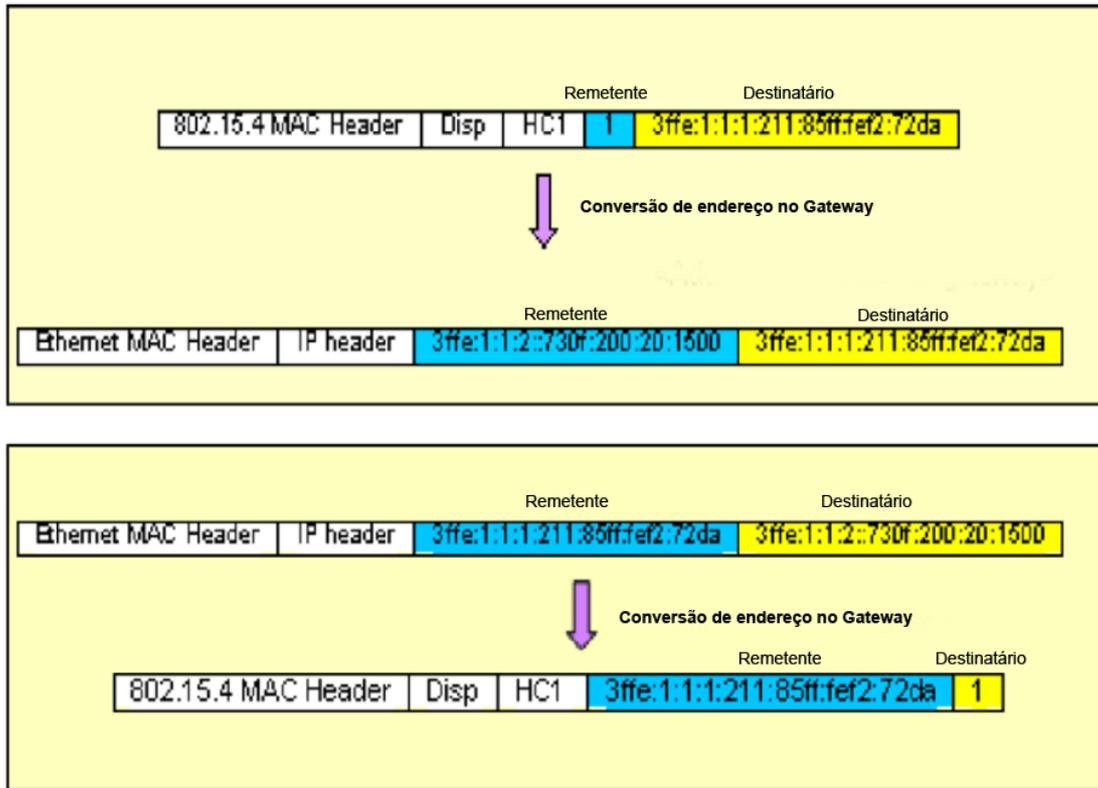


Figura 16 – Conversão bidirecional entre pacotes IPv6 e 6LoWPAN, adaptado de (HASBOLLAH, et al., 2010).

Informações tais como os campos de controle de versão (versionamento), tamanho do pacote, classe de tráfego e tipo de fluxo, podem ser removidas, sendo o restante dos dados, então, comprimido. Campos de remetente e destinatário são compactados utilizando-se uma tabela de conversão de endereços em memória do *gateway*. Desta forma, os endereços curtos de um sensor passam a ter apenas 2 bytes (16 bits) de informação. Todo o cabeçalho *Ethernet* é removido também, guardando-se apenas estes dados em referência no *gateway*.

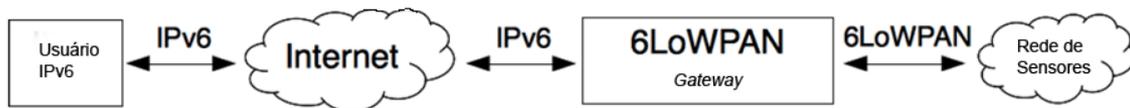


Figura 17 – Intercomunicação entre clientes IPv6 e rede de sensores sem fio (WSN), adaptado de (DA SILVA CAMPOS, et al., 2011)

Algumas iniciativas (YIBO, et al., 2011) procuram avaliar as diferentes pilhas de protocolos 6LoWPAN já implementadas no meio acadêmico e comerciais. São elas:

- *UC Berkeley (B6lowpan, BLIP);*
- *Jacobs University (6lopancli);*
- *SICS;*
- *Sensinode;*
- *Jennic.*

Todas as pilhas de protocolo são baseadas no modelo de camadas da Figura 18, onde são utilizados nas camadas física e de enlace, o protocolo IEEE 802.15.4 e, para a camada de rede o IPv6. Na camada de transporte são utilizados UDP e ICMP. Vale ressaltar que a padronização IEEE 802.15.4, utilizada nas camadas físicas e de enlace de uma rede 6LoWPAN, possui somente 2 modos de comunicação: *unicast* e *broadcast*, trazendo problemas de interoperabilidade com sistemas IPv6 *multicast*.

Todas estas arquiteturas são baseadas em microcontroladores de diversos fabricantes em modelos comerciais, tais como AVR, 8051, MSP430, STM32 e JN5139 (YIBO, et al., 2011) com pelo menos 4KB de memória RAM e 32KB de *flash*.

Protocolos de aplicação		Aplicação
UDP	ICMP	Transporte
IPv6		Rede
LoWPAN		
IEEE 802.15.4 MAC		Enlace
IEEE 802.15.4 PHY		Física

Figura 18 – Estrutura básica de camadas do 6LoWPAN, adaptado de (YIBO, et al., 2011).

Também foi observado que há iniciativas (MUKHTAR, et al., 2008; CHOI, et al., 2009) para desenvolver estratégias de gerenciamento de redes em 6loWPAN. Desta forma, podem ser medidos diversos parâmetros de performance e qualidade da rede, procurando reduzir o tráfego de mensagens de controle necessárias para executar este gerenciamento.

Como uma medida para avaliação da performance da rede, foi observado (MUKHTAR, et al., 2008) que a cada salto adicional em uma rede 6loWPAN, são adicionados, em média, 15.75ms de *delay* no envio das mensagens, como observa-se na Figura 19.

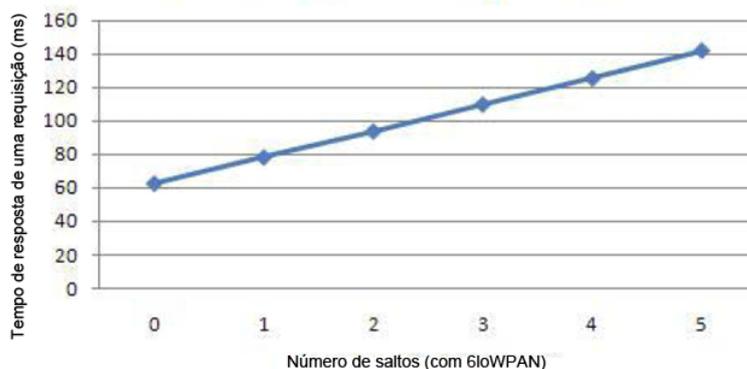


Figura 19 – Tempo de resposta obtido por número de saltos em uma rede 6loWPAN, adaptado de (MUKHTAR, et al., 2008).

Também foi avaliada uma comparação entre a resposta de pacotes SNMP (*Simple Network Management Protocol*) de uma rede IPv6 tradicional com o protocolo especificado para redes 6loWPAN, e, nesta pesquisa (CHOI, et al., 2009), foi percebida uma melhoria expressiva no volume de tráfego de dados gerados e nas perdas de pacote por *overflow*, ou erros de canal, conforme dados da Tabela 2.

Tabela 2 – Comparação da média de mensagens por 10 minutos de SNMP, adaptado de (CHOI, et al., 2009)

Tipo de Tráfego	SNMP IP	SNMP 6LoWPAN	Taxa
Mensagens Enviadas	450,9	254,8	56,5%
Pacotes Encaminhados	118,4	66,9	56,5%
Perdas por <i>overflow</i> de <i>buffer</i>	4,9	2,8	57,14%
Perdas por problemas no canal	1,5	0,8	53,33%

Desta forma, observa-se que o universo de trabalhos científicos dentro do universo das redes de sensores sem fio mostra um cenário extremamente promissor, com múltiplas aplicações, porém, com muito esforço no desenvolvimento futuro ainda a ser executado, de forma a torná-la uma tecnologia amadurecida e tornar-se realidade em grande parte das residências, escritórios, e até mesmo indústrias.

2.4. IEEE 802.11

Embora amplamente citada em publicações, conforme visto na Seção 2.2, nem toda a comunidade científica acredita que a tecnologia IEEE 802.15.4, com *Zigbee* ou outra camada de aplicação, seja a melhor solução para aplicações em *smart grid*.

Observa-se que alguns autores propõem (JIAN, et al., 2011; LI, et al., 2011) a utilização do padrão *WiFi*, como é conhecido popularmente o IEEE802.11, como a melhor alternativa disponível no mercado para o *smart grid*, trazendo maior largura de banda, maior cobertura de sinal, fácil expansão, boa relação custo/benefício, excelente nível de maturidade, além de maior robustez e segurança.

São citadas (LI, et al., 2011) as justificativas e pontos principais para a adoção do *WiFi*:

- **Maior taxa de transmissão de dados** – Com taxas de transmissão de até 300Mbps, o protocolo *WiFi* possui um *delay* menor que o IEEE802.15.4, trazendo um sistema de comunicação com maior capacidade para trabalhar em sistemas de tempo real e evitar o uso de algoritmos complexos para solucionar problemas de congestionamento do tráfego.
- **Capacidade transmitir e ultrapassar obstáculos (NLOS – *Non-Line-of-sight*)** – O autor (LI, et al., 2011) indica que o padrão IEEE802.11 possui características intrínsecas que o permitem obter um melhor desempenho para superar obstáculos físicos, deixando-o em posição de superioridade em relação a outros protocolos de redes sem fio. Tal capacidade não está presente em sistemas com *ZigBee*, desta forma, a transmissão poderá ser interrompida quando houver pontos de interferência destrutiva entre os sinais refletidos por objetos e paredes no uso da plataforma em ambientes *indoor*.
- **Área de cobertura** – Enquanto os rádios *ZigBee* possuem cobertura de 10 a 75m, os dispositivos *WiFi* possuem coberturas de 100-300m o que faz com que ele possa ser utilizado não só em residências, mas como na infraestrutura completa de um edifício.
- **Custo x Benefício** – Com a adoção em massa da tecnologia *WiFi* para dispositivos móveis e o desenvolvimento de microcontroladores integrados

com rádios *WiFi* em larga escala, é possível que em um futuro próximo esta tecnologia possa vir a competir com o baixo custo da tecnologia *ZigBee*.

- **Fácil Expansão** – Cada nó de uma rede *WiFi* pode suportar diversas conexões simultâneas. Desta forma, nós sensores *WiFi* poderão se comunicar com eletrodomésticos, serviços de IPTV, serviços de telemedicina (sensores intracorporais, por exemplo) e outras tecnologias de sensores utilizando uma única infraestrutura.
- **Robustez** – Com tecnologias de rede em malha embarcadas na pilha de protocolo, o *WiFi* é uma tecnologia madura e confiável para tratar possíveis falhas e recuperação de erros do que outras tecnologias de rede sem fio disponíveis no mercado.
- **Baixa perturbação *interlinks*** - Os diversos nós de uma rede *WiFi*, quando atuando em modo de rede em malha, podem habilitar tecnologias “*multi-interface multi-channel*” (KYASANUR, et al., 2004), que permitem aos dispositivos em áreas adjacentes utilizarem canais ortogonais para reduzir a interferência entre as portadoras e os sinais transmitidos, permitindo ao sistema ampliar sua capacidade de transmissão de dados.

Além disto, foi observado (LI, et al., 2011) que o IEEE 802.11 possui uma maior maturidade e interoperabilidade entre diferentes sistemas de redes de sensores.

Assim, a partir de um modelo conceitual para *smart grid* (e outras tecnologias que utilizam serviços de medição e controle), foi proposta a utilização de quatro classes distintas (JIAN, et al., 2011) de equipamentos conectados via *WiFi*, como é possível observar na Figura 20.

- **Classe 1** – É a camada responsável pelos medidores de energia, gás e água. Estão incluídos nesta rede os sensores individualizados e equipamentos que serão capazes de se comunicar por *WiFi*, tais como eletrodomésticos e outros equipamentos;

- **Classe 2** – É a camada que inclui os sensores e medidores mais próximos aos concentradores. Os medidores centrais de cada residência e pontos de serviço irão se interconectar dentro desta rede. Os dados gerados são enviados aos concentradores mestre, para a transmissão da informação a outros equipamentos nas camadas superiores;
- **Classe 3** – É a camada que inclui o concentrador principal dos prédios e outros concentradores. Os equipamentos que irão trocar informações nesta camada da rede deverão utilizar preferencialmente a topologia estrela, entendendo que eles estarão uniformemente e estrategicamente espalhados e distribuídos;
- **Classe 4** – É a camada que faz a comunicação entre o concentrador principal e o centro de gerenciamento municipal. Neste caso o autor cita a utilização de redes 3G com VPN e até WiMAX como tecnologias de transmissão de dados para esta última camada.

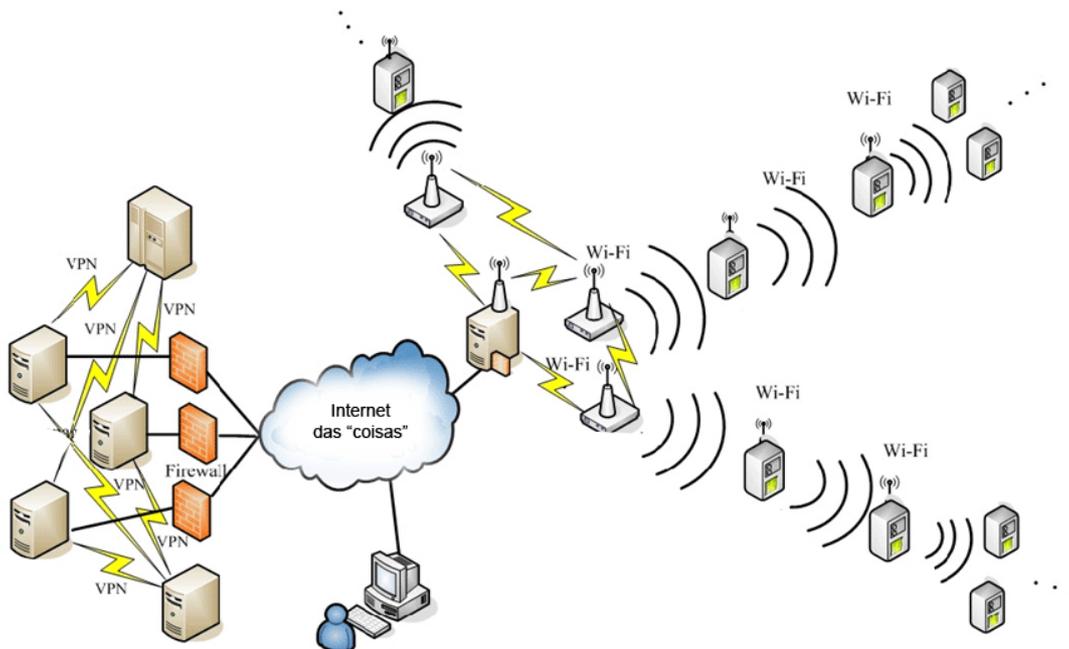


Figura 20 – Arquitetura proposta para redes de medição baseadas em IEEE802.11, adaptado de (JIAN, et al., 2011).

Há também outros estudos que procuram avaliar a utilização do IEEE 802.11 como *link* de dados para longa distância no *smart grid* (TANAKA, et al., 2011). Desta forma, em locais onde não há infraestrutura de telecomunicações, por fibras óticas ou cabos, disponível para esta transmissão de dados o *WiFi* pode ser uma alternativa viável, conforme proposto na Figura 21.

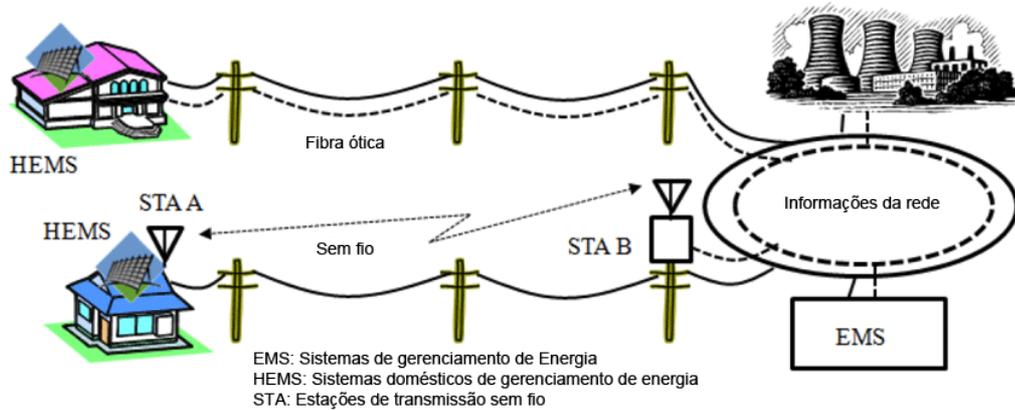


Figura 21 – Utilização de link IEEE 802.11 em áreas de difícil acesso, adaptado de (TANAKA, et al., 2011).

Os resultados obtidos demonstram que, operando em modo DCF (*Distributed Coordination Function*), e com taxas de transmissão de 6Mbps (*delay* de propagação de $40\mu\text{s}$ - o que corresponde a uma distância de 12km), a degradação da taxa máxima do sistema caiu em média apenas 16.8%, dando suporte tecnológico para que o *WiFi* possa vir a ser utilizado para o *smart grid* até mesmo para redes com distâncias acima de 10km.

2.5. PLC

Com uma abordagem diferenciada, alguns autores (SAUTER, et al., 2011) propõem a utilização da tecnologia PLC (*Powerline Communication*). Desta forma, acreditam conseguir um alcance maior de comunicação entre os dispositivos envolvidos, garantindo conexão permanente, e maior confiabilidade e segurança, quando comparados a sistemas sem fio.

Como ponto negativo do sistema são listados o alto grau de complexidade técnica exigida para seu funcionamento e o seu alto custo, algo que pode ser superado com arquiteturas que vêm sendo propostas, por exemplo, por pesquisas europeias como o REMPLI (SAUTER, et al., 2011), mostrado na Figura 22.

Nesta pesquisa, o autor propõe a utilização de um túnel virtual (VPN), via redes de pacotes IP, rodando em cima de uma rede PLC, onde são suportadas, nas camadas de transporte e de rede, a fragmentação e remontagem dos pacotes. Estes estudos demonstram que pacotes IP (normalmente com 1500 *bytes* de MTU), sofrendo fragmentação e remontagem para pacotes de 64 *bytes*, conseguem transferir dados a taxas de 139.7Kbps em redes de banda estreita e, com pacotes maiores foram alcançados até 250Kbps máximos de *throughput* (SAUTER, et al., 2011).

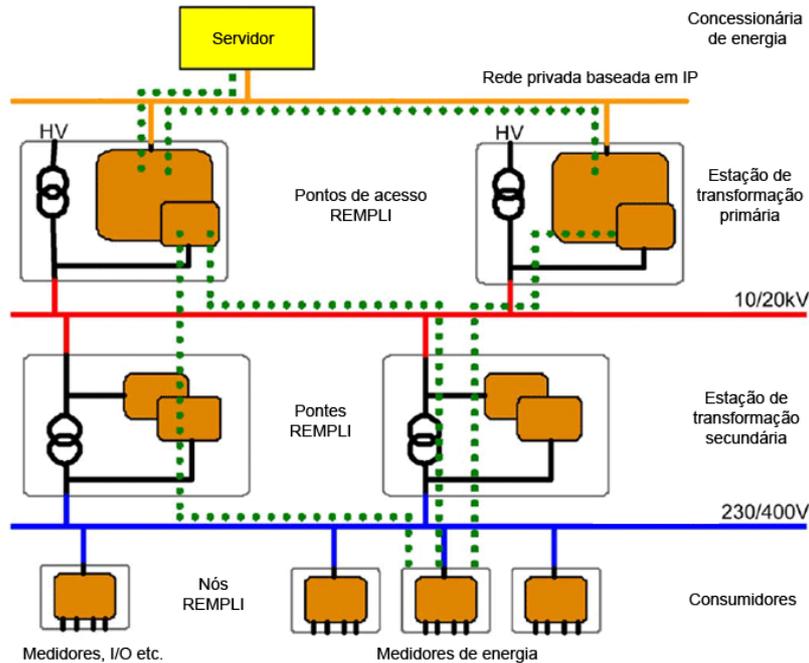


Figura 22 – Protocolo REMPLI como arquitetura de *smart grid*, adaptado de (SAUTER, et al., 2011).

Em outra abordagem utilizando equipamentos PLC (QINGYANG, et al., 2009), é descrita uma experiência prática para sistemas de telemedição automatizada em banda larga (BPL – *Broadband Powerline*) na China.

Como citado anteriormente no Capítulo 2, foi verificado nesta publicação que equipamentos BPL são raramente utilizados em sistemas domésticos por seu alto custo, sendo os sistemas de banda estreita (<2400bps) os mais utilizados para esta finalidade. Todavia, os resultados experimentais obtidos (QINGYANG, et al., 2009) demonstram alta superioridade na velocidade e performance dos sistemas de banda larga, o que pode, mesmo com custos mais altos, acelerar sua adoção em massa com o apelo dos requisitos de sistemas *smart grid* com medição em tempo real.

Em outra publicação (LIU, et al., 2010), os autores buscam avaliar as diferentes tecnologias de BPL existentes com relação à taxa máxima de dados, estabilidade e imunidade a ruídos, mesmo com outros equipamentos elétricos em funcionamento ligados à rede. Alguns resultados estão apresentados na Tabela 3.

Tabela 3 – Tabela sumário dos resultados em diferentes *chipsets* BPL adaptado de (LIU, et al., 2010).

Dispositivo	Vazão máxima (Kbps)	Estabilidade
Intellon 14Mbps	5297.7	Boa
Xeline 24Mbps	6256.6	Normal
Yitran 24Mbps	6180.4	Ruim
Intellon 200Mbps	33216.6	Normal
DS2 200Mbps	49598.4	Boa

Após a implantação destes dispositivos em um projeto piloto, utilizando equipamentos da classe de 200Mbps, foram coletados todos os dados em apenas um minuto, em testes práticos com 378 residências de uma comunidade, o que mostra a alta capacidade do sistema (LIU, et al., 2010).

Porém, novamente, o alto custo da solução ainda é citado como um impeditivo para sua adoção em massa, sendo os dispositivos de banda estreita ainda os que conseguem manter um preço competitivo para tal finalidade.

Como a possibilidade para adoção e utilização de sistemas BPL e sua consequente baixa no custo com o fator de escala, foi relacionado (LIU, et al., 2010) seu uso na implantação de sistemas *triple-play*, onde serviços de televisão, *internet* e telefonia compartilham a mesma infraestrutura física existente (no caso os próprios condutores de energia), podendo vir a causar uma produção massificada dos *chips* e a consequente baixa no custo total da solução para uma adoção em *smart grid*.

2.6. Outros Sistemas

No que diz respeito ao estado da arte em *smart grid*, há ainda alguns artigos na literatura que citam a utilização de sistemas híbridos PLC com redes sem fio (RICCI, et al., 2011) e alternativas que utilizam simuladores via software, para a estimação do consumo de energia elétrica, obtendo resultados muito próximos do real, quando comparada a um sistema completo de medição distribuída (SEUNGHYUN, et al., 2010). Nesta plataforma de simulação foram avaliadas as formas de onda de potência de diversos equipamentos em uma residência, tais como televisores, computadores, geladeiras, carregadores de celular etc., e, a partir destes dados, foi gerado um modelo computacional, dentro de um cenário residencial, para avaliar o consumo, sem utilizar qualquer *hardware*.

Este *software* propõe tornar-se uma plataforma preditiva do consumo individual de cada equipamento instalado ao longo do tempo, sem a utilização de dispositivos eletrônicos.

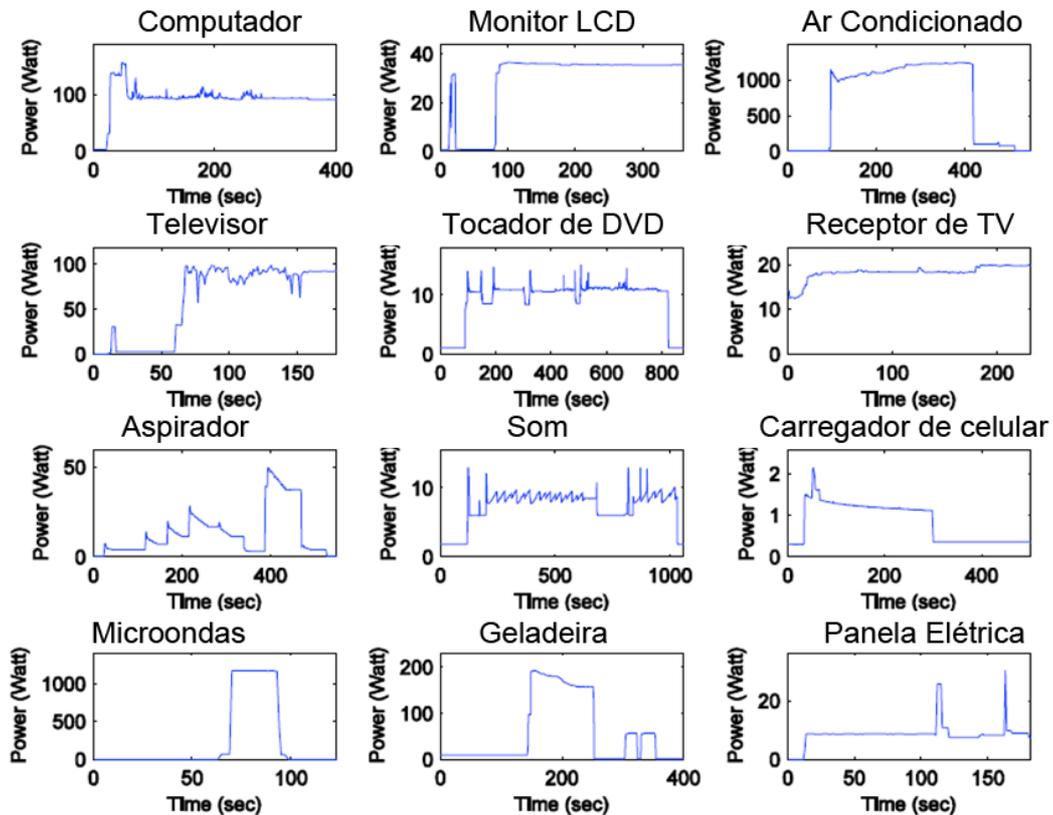


Figura 23 – Padrões de consumo de energia para equipamentos eletroeletrônicos, adaptado de (SEUNGHYUN, et al., 2010).

Desta forma, após um período de treinamento do sistema com insumos de dados individuais dos equipamentos reais, foram criados cenários onde medidores eletrônicos eram comparados com o simulador, propondo validar sua operação.

A partir deste experimento (SEUNGHYUN, et al., 2010), foi obtida uma correlação média de 97.3% nas formas de onda de consumo, comprovando que esta alternativa também pode ser utilizada para prever em parte ou totalmente, um consumo residencial.

Porém, seu uso é limitado. Esta arquitetura não consegue sinalizar, por exemplo, um problema operacional em um equipamento, caso haja discrepância entre a simulação e a medição global aferida pela concessionária de energia. Esta informação só estaria disponível após comparação com a cobrança mensal emitida.

Desta forma, observa-se de forma consolidada, uma grande diversidade nos projetos envolvendo diferentes tecnologias com potencial para serem utilizadas num cenário em *smart grid*, além de entender mais sobre os desafios envolvidos no processo de modernização dos atuais sistemas de medição.

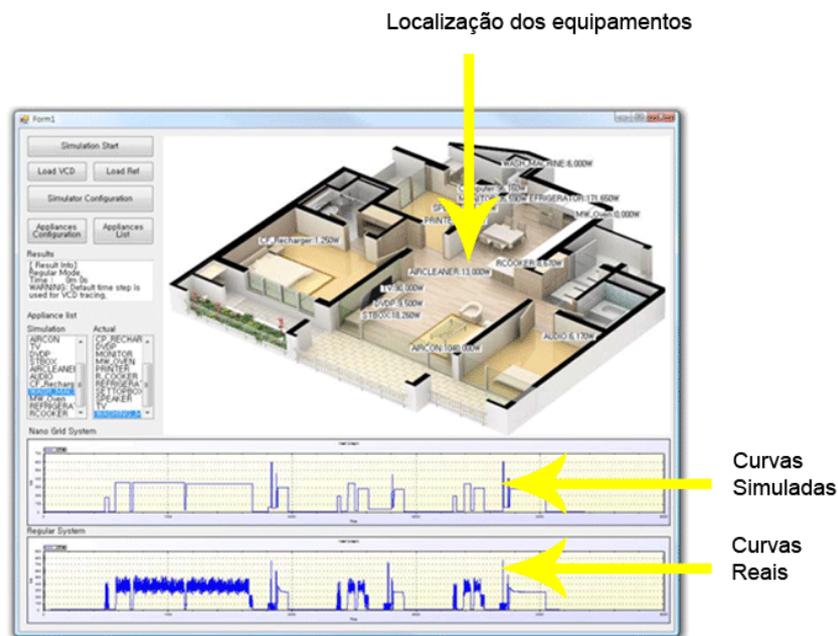


Figura 24 – Interface gráfica do simulador, adaptado de (SEUNGHYUN, et al., 2010)

Há ainda uma grande pressão governamental e da sociedade para evitar desperdícios. Com isso, sistemas e plataformas tecnológicas que propiciem um aumento

da eficiência energética tornam-se prioridade em um cenário de escassez de recursos para a produção de energia elétrica.

Assim, após analisar as referências bibliográficas expostas até o momento, foram ponderadas as características necessárias para a realização deste trabalho que pudessem oferecer baixo custo, visando adoção em massa, flexibilidade de instalação, reduzindo as barreiras de entrada para o mercado e, principalmente, oferecendo ferramentas de controle e automação que entreguem aos usuários e clientes de energia elétrica uma alternativa para obter ferramentas inovadoras e acessíveis para gerenciar seus recursos energéticos, reduzindo custos desnecessários e obtendo informações para conhecer melhor seu padrão de consumo e aumentar sua eficiência.

3. Projeto do sistema

3.1. Visão geral

Este trabalho foi elaborado com o intuito de avaliar tecnicamente, na esfera teórica e prática, uma arquitetura para equipamentos de medição inteligente, que se beneficie de novos recursos em telecomunicações, em um equipamento economicamente viável para adoção em massa, visando um cenário onde poderá ser embarcado em tomadas, disjuntores e medidores, tornando-se parte de uma única rede residencial, conforme ilustrado na Figura 25.

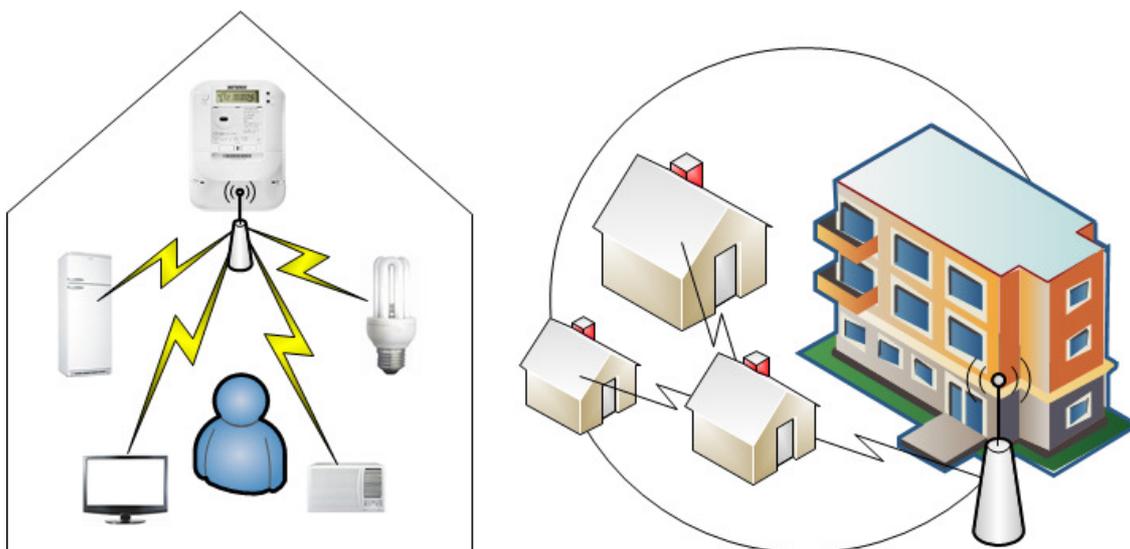


Figura 25 – Projeto de um sistema de comunicação capaz de habilitar o desenvolvimento de uma rede interna de medidores inteligentes.

Partindo deste princípio, dispositivos eletrônicos de baixo custo deverão ser priorizados no processo de escolha, o que, invariavelmente trará à plataforma um baixo poder computacional, de forma a viabilizar o trabalho e sua consequente adoção como produto no mercado.

Este projeto de hardware contempla uma fonte, um microcontrolador capaz de efetuar a medição de energia, um dispositivo emissor e receptor (*transceiver*) de radiofrequência e a instrumentação necessária para o condicionamento dos sinais de

tensão e corrente da rede, além de um dispositivo para acionamento da carga a ser controlada (Figura 26) e, por se tratar de um *hardware* complexo, toda a tecnologia embarcada do sistema de medição, condicionamento de sinais e alimentação (fonte), foi desenvolvido pela empresa “Powersave Soluções de Tecnologia em Energia LTDA”, parceira do Laboratório LEMT da COPPE/UFRJ e, atualmente, uma empresa nativa da Fundação COPPETEC, não fazendo, portanto, parte da discussão deste trabalho.

Assim, o tema central desta dissertação baseia-se, exclusivamente, no desenvolvimento de uma plataforma de comunicação a ser utilizada por estes dispositivos, além de resultados práticos e por simulação, validando os benefícios obtidos do sistema atual de medição frente a dispositivos comerciais.

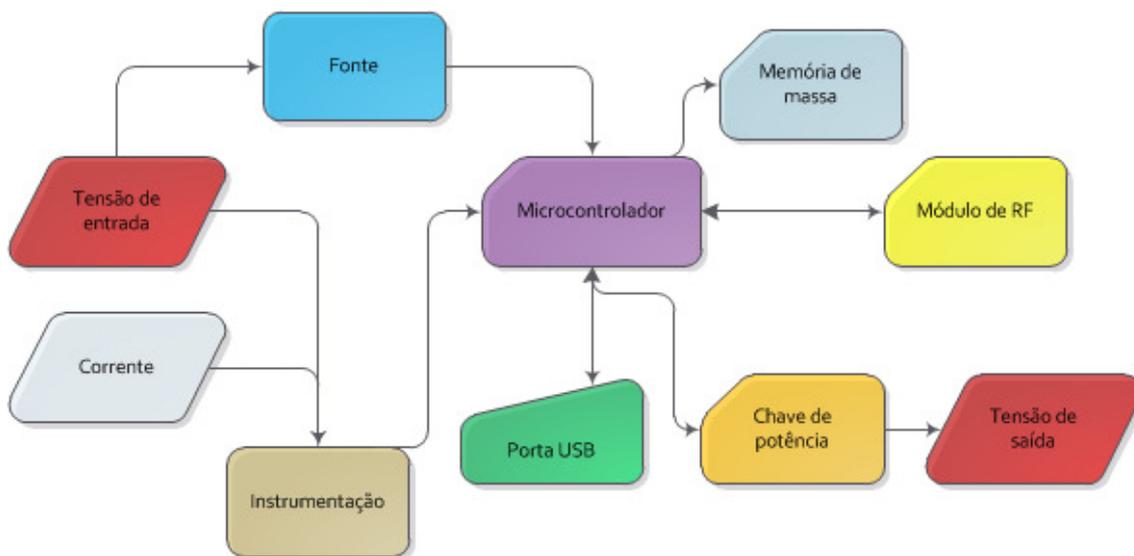


Figura 26 – Esquema proposto para o dispositivo que irá suportar a pilha do protocolo proposto.

Vale ressaltar que não faz parte do escopo deste trabalho o desenvolvimento de um *gateway* de comunicação capaz de suportar diversas tecnologias IP para executar tarefas à longa distância ou mesmo através de dispositivos *smartphones*, em uma rede celular. Este enfoque poderá ser executado em propostas de trabalho futuro baseadas nesta arquitetura de protocolo desenvolvida.

O dispositivo final possuirá a capacidade de executar a telemedição intervalar, com armazenamento de dados em memória de massa, e incluir funcionalidades para acionamento remoto e programação de eventos baseados em data e hora, além da capacidade de habilitar uma rede em malha sem fio para qualquer dispositivo, evitando-se classificações diferenciadas entre os dispositivos, conforme explicitado na Seção 2.2.

3.2. Projeto da plataforma

Tendo em vista todas as funcionalidades propostas para o projeto dispostas na Figura 27, foram feitas algumas restrições aos requisitos mínimos do sistema, no que diz respeito à segurança dos dados transmitidos pelo dispositivo, a taxa mínima de transmissão dos pacotes de dados, ao valor máximo de memória disponível para o armazenamento de dados históricos, ao tamanho máximo do pacote de dados a ser transmitido, e ao número máximo de dispositivos suportados pela rede ao mesmo tempo.

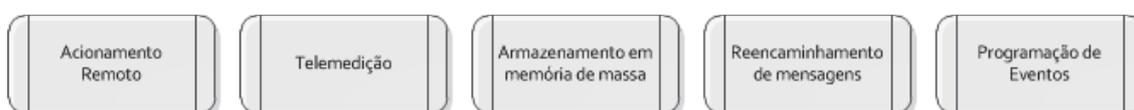


Figura 27 – Esquema básico de ações e funcionalidades que devem ser suportados pelo sistema.

Assim, inicialmente serão examinados os requisitos do *hardware* de comunicação RF (rádio frequência), descrito pela Figura 26. A partir desta definição de *hardware*, será possível propor um sistema de comunicação eficaz, capaz de ser implementado utilizando a totalidade de recursos computacionais existentes com a plataforma escolhida.

3.2.1. *Hardware* de comunicação

Para o projeto de *hardware* foram avaliadas diversas arquiteturas e configurações de produtos disponíveis no mercado antes de inferir qual das plataformas se enquadraria nos requisitos mínimos do sistema a ser desenvolvido. Desta forma, dentro das soluções existentes, seria ideal utilizar componentes com performance satisfatória e compatível com as tarefas de *smart grid*, além do baixo consumo de energia, visto que é um equipamento com o intuito de aumentar a eficiência energética, e com custo médio de mercado que não inviabilize sua adoção e utilização em massa.

De fato, este tipo de tecnologia deve, no mínimo, ter seu retorno financeiro (*payback*) garantido aos usuários, para que os estimulem a adotarem como solução. Para

tal, é necessário avaliar seu mercado potencial. Como uma das formas de mensurar o seu tamanho, foram adotados dois grandes grupos, sendo o primeiro, ‘a quantidade de dispositivos eletroeletrônicos novos fabricados’, e os ‘atualmente já instalados’ (aguardando modernização - *retro-fit*).

Para estimar a quantidade de novos dispositivos eletroeletrônicos, foram utilizados os dados da Associação Brasileira dos Revendedores e Distribuidores de Materiais Elétricos (ABREME, 2006), que estima a venda de quase R\$3Bi em dispositivos elétricos novos (somente disjuntores, tomadas etc.) vendidos em todo o território brasileiro em 2006.

Com números tão grandiosos, percebe-se então que, para a concepção da plataforma, qualquer dólar economizado em componentes eletrônicos nesta escala, poderá potencialmente economizar centenas de milhões de reais anuais, quando a quantidade de dispositivos comercializados no mercado alcançarem estas dimensões.

Assim, dentre estas soluções de mercado disponíveis em mercado, foram analisados três subgrupos de arquiteturas básicas e suas especificações, conforme descrito na Tabela 4. Desta forma, os sistemas de rádio frequência com taxas de transferência de até 500Kbps apresentaram o menor custo total da solução, por consumirem pouca energia, menor complexidade computacional requerida, e com isso menor custo global da solução.

Foram avaliadas também configurações de sistemas utilizando microcontroladores em separado dos *chips* de rádio (dois circuitos integrados distintos) e, em outra configuração, utilizando módulos únicos compostos por microcontroladores e sistemas de rádio embutidos em um único encapsulamento.

Dentre os microcontroladores de grandes fabricantes já renomados no mercado, foi destacado o modelo da família MSP430 da *Texas Instruments* (MSP430FE427A, 2008) por já possuir um *hardware* interno projetado para medição de energia com correção automática de temperatura, facilitando o desenvolvimento e aprimorando a qualidade da medição do consumo de energia elétrica.

Para os *chips* de rádio, foram apontados para a avaliação duas soluções de baixo consumo de energia. O *chip* CC2420 (CC2420, 2008) baseado em IEEE802.15.4 e o *chip* CC2500 (CC2500, 2007), solução proprietária para sistemas sem fio de baixo

consumo de energia com suporte a *drivers* escritos pelo usuário ou fornecidos pelo fabricante.

Tabela 4 - Comparativo entre as alternativas avaliadas de *hardware* para o projeto.

Descrição	CC2420+MSP430	ATZB-24-B0RTR	CC2500+MSP430
Fabricante	<i>Texas Instruments</i>	<i>Atmel</i>	<i>Texas Instruments</i>
Consumo de Energia	18.8mA(RX) 17.4mA(TX) @3.3V	19mA(RX) 18mA(TX) @3.3V	13.3mA(RX) 11.1-21.5mA(TX) @3.3V
Tecnologia e taxa de transferência	IEEE 802.15.4(2.4GHz) 250Kbps	IEEE 802.15.4(2.4GHz) 250Kbps	TI (2.4GHz) 1.2Kbps-500Kbps
Tamanho máximo do pacote	128 <i>bytes</i>	128 <i>bytes</i>	64 <i>bytes</i>
Valor (US\$) (DIGIKEY, 2012)	\$4.29+\$4.45 = \$8.47	\$15.51	\$1.86+\$4.45 = \$6.31
Especificações	32KB(<i>flash</i>) 1KB (RAM) 4ADC(16 <i>bit</i>) <i>EMETER</i> com/TS	128KB (<i>flash</i>) 8KB(RAM) 4ADC(10 <i>bit</i>)	32KB(<i>flash</i>) 1KB (RAM) 4ADC(16 <i>bit</i>) <i>EMETER</i> com/TS
Sensibilidade	-95dBm(@250KBaud)	-101dBm(@250KBaud)	-104dBm(@2.4KBaud) -89dBm(@250KBaud)
Potência Máx. TX	0dBm	3dBm	1dBm

Dentre os módulos de um único encapsulamento, foram avaliadas as soluções baseadas nos processadores da ATMEL para redes sem fio (ATMEL, 2012) por se tratar de uma empresa com boa reputação no mercado de microcontroladores e automação,

onde se baseia a linha *Arduino* (ARDUINO, 2012), buscando um fabricante alternativo para o projeto.

Na literatura podemos encontrar trabalhos com o intuito de projetar medidores eletrônicos, porém, em nenhum deles foi observada a hipótese da sua implantação como um **produto** a ser entregue ao **mercado** e, a avaliação de fato do impacto direto do custo desses equipamentos medidores na melhoria da eficiência energética.

Observa-se, por exemplo, que, em um destes trabalhos (HASAN, et al., 2011), foram utilizados microcontroladores da mesma família MSP430 avaliados na Tabela 4 e um módulo de comunicação *XBee* (DIGI, 2012), trazendo a pilha de protocolos *ZigBee* em um hardware compatível com o padrão IEEE802.15.4. Avaliando somente o custo do módulo *XBee* (\$19.00) (DIGIKEY, 2012), sem considerar o custo do microcontrolador, é possível avaliar que se trata de um trabalho de caráter unicamente acadêmico, visando somente a análise de funcionalidades, e que não possui ambições de se tornar um produto viável comercialmente, sendo adotado em grande escala.

Assim, de forma a avaliar técnica e economicamente a viabilidade do trabalho, deve-se buscar mensurar a quantidade de dados necessária para transmissão de dados (um dos fatores que, além do preço, pode definir o *hardware* da plataforma). Observa-se, então, que um pacote de dados deve possuir no mínimo:

- No caso de uma rede com estrutura estrela:
 - Cabeçalho
 - Tipo de requisição / resposta
 - Endereço do remetente
 - Endereço de destino
 - Identificador de rede
 - Dados contendo
 - Tensão RMS
 - Corrente RMS
 - Potência Média máxima do período
 - Consumo de energia

- Data e hora da gravação do evento
 - Tipo de evento (desligamento, liga/desliga, medição intervalar etc.)
 - CRC para correção de erro
 - E/ou Rodapé de protocolo
- No caso de uma rede com estrutura em malha:
 - Cabeçalho
 - Tipo de requisição / resposta
 - Endereço do remetente
 - Endereço de destino
 - Endereço do próximo salto
 - Endereço do agente transportador
 - Identificador de rede
 - Dados contendo
 - Tensão RMS
 - Corrente RMS
 - Potência Média máxima do período
 - Consumo de energia
 - Data e hora da gravação do evento
 - Tipo de evento (desligamento, liga/desliga, medição intervalar etc.)
 - CRC para correção de erro
 - E/ou Rodapé de protocolo

Como parâmetro de avaliação, toda a estrutura de dados deve conter:

- **float** (Tensão RMS) – 4 *bytes*
- **float** (Corrente RMS) – 4 *bytes*

- **float** (Potência) – 4 *bytes*
- **float** (Consumo de energia) – 4 *bytes*
- **struct** (Data e hora) – 5 *bytes*
- **struct/enum** (Tipo de Evento) - 1 *byte*

Desta forma, em termos de tráfego de dados, teremos, pelo menos, 22 *bytes* de informação a serem transmitidos. Avaliando o impacto causado pela inserção deste *overhead* nos protocolos existentes, citados no Capítulo 2, é possível estimar de forma segura que 64 *bytes* (o menor *buffer* dos equipamentos listados) são suficientes para transmitir todo o pacote da mensagem, incluindo a sobrecarga adicional do protocolo.

Nas especificações relacionadas a consumo de energia, pode ser observado na Tabela 4 que praticamente todos os módulos analisados competem em igualdade, com uma leve vantagem técnica para o chip CC2500, que possui o menor consumo de energia em RX ON (*receive mode on*), ou seja, valor que é consumido quando o dispositivo está ativo aguardando o recebimento de mensagens na maior parte do tempo.

Esta pequena diferença no consumo representa, em média, 30% a menos de energia consumida do que os seus competidores na maior parte do tempo, visto que o dispositivo passa pouco tempo útil em estado de transmissão de mensagens, porém aguarda quase todo o tempo em modo de recepção (RX ON).

Ao avaliarmos a taxa de dados útil com relação à utilização máxima do espectro, utilizando apenas um canal de transmissão dentro da faixa ISM (*Industrial, Scientific and Medical radio bands*) permitida (2400-2483MHz) temos que o padrão IEEE802.15.4 obedece a sequência de pacotes acordo com a Figura 28. Esta abordagem propicia reduzir o risco de colisões, embutindo períodos de silêncio SIFS (*Short Inter-Frame Space*) e LIFS (*Long Inter-Frame Space*) dentro da estrutura de um único pacote dados, reduzindo, em modo contínuo a taxa máxima teórica a ser atingida. Estes mecanismos procuram, com certa taxa de probabilidade, evitar o risco de colisão de pacotes.

Alguns modos de operação dos protocolos IEEE 802.11 e IEEE 802.15.4 seguem também o padrão de *superframe*, onde há disputa pelo meio em alguns instantes de tempo e, em outros, há a distribuição de *slots* de tempo aos dispositivos que necessitem de QoS (*Quality of Service*). Caso o número de dispositivos aumente, a utilização da

estrutura de pacotes sincronizados (com sinalização de *beacons*) com períodos sem disputa pode vir a ser vantajosa para evitar colisões (LATR E, et al., 2005).

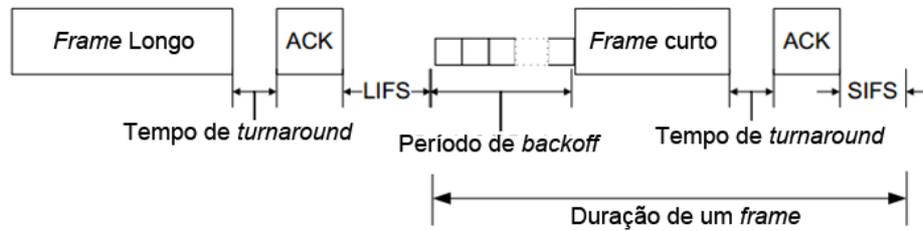


Figura 28 – Sequência de pacotes do IEEE 802.15.4, adaptado de (LATR E, et al., 2005).

Procurando obter dados teóricos mais aprimorados, observa-se uma avaliação da taxa de dados efetiva a ser transmitida, utilizando-se a estrutura do pacote da Figura 29. Desta forma, para calcular teoricamente a máxima taxa de dados (*throughput*) podemos utilizar a equação 3.1, onde x representa o número de *bytes* necessários para se transmitir um pacote.

$$TP_{max} = \frac{8.x}{delay(x)} \quad (3.1)$$

O *delay* da transmissão de dados pode ser obtido através da equação 3.2. Temos considerados nestes cálculos o período de *backoff* (T_{BO}), o tempo de transmissão do *payload* de dados com x *bytes* ($T_{frame}(x)$), o tempo de “*turnaround*” (T_{TA}) do sistema (tempo de execução gasto pelo processamento das tarefas executadas), o tempo de transmissão de um ACK (T_{ACK}) e o tempo de IFS ($T_{IFS}(x)$) que pode variar de 192 μ s (SIFS) a 640 μ s (LIFS).

Pela especificação (IEEE 802.15.4, 2012) quando a PDU (*Packet Data Unit*) da camada de enlace for menor ou igual à 18 *bytes* é utilizada SIFS, de outra forma é utilizada a temporização LIFS. Como temos pacotes sempre maiores que 22 *bytes*, conforme já avaliado, será sempre utilizado o valor (LIFS) para o cálculo teórico.

$$delay(x) = T_{BO} + T_{frame}(x) + T_{TA} + T_{ACK} + T_{IFS}(x) \quad (3.2)$$

O tempo de *backoff* (T_{BO}) pode ser obtido pela equação 3.3, onde é estipulado o número de *slots* (BO_{slots}) escolhido entre o intervalo $[0, 2^{BE} - 1]$, sendo BE é um expoente escolhido de valor mínimo 3; e o tempo de *backoff* ($T_{BO\ slot}$), dado pela especificação (IEEE 802.15.4, 2012) o valor de 320us.

Como valor médio do número de *slots* foi utilizado o valor de 3.5, obtido com pelo cálculo de valor médio $\frac{2^N-1}{2}$ onde N=3 (previamente avaliado em 2.2).

$$T_{BO} = BO_{slots} \cdot T_{BO\ slot} \quad (3.3)$$

De acordo com a Figura 29, podemos observar que o tempo de transmissão do pacote de dados ($T_{frame}(x)$) deve ser calculado através da equação 3.4. Caso haja a necessidade de utilização de reconhecimentos ACK (desejável para garantir que o pacote foi recebido pelo receptor), também deverá ser incorporado o tempo para a transmissão do ACK (T_{ACK}) conforme equação 3.5.

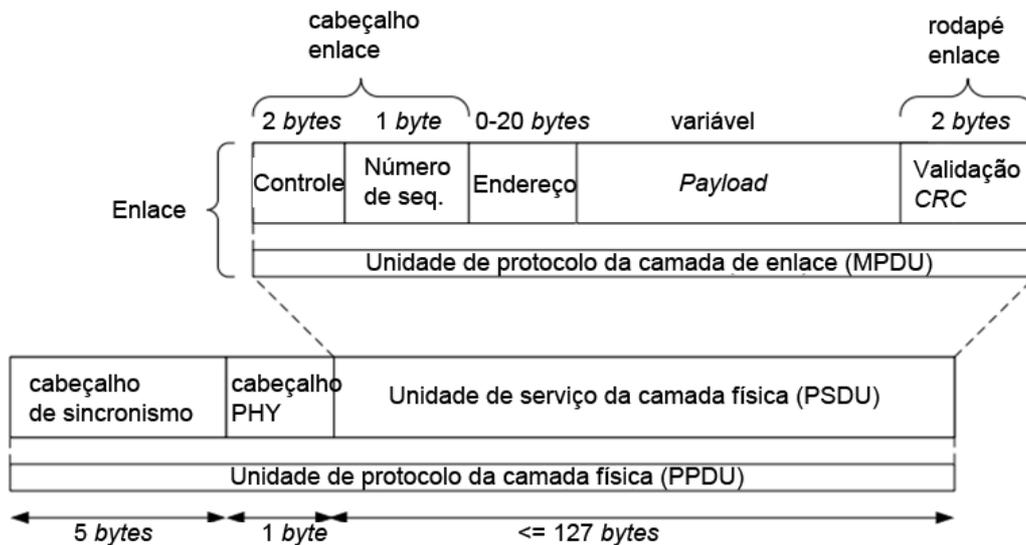


Figura 29 – Estrutura do pacote de dados do IEEE 802.15.4, adaptado de (LATR'E, et al., 2005).

$$T_{frame}(x) = 8 \cdot \frac{L_{PHY} + L_{MAC_Header} + L_{address} + x + L_{MAC_Footer}}{D_{rate}} \quad (3.4)$$

$$T_{ACK} = 8 \cdot \frac{L_{PHY} + L_{MAC_Header} + L_{MAC_Footer}}{D_{rate}} \quad (3.5)$$

Onde:

- L_{PHY} – (6 bytes) Tamanho da camada física e do cabeçalho de sincronização
- L_{MAC_Header} – (3 bytes) Tamanho do cabeçalho da camada de enlace (MAC)
- $L_{address}$ – (4-10 bytes) Tamanho do endereço MAC (2-8 bytes) associado ao endereço da PAN (*Personal Area Network*) (2 bytes)
- L_{MAC_Footer} – (2 bytes) Tamanho do rodapé da camada de enlace
- D_{rate} – Taxa de dados (250Kbps)

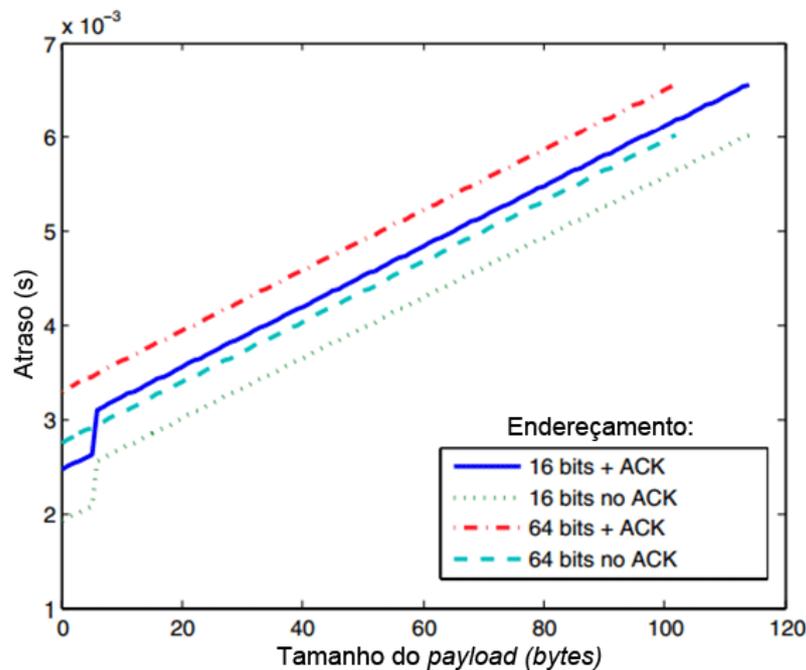


Figura 30 – Atraso da mensagem por tamanho do TP_{max} , adaptado de (LATRÉ, et al., 2005).

Desta forma, a equação final deverá depender do tamanho do pacote, que influenciará nos tempos para utilização de SIFS ou LIFS, do tamanho do endereçamento. Assim, organizando a equação 3.2 no formato $ax + b$ (reta) para situações com o envio de ACK teremos para pacotes com endereçamento de 2 bytes (16 bits):

$$\text{delay}(x) = 0.000032x + 0.002912,$$

E para endereços de 8 bytes (64 bits):

$$\text{delay}(x) = 0.000032x + 0.003296,$$

Como observado na Figura 30, estes valores podem chegar a quase 7ms caso a mensagem possua mais de 110 bytes, porém, no caso médio, seu valor oscila em torno de 5ms.

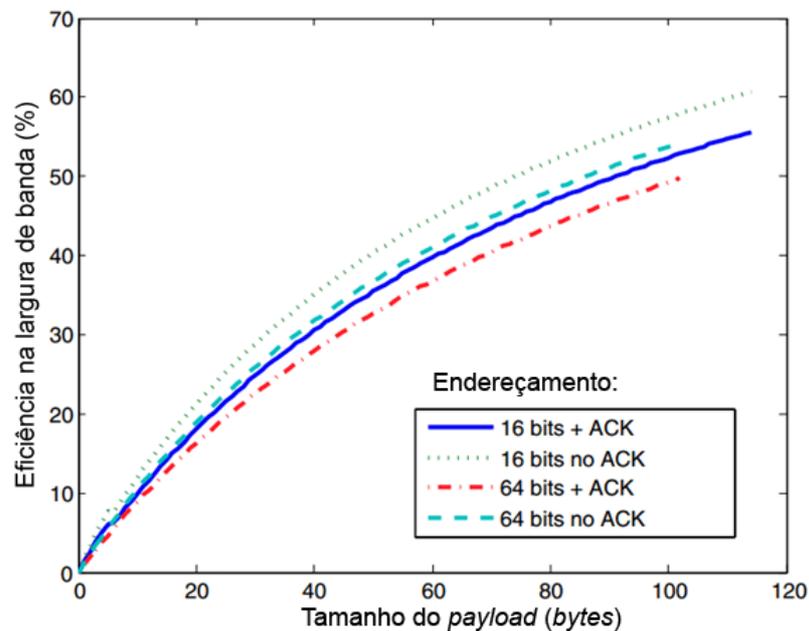


Figura 31 – Eficiência da banda por tamanho do TP_{max} , adaptado de (LATR'E, et al., 2005).

Estes valores de $delay(x)$ unem-se ao cálculo do valor de TP_{max} pela equação 3.1. Assim, de posse dos valores da taxa máxima D_{rate} com TP_{max} verifica-se que a eficiência do sistema de transmissão de dados $\frac{TP_{max}}{D_{rate}}$ pode chegar a 49.8% (124,5 kbps) quando utilizado o valor de endereçamento de 64 *bits* com recebimento de ACK, demonstrado graficamente na Figura 31 para todos os valores de *payload* possíveis.

Para o caso dos dispositivos que não utilizam o padrão IEEE802.15.4, o *driver* fornecido pelo fabricante *Texas Instruments*, chamado MRFI (*Minimal Radio Frequency Interface*) que é camada de mais baixo nível do protocolo proprietário SimpliciTI (TEXAS INSTRUMENTS, 2012), é utilizado como padrão para envio dos dados pela porta *SPI* do microcontrolador ao dispositivo de rádio.

Desta forma, torna-se possível customizar o pacote de dados, fixando-se apenas o primeiro caractere do pacote de dados com o tamanho do *payload* a ser enviado, restrição esta que é imposta pelo *driver*. Assim, dos 64 *bytes* do *buffer* disponíveis para o envio, é possível utilizar integralmente os 63 *bytes*, sendo garantido o uso do driver MRFI, mantendo compatibilidade com outros dispositivos fabricados pela TI compatíveis com SimpliciTI, ficando o pacote como mostrado na Tabela 5.

Tabela 5 - Pacote de rádio com dados úteis do protocolo proposto a partir dos dados do fabricante (TEXAS INSTRUMENTS, 2012).

Preâmbulo	Sincronização	Tamanho	Pacote	CRC
2-4 <i>bytes</i>	2-4 <i>bytes</i>	1 <i>byte</i>	63 <i>bytes</i>	2 <i>bytes</i>
5-9 <i>bytes</i>			Pacote disponível	2 <i>bytes</i>
Cabeçalho PHY		Pacote MRFI		Rodapé PHY
Pacote completo de rádio PHY				

Assim, observa-se que, para este pacote de dados, o atraso encontrado será obtido de acordo com a equação 3.6. Comparando-o com a equação 3.2, percebe-se que o atraso total apresentado para este sistema pode vir a ser menor que no IEEE 802.15.4.

$$delay(x) = T_{BO} + T_{frame}(x) + T_{ACK} + T_{TA} \quad (3.6)$$

Da mesma forma, comparam-se as equações 3.4 e 3.5 com o padrão apresentado pelo protocolo simplificado, baseado nos *drivers* MRFI, regido pelas equações 3.7 e 3.8.

$$T_{frame}(x) = 8 \cdot \frac{L_{PHY} + L_{MRFI_Header} + L_{address} + x + L_{PHY_Footer}}{D_{rate}} \quad (3.7)$$

$$T_{ACK} = 8 \cdot \frac{L_{PHY} + L_{MRFI_Header} + L_{address} + L_{PHY_Footer}}{D_{rate}} \quad (3.8)$$

Onde:

- L_{PHY} – (4-8 *bytes*) Tamanho da camada física e do cabeçalho de sincronização
- L_{MRFI_Header} – (1 *byte*) Tamanho do cabeçalho da camada de enlace (MAC)
- $L_{address}$ – (14 *bytes*) Tamanho do endereço MAC para 4 dispositivos (8 *bytes*) associado ao endereço da PAN (*Personal Area Network*) (6 *bytes*)
- L_{PHY_Footer} - (2 *bytes*) Tamanho do rodapé da camada física
- D_{rate} - Taxa de dados (250Kbps)

Pode-se, então, calcular então a vazão de dados máxima da rede a partir da equação 3.1, da mesma forma que foi calculado para os módulos que utilizam o padrão IEEE 802.15.4. Utilizando um *slot* de tempo do *backoff* equivalente ao escolhido pelo padrão IEEE802.15.4 (320 μ s) obtém-se uma eficiência de 39,51% (ou 98.8Kbps) para pacotes de 32 *bytes* e 60,06% (ou 150.2Kbps) para pacotes de 50 *bytes* de dados, tornando atraente utilizar esta abordagem para um rede elétrica inteligente, visto que é possível enviar até 2 pacotes de dados em uma vez (com 22 *bytes* cada, conforme avaliação da estrutura de dados nesta seção) e, restarem ainda 6 *bytes* dos 50 para o tráfego de

variáveis de controle. Vale ressaltar que este módulo não utiliza o padrão proposto pelo IEEE e sim um padrão proprietário da *Texas Instruments*.

Após a apresentação e cálculo das respectivas eficiências associadas, será executada a avaliação, a partir do cálculo da banda necessária para uma rede em *smart grid*, se é factível a utilização destes padrões, considerando o pior caso (25% de eficiência com pacotes de 32 *bytes* ou 62.5Kbps de uma rede IEEE802.15.4 – vide Figura 31), para conseguir atender à demanda residencial populosa sem grandes atrasos no sistema.

Para este levantamento é requerido que sejam utilizadas algumas premissas e regras de negócio, conforme ilustrado na Figura 32, descritas:

- Só será utilizado um canal livre de rádio por vez;
- Uma residência conseguirá interferir com até seus dois vizinhos adjacentes;
- A medição intervalar será feita a cada cinco minutos;
- Uma residência média possuirá em média dez dispositivos;
- Será transmitida uma mensagem do tipo *keep alive* por segundo, por dispositivo.

Desta forma, tem-se como calcular, para cada intervalo de cinco minutos, a banda necessária pelos equipamentos através da equação 3.9.

$$Banda(t) = \frac{N_{disp} \cdot 8 \cdot x \cdot (2 + 60 \cdot t)}{60 \cdot t} \quad (3.9)$$

Onde:

- t - tempo em minutos (5)
- N_{disp} - número de dispositivos concorrentes (30 unidades)
- x - quantidade de *bytes* de uma mensagem (44)

Resolvendo esta equação, obtém-se o resultado de 10.7Kbps, que é a banda mínima necessária para os dispositivos transmitirem todos os dados coletados durante o intervalo. Desta forma, verifica-se que todos os sistemas apresentados possuem uma banda no mínimo oito vezes maior que a necessária para este tipo de aplicação, validando-se então a continuidade do processo de avaliação dos requisitos sem o descarte de nenhum dos dispositivos apresentados na Tabela 4.

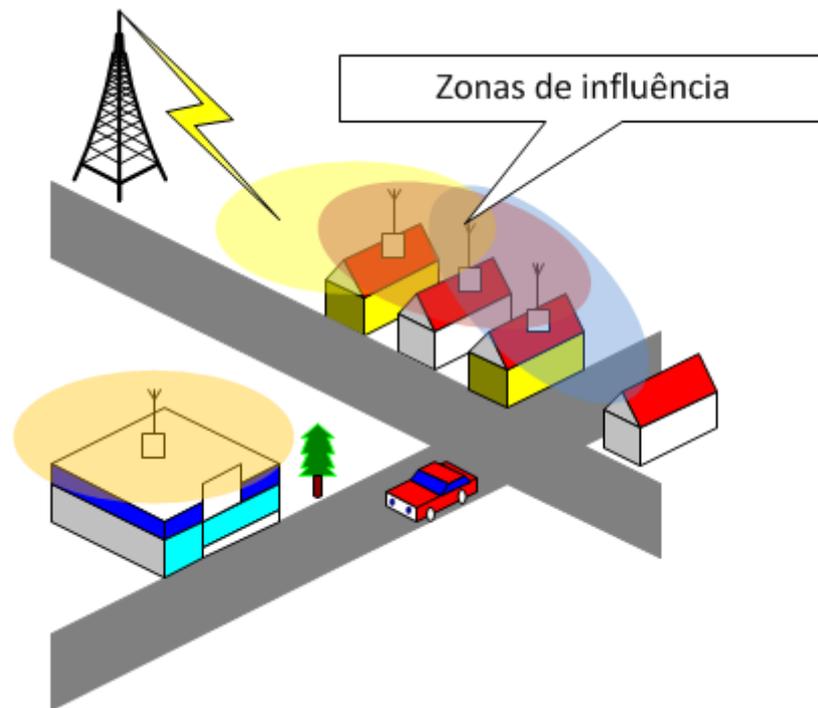


Figura 32 – Esquema proposto para o levantamento da banda necessária por canal.

Com relação à sensibilidade dos rádios, após a análise das especificações da Tabela 4, verifica-se uma vantagem de pelo menos 6dB aos equipamentos que utilizam a camada física IEEE802.15.4, operando na taxa de dados de 250Kbps. Experimentalmente este fato já havia sido observado (LANGHAMMER, et al., 2012).

Estes mesmos experimentos demonstraram, porém, que não haveria falhas na conectividade na rede estudada com equipamentos que apresentassem sensibilidade de até -80.5dB em redes *indoor* com características de rede em malha. Desta forma, todos os equipamentos listados da Tabela 4 ainda possuem especificações técnicas compatíveis com o nível de atenuação a que serão submetidos em situações reais de utilização.

Portanto, todos os equipamentos sugeridos apresentam características técnicas compatíveis com a operação em um ambiente voltado ao *smart grid*. Entende-se, então, que a escolha da solução será baseada em uma análise de mercado, onde o custo de fabricação da plataforma possui maior importância estratégica, visto que tecnicamente todas as soluções atendem aos requisitos mínimos de transmissão de dados para redes elétricas inteligentes.

Desta forma, após averiguação dos valores a partir da Tabela 4, fica constatado que há uma grande disparidade de preço entre os módulos apresentados. A solução estudada que apresenta maior economia de preço de mercado é a solução que utiliza os chips **MSP430** e **CC2500**, ambos da empresa *Texas Instruments*, com valor final ao consumidor (para lotes acima de 1000 unidades) em U\$6.31, contra U\$15.41 da solução ATMEL (redução de 59% no custo) e U\$8.47 (redução de 25.5% no custo) da solução com o mesmo microcontrolador TI com rádio de especificações IEEE802.15.4.

Vale ressaltar, que posteriormente a este estudo preliminar e desenvolvimento inicial da solução foi lançado um SoC (*System on Chip*) da TI contendo um microcontrolador 8051 de baixo consumo de energia em módulo com chip CC2500 (em um único encapsulamento) com o custo unitário de U\$2.86. A utilização de uma solução deste tipo reduziria o custo da solução escolhida atualmente ainda em 54.7%, ficando a avaliação do impacto do custo na performance e seu desenvolvimento como sugestão para trabalhos futuros na Seção 6.2. Esta solução, porém, não foi utilizada para o desenvolvimento do protótipo.

3.2.2. *Software de comunicação*

Após a escolha da plataforma de *hardware*, torna-se necessária a execução de uma metodologia para avaliar, baseado nas limitações do dispositivo, quais parâmetros serão fundamentais dentro da arquitetura, para a criação de uma plataforma de software capaz de suportar a complexidade do projeto e seus posteriores aprimoramentos.

Desta forma, como parte do trabalho de desenvolvimento de uma plataforma completa utilizando o dispositivo CC2500, foi proposto o quadro de dados da Figura 33. Foram propostos ao pacote da mensagem, conforme descrito em 3.2.1, blocos de dados importantes para que a mensagem seja interpretada de forma correta e segura pelo receptor e para que os dispositivos possam trocar mensagens de forma eficiente.

Alguns itens foram acrescentados ao protocolo (campos adicionais), de forma a facilitar o desenvolvimento e depuração de erros durante o processo de codificação do programa, tais como o campo 9 (*debug*), 12 (indicador de qualidade da rota) e 4 (versionamento do protocolo). Desta forma, é possível evitar que protocolos e testes com funcionamentos distintos consigam trocar mensagens e dados entre si. Vale ressaltar que estes campos ainda podem adquirir novas funcionalidades em versões posteriores do protocolo.

Para o funcionamento correto de uma rede em malha, serão necessários ainda quatro campos de ‘endereço’ por pacote de dados, gerando-se então um *overhead* adicional de dados, já previstos no cálculo da equação 3.6, de oito *bytes*. Este formato de endereçamento é baseado em protocolos de roteamento (RFC3561, 2003) com algoritmos reativos para descoberta de rotas, ou seja, para todo pacote a ser transmitido, é feita uma requisição de endereços na rede que, ao receber uma resposta positiva, indica qual sensor receberá o pacote a ser transmitido.



Figura 33 – Quadro de dados proposto para o protocolo.

A comunidade científica vem discutindo aprimoramentos a estes protocolos (IETF MANET, 2012), como pode ser avaliado em novos desenvolvimentos de soluções para descobrimento de rotas (TINGRUI, et al., 2009; CAO, et al., 2010), baseadas em AODV (*Adhoc On-demand Distance Vector Routing*) (RFC3561, 2003). Este protocolo possui alguns mecanismos intrínsecos para adaptação e integração de alta mobilidade e também possui recursos para manutenção das rotas mais utilizadas.

Em uma rede em malha, os dispositivos podem possuir visada direta para seus destinos, ou a transmissão de dados incidirá através de um dispositivo intermediário que encaminhará as mensagens até o destino, conforme pode ser observado no esquema descrito na Figura 34.

Assim, especificamente para o protocolo objeto deste trabalho, todo o processo de descobrimento das rotas intermediárias da rede procura popular dados para tabelas de roteamento. Estas tabelas possuem o ‘endereço de destinatário’ e o ‘endereço do próximo dispositivo’ (próximo salto) onde será enviada esta mensagem, conforme exemplo mostrado na Tabela 6.

No caso de dispositivos com visada direta, o dado do “próximo salto” populado no campo da tabela é exatamente o mesmo do endereço do destinatário. No caso de dispositivos que necessitem de um equipamento intermediário para a entrega de mensagens, o pacote de dados será encaminhado ao primeiro dispositivo que respondeu à requisição executada pelo remetente.

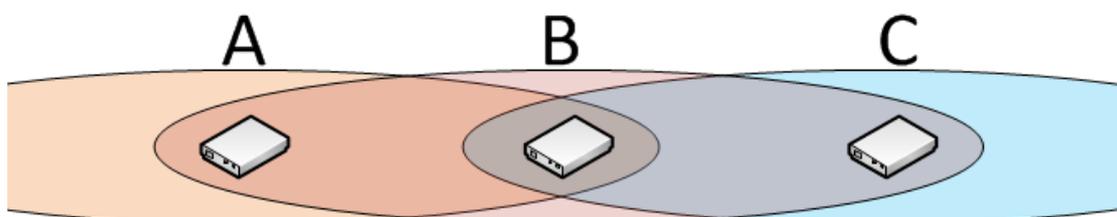


Figura 34 – Modelo conceitual para comunicação entre dispositivos com múltiplos saltos.

Tabela 6 - Tabela exemplo para configuração da Figura 34.

Rotas de A		Rotas de B		Rotas de C	
Destino	Próximo Salto	Destino	Próximo Salto	Destino	Próximo Salto
B	B	A	A	B	B
C	B	C	C	A	B

De forma a validar o pacote recebido pelo dispositivo, o campo ‘identificador da rede’ é utilizado no algoritmo de recepção do pacote pelo dispositivo. Tal ocorrência deve-se pelo fato do ar ser um meio de transmissão de dados compartilhado, onde todos os dispositivos de diversas redes distintas disputam o mesmo recurso, e assim, pacotes podem ser transmitidos e capturados, dentro de um mesmo raio de ação, por dispositivos aptos ou não a receber e analisar os dados em trânsito.

Assim, obtém-se a execução da verificação do endereço de rede da mensagem, onde pacotes de redes distintas com a mesma região de atuação são descartados, além da validação do endereço de destino da mensagem. Caso o endereço lógico do dispositivo ou um endereço de *broadcast* de rede exista na mensagem, ela poderá então ser processada pelo equipamento.

Além disto, como solução para o isolamento entre as redes, foram adicionadas, ao pacote de dados do protocolo deste trabalho, estruturas para versionamento, segurança, e identificação, como forma de prevenção à entrega de pacotes indesejados, ou utilização indevida de dados por dispositivos externos, conforme ilustrado na Figura 35.

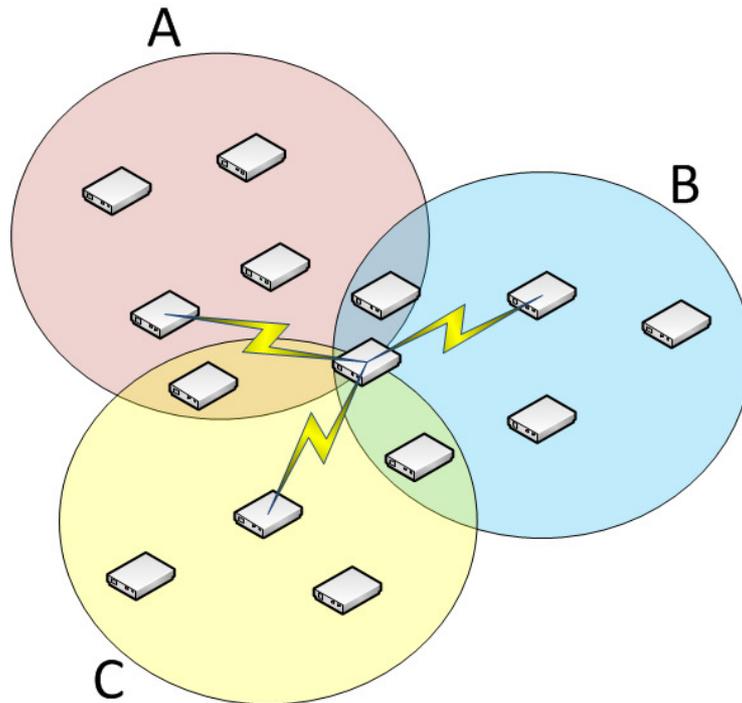


Figura 35 – Diagrama de comunicação exemplificando áreas de influência e interferência entre as redes distintas A, B e C convivendo em um mesmo espaço físico.

No atual estágio da pesquisa e desenvolvimento deste trabalho, ainda não foi objeto de estudos a utilização de criptografia para o encaminhamento de pacotes, porém, a sua utilização é necessária ficando o seu desenvolvimento como sugestão para trabalhos futuros na Seção 6.2, visto que seu desenvolvimento envolve um complexo sistema de troca de chaves públicas e privadas e validação de algoritmos que não fazem parte do escopo desta pesquisa. Porém, já é possível vislumbrar uma proposta (FOUDA, et al., 2010) de segurança para a arquitetura do padrão *Zigbee* IEEE 802.15.4 em maiores detalhes, tomando-se como base para desenvolvimentos futuros.

Todo o processo de validação da mensagem é executado de acordo com o fluxograma da Figura 36. No processo inteiro são validados os campos que garantem a integridade do pacote, tais como cabeçalho, rodapé, além de ser executada a verificação de segurança com o cálculo do CRC da mensagem, onde se utiliza todos os *bytes* da mensagem, além de um código de verificação específico, que garante que o receptor deve reconhecer o algoritmo de montagem e validação do pacote.

Em uma análise mais detalhada do processo de recebimento de mensagens, observa-se na Figura 37 uma porção da complexidade deste sistema. O processo de recebimento de dados começa com uma interrupção (ISR - *Interrupt Service Routine*) do microcontrolador, executada externamente pelo dispositivo de rádio. Pelo fato do dispositivo não saber o instante de tempo que aquela interrupção está acontecendo, e em qual contexto de programa ele foi interrompido, é necessário que seja feita uma avaliação do uso de rede, representada pela variável *EmUso*.

Este recurso permite ao protocolo validar se há alguma transmissão ou processamento de dados em curso, e, em caso positivo, o processo de interrupção deverá ser abortado, liberando recursos computacionais para que a ação em curso continue a ser executada. Neste caso a mensagem recebida será eliminada do *buffer* do rádio e não será aproveitada em outra oportunidade, ou seja, há o descarte do pacote.

Caso o dispositivo esteja livre para a execução da tarefa, é feita a chamada de procedimento para recebimento da mensagem, neste caso específico via interface SPI (*Serial Peripheral Interface Bus*) para a memória de trabalho do microcontrolador, onde será analisada pelo fluxograma apresentado na Figura 36. Com a validação positiva, o pacote é então encaminhado às camadas superiores do protocolo para que seja avaliado qual o tipo de mensagem que o dispositivo está tratando.

A partir deste momento, o dispositivo de rádio não poderá mais interromper o processo em curso, até que se possa tomar alguma ação de processamento, ou, caso seja uma mensagem espúria, haja a ação de descarte do pacote. Neste momento é importante validar se há algum procedimento em paralelo aguardando o recebimento de ACKs (*acknowledgements*), visto que o processamento de uma mensagem ACK deverá encerrar um processo em andamento, enquanto que o recebimento de uma **tarefa** ou **resposta de tarefa** indicará a necessidade de pós-processamento e o envio de uma mensagem de reconhecimento ao remetente, garantindo que houve o entendimento completo do pacote recebido por parte do destinatário.

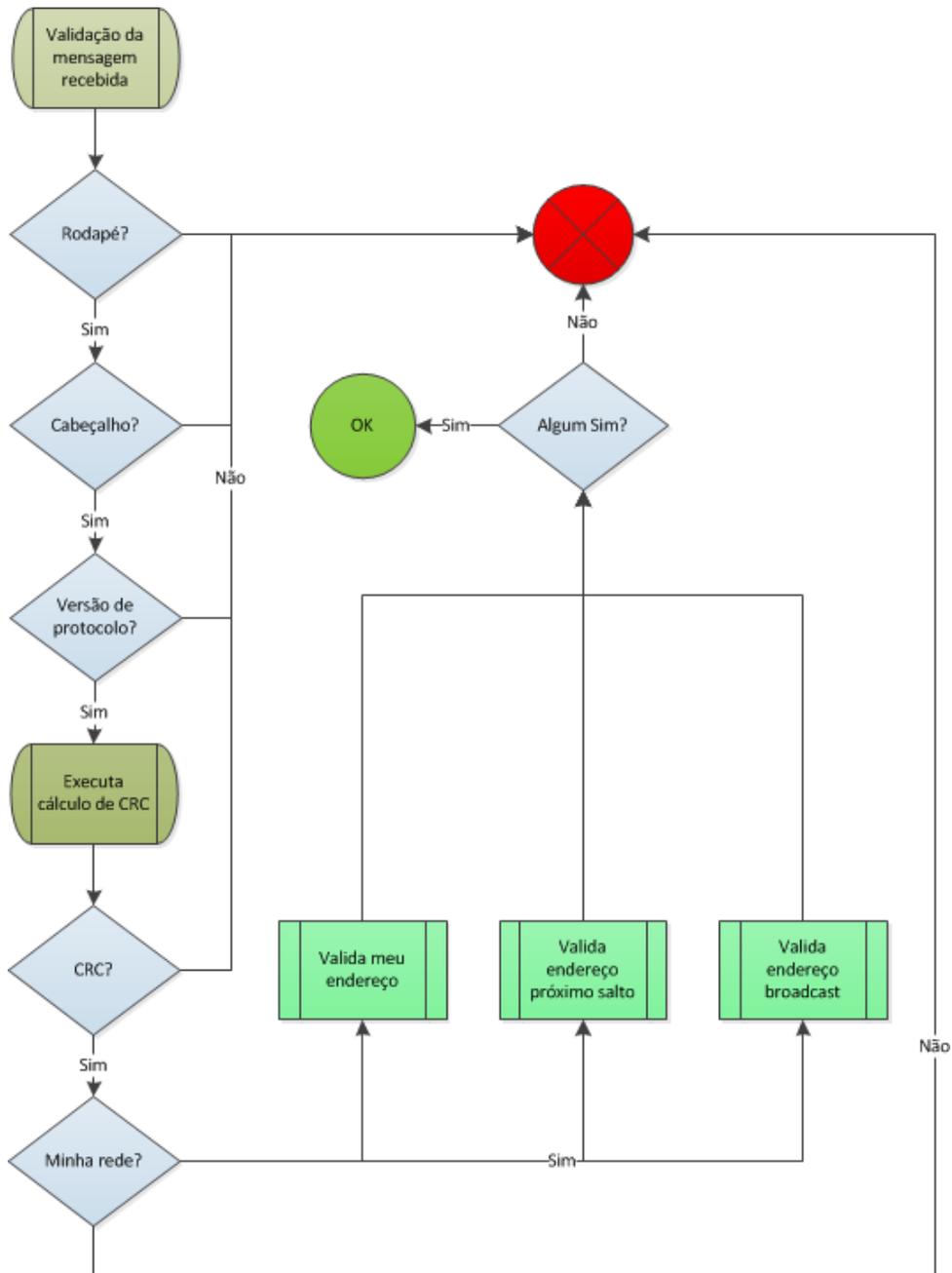


Figura 36 – Fluxograma de validação das mensagens recebidas pelo protocolo.

Para isto, foi utilizada no escopo do trabalho a identificação da variável *JOB*, indicando um estado de “execução de tarefas”, onde o microcontrolador é então acordado do seu estado de latência com baixo consumo e passa a executar a tarefa disponível, gravada no *buffer* do microcontrolador, e recebida do pacote do *hardware* de radiofrequência.

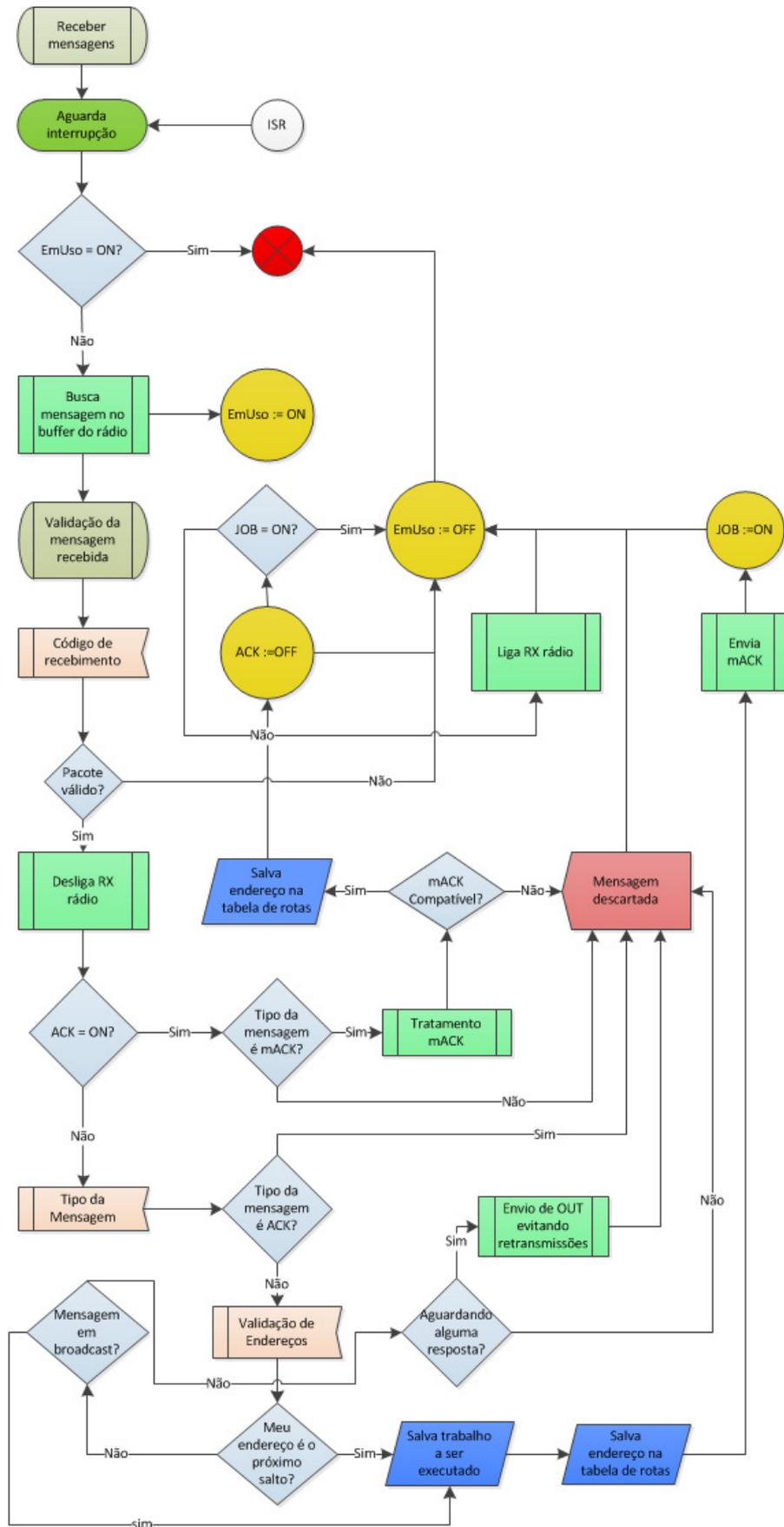


Figura 37 – Fluxograma de recebimento e validação das mensagens pela camada de enlace.

3.3. Levantamento de tarefas e respostas

Para o protocolo escopo deste trabalho, todo o processo de troca de mensagens entre dispositivos que o utilizem está baseado no conceito de tarefas (ou *tasks*) e respostas (ou *replies*). A organização dos processos a serem executados pela rede em *smart grid* é um ponto onde este trabalho diverge das redes e padrões expostos como estado da arte.

É notório que esta discussão venha do fato que, empiricamente, ao exercer tarefas e serviços genéricos de rede, há uma pressão forte no desenvolvimento do projeto para a inserção de sobrecarga de processamento, visando identificar as generalidades da transmissão de dados e somente então, em camadas lógicas superiores, executar as tarefas específicas do *smart grid*. Foi observada uma análise aprofundada (SAAKSJARVI, 2002) sobre o alinhamento das expectativas genéricas de um software proposto com custo, eficiência, manutibilidade, abrangência do sistema e impacto técnico de arquiteturas generalistas ou especialistas.

Desta forma, tendo em vista os componentes utilizados neste tipo de equipamento (listados na Seção 3.2.1), com baixo poder computacional e baixo custo, é então esperado que haja uma pressão para uma maior especialização do protocolo objeto deste trabalho, justificando-se a escolha por componentes que consigam atender a uma finalidade específica de forma a, ainda assim, possuir alta competitividade quando comparado à sistemas mais generalistas, tais como o *Zigbee* (ZIGBEE ALLIANCE, 2012).

Assim, o protocolo utiliza na própria camada de enlace a pré-definição de tarefas específicas de *smart grid*, automação e controle, necessárias ao funcionamento do sistema, evitando classificações e estruturas de dados generalistas complexas em excesso e/ou exageradas, que possam sobrecarregar o sistema com *overhead* de dados, limitando seu potencial de funcionamento, tendo em vista um cenário de adoção em massa da plataforma.

Assim, por definição, toda mensagem deverá possuir um campo, mostrado na Figura 33, com a identificação da mensagem. Este serviço disponibiliza até 256 primitivas/possibilidades de tarefas distintas, sendo que para cada tarefa, há sempre associada uma resposta. Ou seja, para cada pedido de **ação**, haverá o disparo de um **procedimento de execução** e o envio de uma **confirmação** do recebimento, como pode

ser visto na Figura 38. Ou seja, para cada código de tarefa associado, há uma resposta a ser enviada de forma a confirmar o recebimento do comando pela estação de destino. Desta forma, observa-se uma maior robustez na comunicação do sistema, garantindo que a tarefa foi executada de forma suficiente pelo executor.

Além disto, códigos pré-definidos informam falhas na execução das tarefas. Assim, o dispositivo que envia a mensagem, consegue discernir entre tarefas executadas com sucesso (via resposta), e tarefas que não foram executadas ou recebidas pela estação receptora. No caso da falta do reconhecimento (ACK) ou de uma resposta válida por parte do destinatário, o remetente deverá iniciar o procedimento de reenvio automático de mensagens, conforme será descrito na Seção 3.4.

Mensagens recebidas em *broadcast* não esperam recebimento de ACK, como pode ser avaliado pelo fluxograma da Figura 37. Esta categoria de mensagem está restrita a mensagens de controle, associação e inserção às tabelas de roteamento, e descoberta de novas rotas.

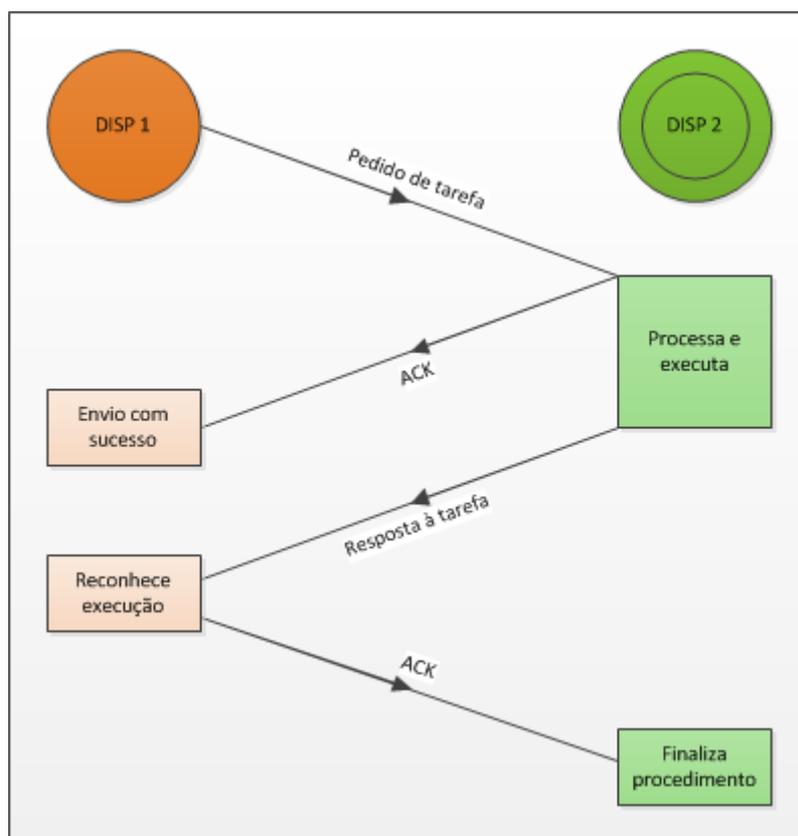


Figura 38 – Sequência de troca de mensagens do tipo tarefa e resposta.

Assim, todo o desenvolvimento do protocolo de roteamento para esta plataforma foi baseado em um modelo híbrido, que se utiliza de dois princípios básicos:

- **A utilização da descoberta de rotas em modelo proativo**

- Pelo fato de não haver grande mobilidade entre os nós da rede em um cenário residencial e/ou comercial em *smart grid*, visto que tomadas e disjuntores estão fisicamente conectados à infraestrutura, são enviados pacotes de sinalização na frequência de um segundo, avisando aos dispositivos próximos, de uma mesma rede e alcance, seu *status* e conectividade;
- Estes avisos populam a tabela de rotas do dispositivo, criando, desta forma, uma identificação local dos dispositivos próximos ativos no momento;
- Após falhas sucessivas na recepção destes avisos, o dispositivo é, então, removido da tabela.

- **A utilização da descoberta de rotas em modelo reativo**

- Pelo fato de não haver grande capacidade de armazenamento de dados em memória em dispositivos embarcados de baixo custo, a tabela de rotas disponível possui um número limitado de entradas, e com isso são limitados os possíveis destinos em memória (atualmente em oito). Desta forma, na tentativa de enviar mensagens para endereços não listados, ou removidos da tabela de roteamento, o dispositivo inicia o procedimento de descoberta de novas rotas;
- É enviado então, em *unicast*, de forma sequencial, para cada um dos dispositivos da lista de roteamento, um requerimento solicitando a localização do destinatário desconhecido. No caso de uma tabela de rotas vazia, a operação é abortada;
- Após a varredura de toda a lista e o envio das solicitações de nova descoberta de rotas, são aguardadas mensagens de resposta, com o algoritmo descrito na Figura 38, quando simultaneamente são disparados temporizadores que limitam o tempo da validade máxima do pedido de rota.

- Caso sejam recebidas respostas dentro do tempo esperado, o dispositivo adiciona uma nova rota na tabela e faz o envio da mensagem com o novo dado em memória;
- No caso de respostas recebidas por outros dispositivos após a adição da primeira rota, novas tentativas de inserção serão descartadas, até que a entrada primária na tabela seja removida por falta de conectividade com o nó encaminhador;
- Caso haja desconectividade com algum nó da rede contido em sua tabela de rotas, é emitida uma mensagem de erro em *broadcast* avisando à zona de influência que a conexão com aquele dispositivo foi desfeita. Caso algum dispositivo possua uma rota em ponte para o destinatário informado pelo nó encaminhador da mensagem de erro, a ligação é então desfeita em cadeia, por inundação.
- Caso seja emitida uma mensagem para um nó, com destino diferente do endereço de próximo salto, e este destinatário não esteja em sua tabela de rotas, é emitida outra mensagem de erro em *broadcast* para que a rede torne-se coesa, após o rompimento da ligação entre os nós envolvidos.

O fluxograma do procedimento descrito para descobrimento de rotas pode ser avaliado na Figura 39. A lista de tarefas e respostas, com descrição das primitivas de cada comando utilizado pode ser vista na Tabela 7.

Tabela 7 – Primitivas de tarefas e respostas utilizadas pelo protocolo.

Primitiva	Descrição
MT_DISCOVER_RANGE	Mensagem enviada a cada segundo, em broadcast para identificação das zonas de influência entre nós e preenchimento de tabelas de rotas locais.
MT_GET_SERIAL_NUMBER	Mensagem que busca o número de série do equipamento.
MT_GET_SERIAL_NUMBER_REPLY	Resposta à mensagem que busca o número de série do equipamento. Dados do número de série são transferidos no pacote de dados.
MT_GET_ENERGY_DATA	Mensagem que faz a requisição do <i>status</i> atual completo de energia elétrica na rede.
MT_GET_ENERGY_DATA_REPLY	Resposta à mensagem que faz a requisição do <i>status</i> atual de energia elétrica na rede. Dados da última medição de tensão _{rms} , corrente _{rms} potência ativa, data e hora, e o

	acumulador de kWh são transferidos no pacote.
MT_TURN_ON_ENERGY	Mensagem que indica um pedido de execução para acionamento da chave de potência e energização da carga.
MT_TURN_ON_ENERGY_REPLY	Resposta de confirmação da mensagem do pedido de execução para acionamento da chave.
MT_TURN_OFF_ENERGY	Mensagem que indica um pedido de execução para desligamento da chave de potência e desenergização da carga.
MT_TURN_OFF_ENERGY_REPLY	Resposta de confirmação da mensagem do pedido de execução para desligamento da chave.
MT_DISCOVER_RSSI	Mensagem que busca o valor do RSSI (<i>Received signal strength indication</i>) do dado recebido.
MT_DISCOVER_RSSI_REPLY	Resposta à mensagem que busca o valor do RSSI do receptor. Dados do RSSI são transferidos no pacote.
MT_MSG_ACK	Mensagem com identificador de ACK (<i>Acknowledgement</i>). É utilizada como resposta a toda mensagem transmitida em <i>unicast</i> como uma garantia de entrega.
MT_MSG_RREQ	Mensagem utilizada para envio de pedidos para descoberta de rotas aos dispositivos próximos (dentro do raio de alcance).
MT_MSG_RREPLY	Resposta à mensagem com o pedido de descoberta de rotas, confirmando uma entrada em sua tabela.
MT_MSG_RERROR	Mensagem em <i>broadcast</i> utilizada para informar a desconexão de um dispositivo a uma entrada em sua tabela de rotas. Caso haja um pedido de encaminhamento de mensagens para uma entrada que não esteja em sua lista, esta mensagem também é enviada.
MT_NEW_DATETIME	Mensagem que faz o envio da atualização de nova data e hora que irão ser adicionadas ao dispositivo. Dados de data, hora, minuto, segundo, dia, mês e ano são transmitidos ao dispositivo.
MT_NEW_DATETIME_REPLY	Resposta de confirmação à mensagem de atualização de data e hora.
MT_RESET	Mensagem que faz o pedido de reinicialização do dispositivo. Esta mensagem é a única que gera o envio de confirmação de execução antes de acionar o mecanismo de <i>watchdog</i> do microcontrolador.
MT_RESET_REPLY	Resposta de confirmação à mensagem de pedido de reinicialização do dispositivo. Esta mensagem é enviada antes da execução da tarefa, visto que toda a memória e contexto são apagados após sua execução.
MT_OFF_ALERT	Mensagem que chama uma função de “Alerta” ao dispositivo. Pode indicar com o acendimento de um <i>LED</i> que o dispositivo está entrando em horário de ponta (maiores tarifas), ou pode indicar um desligamento programado alguns minutos antes do efetivo evento ocorrer. Deve ser utilizada de acordo com a funcionalidade proposta pelo desenvolvedor.
MT_OFF_ALERT_REPLY	Resposta de confirmação à função de alerta, indicando que foi executado o pedido de alerta.
MT_GET_DUMP	Mensagem que faz o pedido de recuperação completa da memória do dispositivo. Todos os arquivos gravados são descarregados em modo <i>burst</i> para o requerente dos dados. Esta funcionalidade ainda não possui implementação nesta

	versão do dispositivo, mas já está prevista a sua implantação.
MT_GET_DUMP_REPLY	Resposta única de confirmação do recebimento das mensagens em modo <i>burst</i> do pedido de recuperação completa da memória do dispositivo.
MT_SET_REC_MODE	Mensagem que indica o modo de gravação que deverá ser utilizado na memória de massa. Até o momento existem os modos com sobrescrita, ou seja, modo de gravação cíclico, ou sem sobrescrita, onde o dispositivo deixa de gravar ao completar a memória de massa. É enviado no <i>payload</i> da mensagem um <i>byte</i> indicando o novo modo de gravação.
MT_SET_REC_MODE_REPLY	Resposta de confirmação da execução na alteração do modo de gravação do dispositivo. É enviado como confirmação o mesmo <i>byte</i> que foi recebido na mensagem da tarefa, garantindo que o dado foi entendido e está íntegro.
MT_SET_LOG_INTERVAL_TIME	Mensagem que altera o valor do intervalo de gravação da memória de massa do dispositivo. O valor é definido em minutos e um <i>byte</i> é enviado no <i>payload</i> da mensagem com o valor desejado. O valor mínimo acreditado pelo dispositivo é de 1 minuto e o valor máximo é de 120 minutos.
MT_SET_LOG_INTERVAL_TIME_REPLY	Resposta de confirmação para a alteração do valor do intervalo de gravação da memória de massa do dispositivo. Dentro do pacote de confirmação, é enviado o novo valor que foi configurado no dispositivo.
MT_GET_LOG_INTERVAL_TIME	Mensagem que busca o valor atual do intervalo de gravação da memória de massa do dispositivo. O valor mínimo respondido pelo dispositivo deve ser de 1 minuto e o valor máximo de 120 minutos.
MT_GET_LOG_INTERVAL_TIME_REPLY	Resposta de confirmação, trazendo o valor atual do intervalo de gravação da memória de massa do dispositivo dentro do <i>payload</i> da mensagem.
MT_GET_LAST_ADDR_REC	Mensagem que busca receber o valor atual do ponteiro de endereçamento de gravação da memória de massa do dispositivo. Por possuir um mapeamento de 16 <i>bits</i> , o valor mínimo respondido pelo dispositivo deve ser 0 e o valor máximo de 65535.
MT_GET_LAST_ADDR_REC_REPLY	Resposta de confirmação, trazendo o valor atual do ponteiro de endereçamento de gravação da memória de massa do dispositivo dentro do <i>payload</i> da mensagem.
MT_GET_TOTAL_UNSENT_REC	Mensagem que busca receber o valor do total de registros ainda não enviados pelo dispositivo. Por possuir um mapeamento de 16 <i>bits</i> , o valor mínimo respondido pelo dispositivo deve ser 0 e o valor máximo de 65535.
MT_GET_TOTAL_UNSENT_REC_REPLY	Resposta de confirmação, trazendo o valor atual do total de registros não enviados pelo dispositivo dentro do <i>payload</i> da mensagem.
MT_GET_SPECIFIC_REC	Mensagem que busca receber o valor dos dados gravados em um endereço específico de memória do dispositivo. Por possuir um mapeamento de 16 <i>bits</i> , o valor mínimo do pedido executado deve ser 0 e o valor máximo de 65535.
MT_GET_SPECIFIC_REC_REPLY	Resposta de confirmação, trazendo o valor completo do registro em memória gravado pelo dispositivo dentro do

	<i>payload</i> da mensagem.
MT_CLEAR_EMETER	Mensagem que apaga o valor dos dados gravados em toda a memória do dispositivo. O pedido executa o apagamento completo na memória de massa.
MT_CLEAR_EMETER_REPLY	Resposta de confirmação da mensagem que executa o apagamento dos dados.
MT_CLEAR_EVENT_PROG	Mensagem que apaga o valor dos dados gravados em toda a memória destinada a programações de eventos do dispositivo. Cada pedido executa o apagamento completo nas programações.
MT_CLEAR_EVENT_PROG_REPLY	Resposta de confirmação da mensagem que executa o apagamento das programações.
MT_SEND_EVENT_PROG	Mensagem que envia uma determinada programação desejada para um endereço específico de memória do dispositivo. Por possuir um mapeamento de 8 <i>bits</i> , o valor máximo do número de programações a serem executadas pelo dispositivo é de 256.
MT_SEND_EVENT_PROG_REPLY	Resposta de confirmação, trazendo o valor do registro de programação em memória gravado pelo dispositivo dentro do <i>payload</i> da mensagem. Caso o valor do pedido executado já esteja preenchido, é enviado o valor do próximo endereço do banco de memória livre para que seja feito o reenvio das informações.
MT_READ_EVENT_PROG	Mensagem que busca o valor da programação gravada em um endereço específico de memória do dispositivo. Por possuir um mapeamento de 8 <i>bits</i> , o valor mínimo do pedido executado deve ser 0 e o valor máximo de 255.
MT_READ_EVENT_PROG_REPLY	Resposta de confirmação, trazendo o valor completo do registro de programação em memória gravado pelo dispositivo dentro do <i>payload</i> da mensagem.

O microcontrolador utilizado pelo projeto possui pouca memória, e, desta forma, o projeto e estrutura do algoritmo de roteamento tiveram que ser adaptadas para utilização ótima dos recursos computacionais. Assim, apenas oito rotas são mapeadas por vez na tabela de roteamento de cada dispositivo.

Por possuir características dinâmicas, é indicado guardar os dados da tabela de roteamento na memória RAM (*Random Access Memory*). Evita-se, desta forma, a utilização de memória *flash* para este fim, pois após diversos apagamentos em uma mesma região de memória há a consequente redução da vida útil do equipamento.

Cada entrada na tabela de roteamento do dispositivo possui 6 *bytes*, sendo 2 *bytes* para cada endereço de destino e próximo salto (4 *bytes* no total), um *byte* para um contador denominado “válido”, que é decrementado a cada segundo pelo temporizador, e, elevado ao seu máximo a cada mensagem do tipo MT_DISCOVER_RANGE (vide

Tabela 7) recebida, indicando que há conectividade com o dispositivo; O último *byte* da tabela é utilizado para identificação da rota, porém, em desenvolvimentos futuros, poderia vir a ser utilizado para guardar dados estatísticos de transmissão, sendo um campo essencial para a escolha de qual rota deve ser removida no caso de uma sobrescrita e/ou esgotamento da capacidade total da tabela de rotas. Vale lembrar que o campo “válido” é atualizado de acordo com o próximo salto, e não com o endereço de destino na tabela.

Assim, com estas dimensões (8 posições x 6 *bytes*), há um consumo total de 48 *bytes*, de uma memória com 1024 *bytes* de RAM (para todas as variáveis de programa, *buffers* e pilha de memória).

O atual algoritmo de substituição de rotas, utilizado para apagar uma entrada da tabela e descrito na Figura 39, funciona com o princípio de:

- Buscar um índice na tabela onde o campo “válido” é igual à zero;
 - Em caso positivo, utiliza-se a posição e é incrementado o valor do contador de rotas;
 - No caso da tabela estar cheia, remove-se um dispositivo com a rota mais antiga (menor valor do identificador de rotas) e utiliza-se a posição para a nova rota, sobrescrevendo-a.

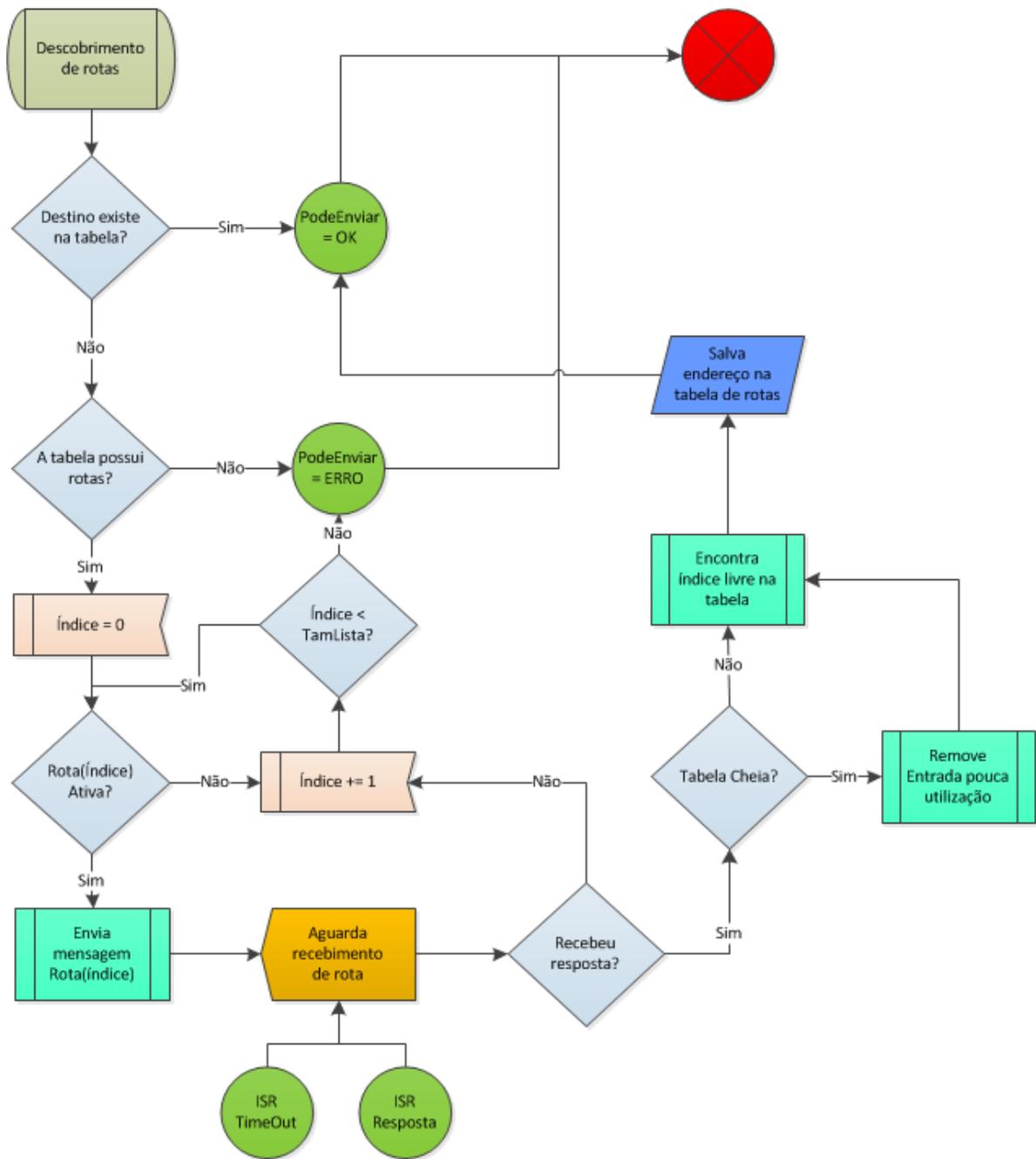


Figura 39 – Fluxograma de descobrimento de rotas.

3.4. Procedimento de envio de mensagens

Todo o procedimento para envio de mensagens está descrito no fluxograma da Figura 40. O processo é iniciado com o recebimento, por parte do microcontrolador, de um pedido de ação via comunicação USB (no caso de dispositivos controlados e ligados diretamente a um computador), ou através de uma atividade/tarefa, descritas na Tabela 7, que tenham como resposta uma mensagem de confirmação e/ou dado requisitado.

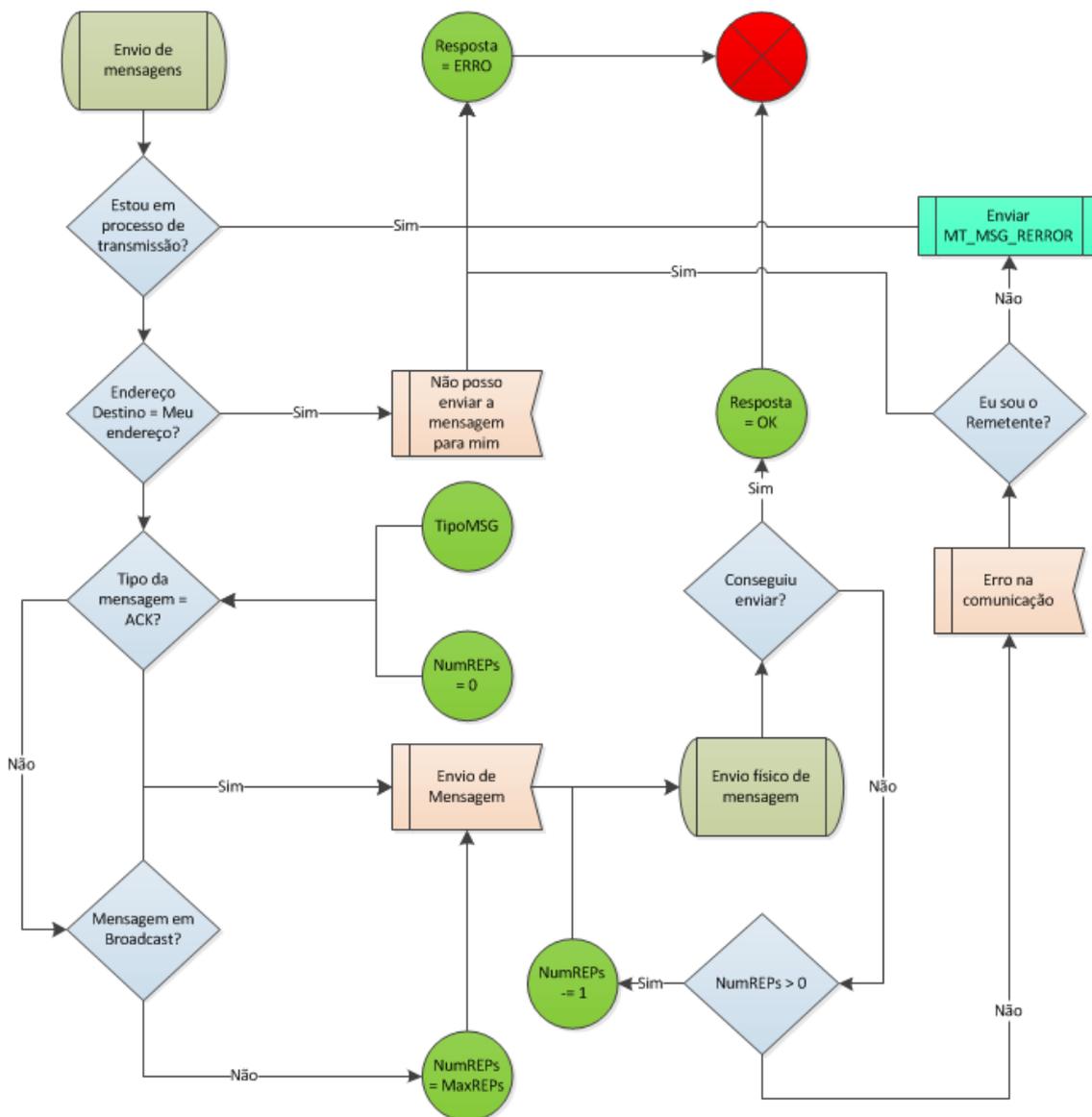


Figura 40 – Fluxograma geral do procedimento de envio de mensagens.

No *chip* CC2500 de rádio, utiliza-se a funcionalidade de avaliação de ocupação do meio através do recurso CCA (*Clear Channel Assessment*). Através desta característica intrínseca ao *hardware* é possível indicar, com certo nível de precisão, se há outros dispositivos transmitindo no canal antes de executar o envio de dados.

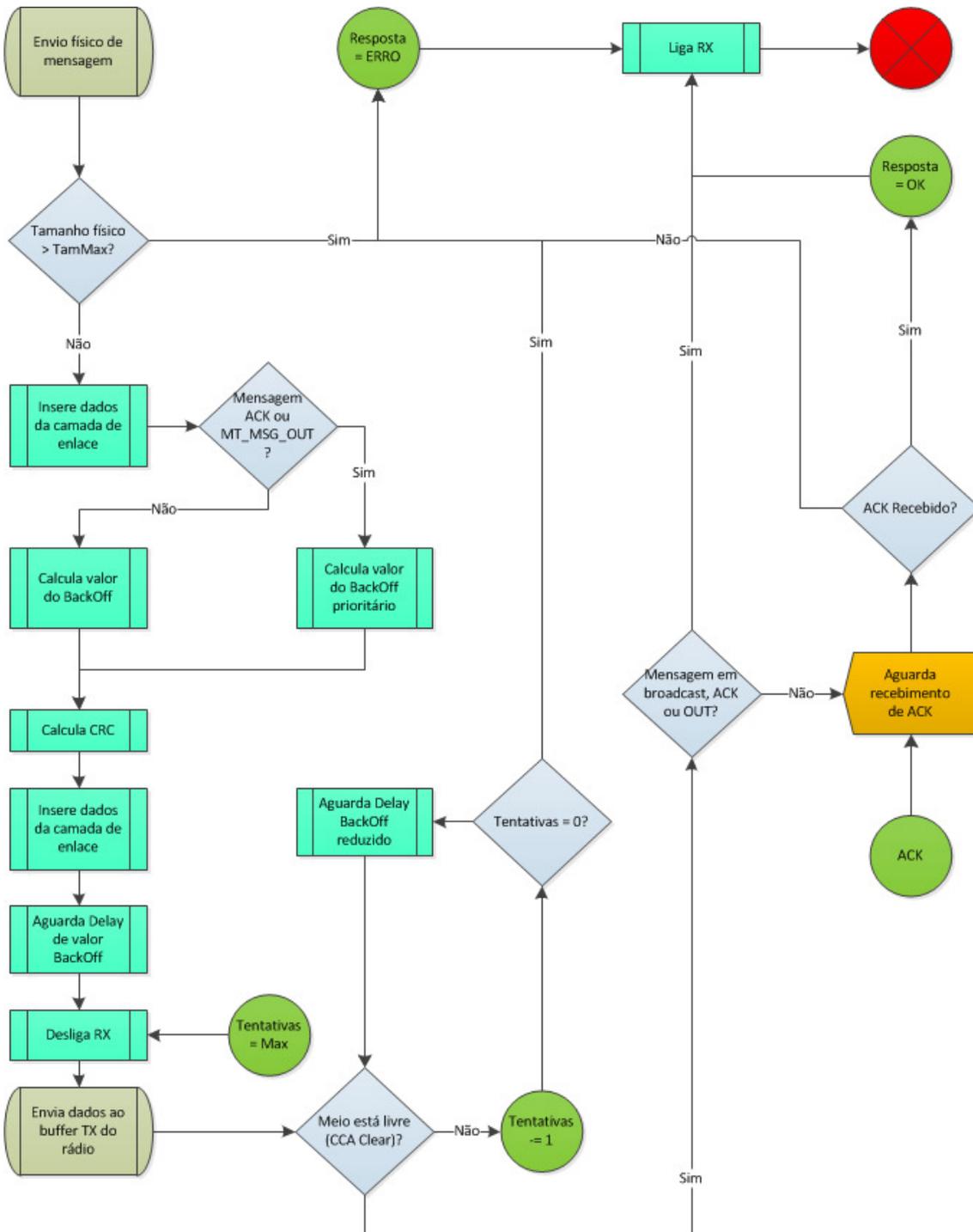


Figura 41 – Fluxograma do procedimento de envio da camada de enlace do dispositivo.

Desta forma, o sistema, então, detecta e avalia se há condições de enviar um pacote, e informa ao microcontrolador, através de uma porta IO, se é possível transmitir, evitando assim colisões. A utilização deste mecanismo, como descrito no fluxograma da Figura 41, é similar ao processo de CSMA/CA (FEDERAL STANDARD 1037C, 1996) inserido em diferentes tecnologias de redes sem fio, já descritas e avaliadas no Capítulo 2.

O processo completo para envio de mensagens inclui o cálculo de uma temporização para o processo de disputa do meio, chamada *backOff*. As mensagens são agrupadas em dois grupos distintos: mensagens de alta prioridade (ACKs e MT_MSG_RERROR) e mensagens de prioridade normal (todos os outros tipos). As mensagens de alta prioridade possuem esta designação por se tratarem de mensagens que devem ser enviadas antes da próxima tentativa de retransmissão por parte do remetente, procurando, assim, melhorar a integridade e coesão da rede e suas rotas.

Desta forma, mensagens de ACK podem ser trocadas no intervalo entre uma mensagem e outra, com probabilidade reduzida de que outro dispositivo obtenha o acesso ao meio neste intervalo. No caso de outro dispositivo transmitir entre o envio de uma mensagem e uma resposta de ACK, haverá a retransmissão da mensagem de reconhecimento com tempos de prioridade sempre mais baixos que os de uma mensagem comum, com um número máximo de tentativas de retransmissão.

No intuito de validar o diagrama de estados de uma rede com disputa do meio, foi executada uma análise para a validação do processo por Redes de Petri (Seção 3.5), onde é modelado o controle de acesso ao meio e a disputa, de forma a garantir que o processo não possua falhas críticas e/ou estados indesejáveis e irreversíveis no fluxograma de funcionamento do sistema.

3.5. Validação por Redes de Petri

Na busca por analisar possíveis falhas no algoritmo proposto, foi criada uma análise do modelo de estados a partir de Redes de Petri. Assim, como observado na Figura 42, foi gerado um modelo conceitual de troca de mensagens entre serviços do sistema a ser modelado.

Observa-se que o protocolo chama a primitiva *SEND* no cliente, invocando a primitiva *RECV* do receptor. Isto gera uma tarefa a ser executada e uma retransmissão de dados e/ou confirmação a ser enviada, descritos por *SENDDATA*. Note que todos os reconhecimentos (*ACKs*) são executados de forma transparente, sem que a camada de rede tome conhecimento de sua transmissão e confirmação. Desta forma, fica bem clara a diferenciação entre ‘procedimento de envio de mensagens’ e ‘procedimento de envio físico de mensagens’, conforme descrito no item 3.4 (Figura 40 e Figura 41). Eles atuam em camadas distintas de serviço, porém sob um mesmo protocolo.

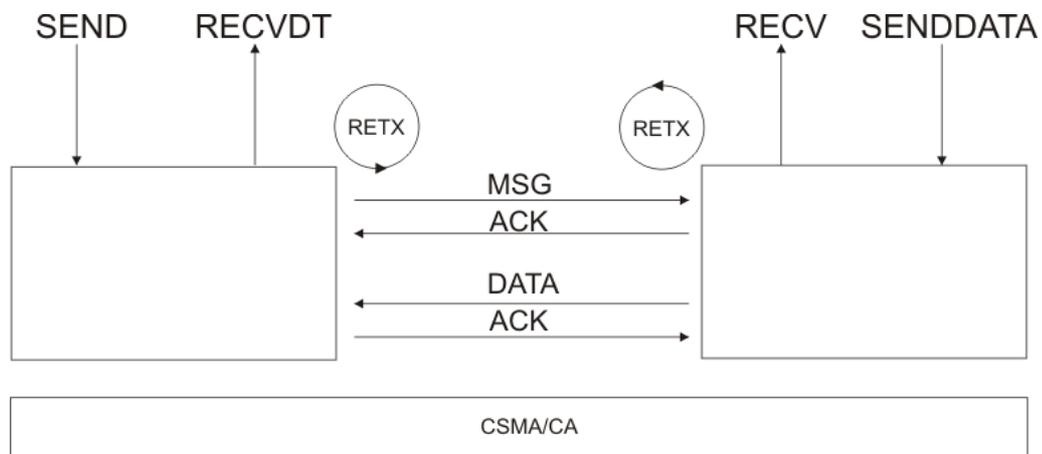


Figura 42 – Primitivas de serviço para modelagem em Redes de Petri.

Após o recebimento de *RECV* e posterior processamento e envio de *SENDDATA*, é então iniciado, no cliente, o processo de *RECVDT*, indicando a entrega com sucesso da mensagem, ou seja, a sua confirmação com os dados necessários ao agente emissor.

O serviço destacado como CSMA/CA na camada inferior (PHY) é o responsável por evitar a transmissão no momento que o meio estiver ocupado, descrito no tópico 3.4, ordenando, e administrando, o tráfego de pacotes em uma rede onde o meio de

transmissão é compartilhado entre os dispositivos, possuindo um impacto relevante na redução do número de colisões, conforme já comentado anteriormente.

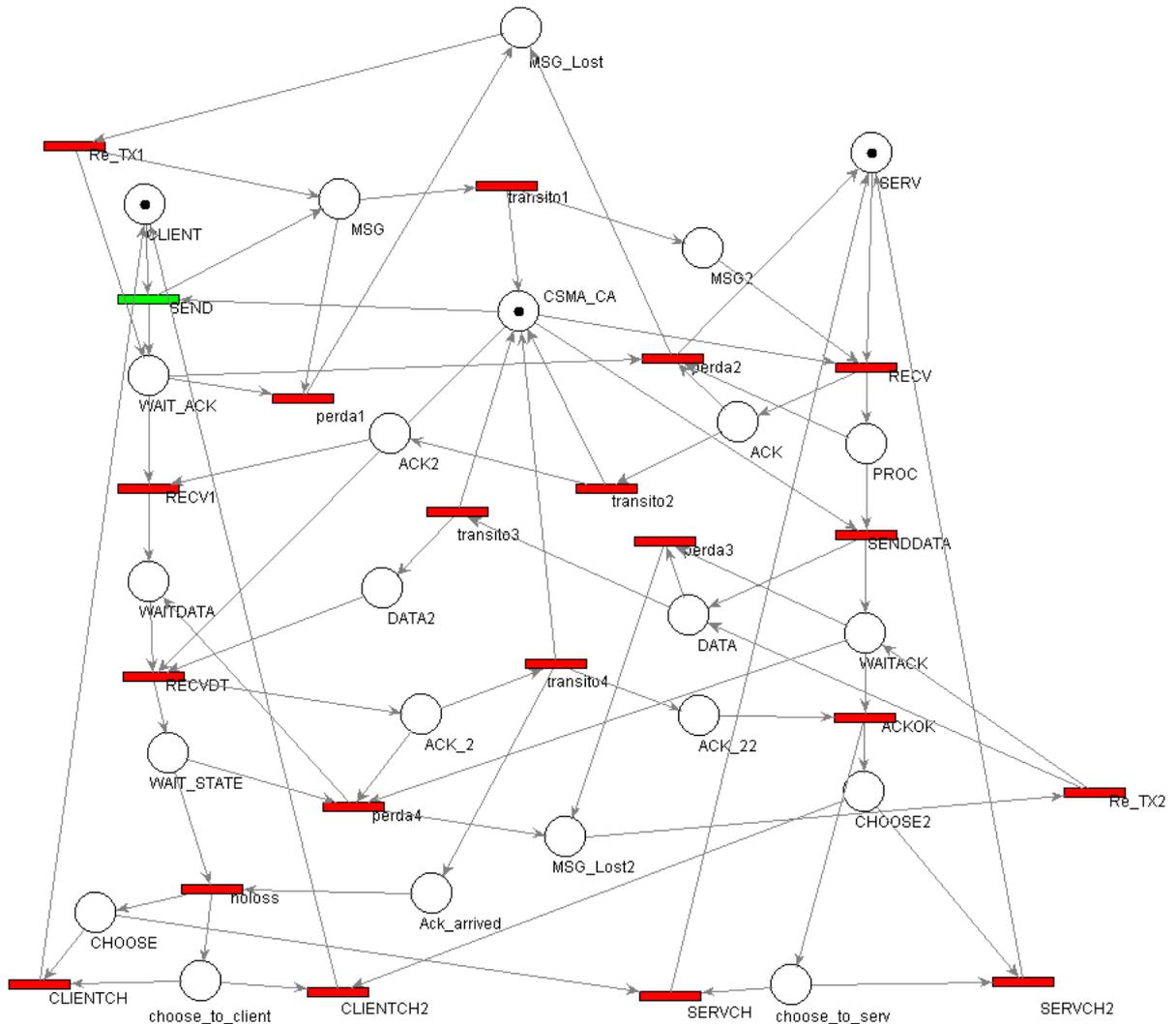


Figura 43 – Rede de Petri do protocolo para dois dispositivos – Emissor / Receptor.

A rede de Petri, descrita na Figura 43, procura modelar todo o procedimento de envio e recepção de mensagens, com troca de ACKs e até perda de pacotes por desvanecimento do sinal, com retransmissão automática de pacotes perdidos.

A partir de uma análise automatizada da Rede de Petri pelo software ARP (MAZIERO, 1990), são avaliados todos os loops e estados possíveis desta rede, analisando-se, então, os pontos críticos de acessibilidade e invariância necessários a um sistema estável e reinicializável de rede.

O resultado completo da avaliação pode ser analisado no Anexo C. Nos resultados observa-se que há na rede trinta e três estados distintos, sendo a rede binária, permitindo apenas uma ficha em cada lugar (não há acúmulo de fichas), não sendo, porém, estritamente conservativa. A rede também é considerada viva, e reinicializável sempre para o estado M0.

Não foram detectados *dead-locks* nem *live-locks*. A análise de desempenho também indicou que, com os possíveis disparos de transições, apenas 0.05% foram disparos improdutivos, necessários à avaliação da rede, sendo a modelagem por Redes de Petri simples (sem temporização) o causador deste indicador.

Não foram observadas também transições obrigatórias e/ou proibidas, garantindo total flexibilidade dos serviços prestados. Toda a definição dos termos aqui descritos podem ser observados na literatura (LISBOA, 2003).

Com isto, todo o modelo teórico pôde ser validado, garantindo que a máquina de estados idealizada possui funcionamento correto e que pode ser implementada em um sistema de simulação e um dispositivo microcontrolador para sua avaliação de performance.

Desta forma, o sistema será posto a prova em um ambiente de simulação, para validação de sua adoção em massa, e, posteriormente, após a sua validação de projeto, será feita a implementação em um sistema real, com testes e resultados práticos.

4. Simulação

4.1. Descrição do problema

A fim de validar o funcionamento da rede com um número maior de dispositivos funcionando de forma simultânea, faz-se necessária a utilização de ferramentas computacionais de simulação para garantir que sua integridade, em cenários de maior complexidade, será mantida.

Tal abordagem foi conferida ao projeto pelo ensejo de se obter uma extrapolação na utilização da rede, sem que fosse necessário o desenvolvimento, construção e montagem para a depuração de erros na plataforma de *hardware*, servindo de base para o funcionamento correto do objeto de estudos deste trabalho.

Desta forma, pode-se validar diversos cenários, buscando-se fragilidades na implementação da arquitetura, com baixo custo e grande velocidade, acelerando o processo de depuração e trazendo melhorias no resultado do trabalho, e reduzindo a complexidade de um teste de bancada.

4.2. Escolha do módulo de simulação

Observa-se na literatura (ROCHOL, et al., 2003), que existem algumas soluções para simulação de redes. Dentre elas, pode-se citar:

- **TeleSim (ATM-TN)** – *Calgary University, Canada*
- **ATM/HFC** – *Network Simulator NIST*
- **Network Simulator 2 / 3 (NS)** – *UCB/LBNL*
- **GloMoSim** – *Global Mobile Information System Simulation Library*
- **NCTUns 1.0** – *National Chiao Tung University of Taiwan*

Cada plataforma de simulação possui características intrínsecas em relação à flexibilidade de topologias e componentes da rede, linguagem de programação, sistema operacional de suporte, código aberto/fechado, produto comercial, entre outras.

Dentre estes, o simulador de redes mais difundido pela comunidade científica é o NS-2. Esta ferramenta fornece um amplo suporte para avaliação de desempenho dos

protocolos IP (TCP, UDP, RTP, FTP, HTTP etc.) com tráfego de dados em taxas constantes ou variáveis. Todo o projeto utiliza a linguagem C++ e roda em cima da plataforma LINUX/UNIX, sendo possível, através de *scripts* TCL (*Tool Command Language*), inserir agentes ao cenário de testes, e operar em diferentes tipos de redes, tais como: *Frame Relay* e ATM (ROCHOL, et al., 2003).

Porém, inserir novos tipos de agentes e protocolos demanda alta complexidade e o entendimento a fundo de sua arquitetura requer experiência prévia com suas ferramentas. Há algumas iniciativas na comunidade científica para, por exemplo, inserir redes *Bluetooth* no NS (não suportado nativamente) (WANG, et al., 2003; MIT BLUWARE, 2002), e de fato, após uma análise aprofundada, percebe-se que construir módulos para o sistema NS, e ainda direcionar sua tecnologia de rede para o *smart grid*, não é, de fato, algo trivial.

Simular tarefas relacionadas ao *smart grid*, tais como: medição intervalar, e acionamento remoto de cargas, não são suportados dentro dos simuladores avaliados e utilizados pela comunidade científica. Vale ressaltar que, por se estar criando um novo padrão de comunicação, voltado exclusivamente às tarefas de redes elétricas inteligentes, torna-se interessante o desenvolvimento de uma **nova plataforma** de simulação, que suporte adendos futuros, voltada para os diferentes cenários de uma rede em *smart grid*, onde se consiga modelar, de forma eficiente e realista, as adversidades deste ambiente.

4.3. Simulador

A partir dos argumentos apresentados na Seção 4.2, desenvolveu-se então um novo simulador, em linguagem C++, com o intuito de suportar futuros *upgrades* de equipamentos, tecnologias e funcionalidades. Desta forma, em sua construção, foi utilizada a orientação a objetos com classes virtuais. Assim, o *software* possui uma arquitetura genérica, suportando futuros adendos, e implementou-se apenas os módulos relacionados a este trabalho.

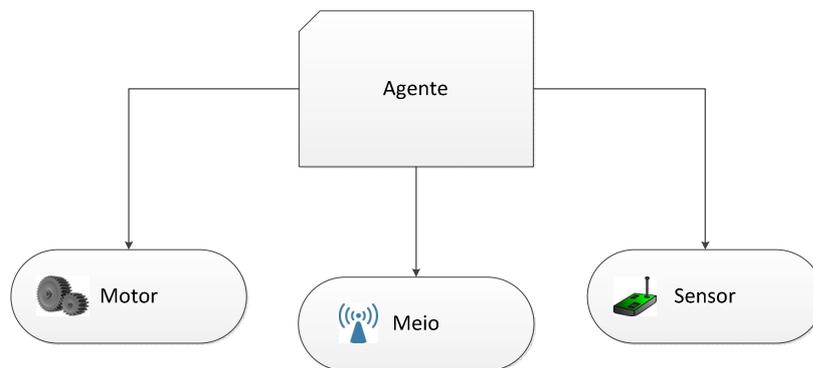


Figura 44 – Esquema de classes para agentes do sistema.

Todo o sistema é baseado em um modelo de simulador de eventos discretos, onde um evento inicial (semente) gera eventos subsequentes, inserindo-os em uma fila ordenada no tempo, que serão executados em um momento futuro, até o tempo máximo designado.

Como se pode observar na Figura 44, a classe “Agente” é uma classe virtual base a todos os agentes do sistema. Para o caso específico deste simulador, foram implementadas as classes “Motor”, responsável pelo funcionamento e gerenciamento do tempo de simulação, lista de eventos e execução dos métodos a serem simulados; a classe “Meio” que representa o meio físico (no caso o ar) de transporte dos dados e também o local onde os sensores são armazenados em um ambiente bidimensional [X,Y]; e a classe “Sensor”, que implementa todos os métodos e classes relacionadas aos sensores e nós que estarão trocando dados e mensagens durante o processo de simulação. Seu diagrama de classes detalhado pode ser avaliado na Figura 45.

Cada evento inserido na lista de eventos é executado pelo motor através do método virtual “executar” da classe base “Evento”, mostrada na Figura 46. Todo o código de execução é implementado nos eventos que herdam da classe base, sendo, portanto, essencial a escolha da arquitetura orientada a objetos, pois é possível a qualquer momento adicionar eventos distintos ao trabalho sem que seja necessário alterar eventos previamente codificados.

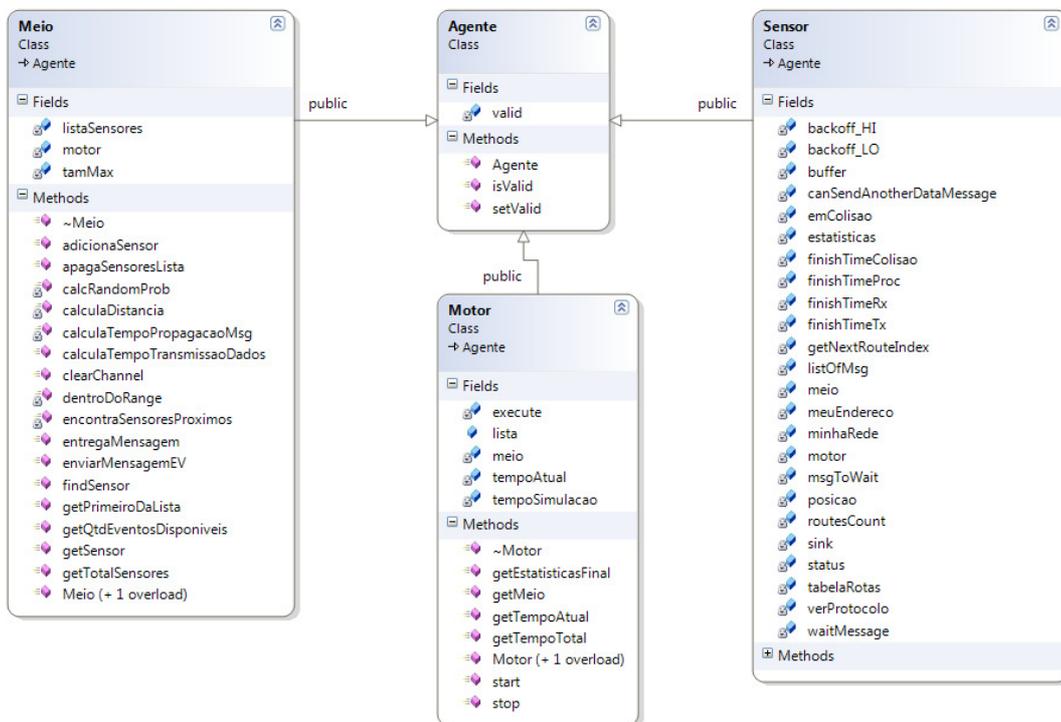


Figura 45 – Diagrama de classes dos agentes do simulador.

De forma a complementar a estrutura de troca de mensagens, outras classes foram desenvolvidas para que a troca de mensagens e os dados estatísticos do sistema pudessem ser computados em tempo de execução do programa.

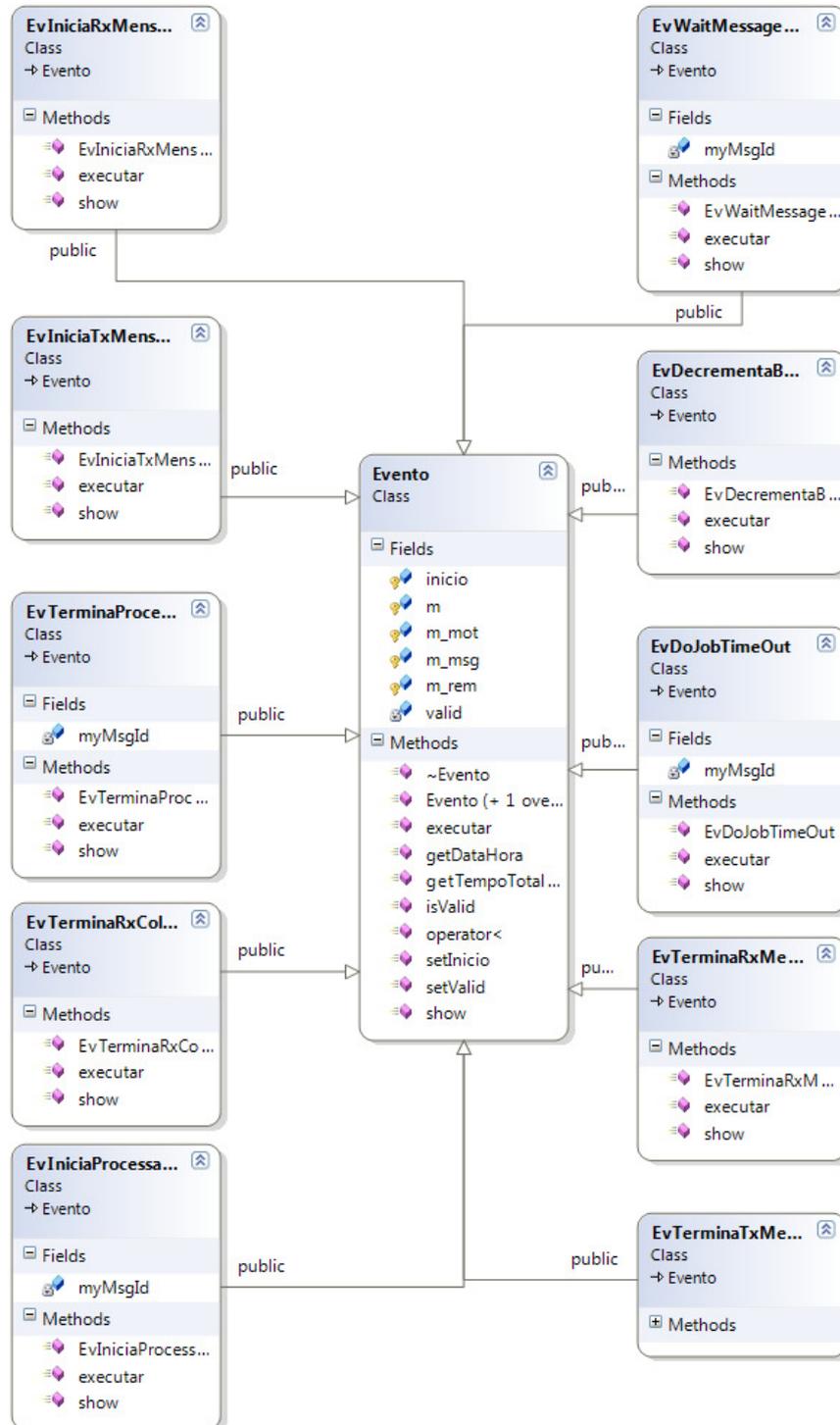


Figura 46 – Diagrama de classes dos eventos do simulador.

Estas estruturas, explicitadas na Figura 47, são de fundamental importância para o funcionamento das classes derivadas dos “Agentes”, onde se precisa guardar uma grande quantidade de dados estatísticos, eventos, rotas e mensagens que serão utilizadas para, ao final do processo de simulação, avaliar o desempenho das redes em distintos cenários.

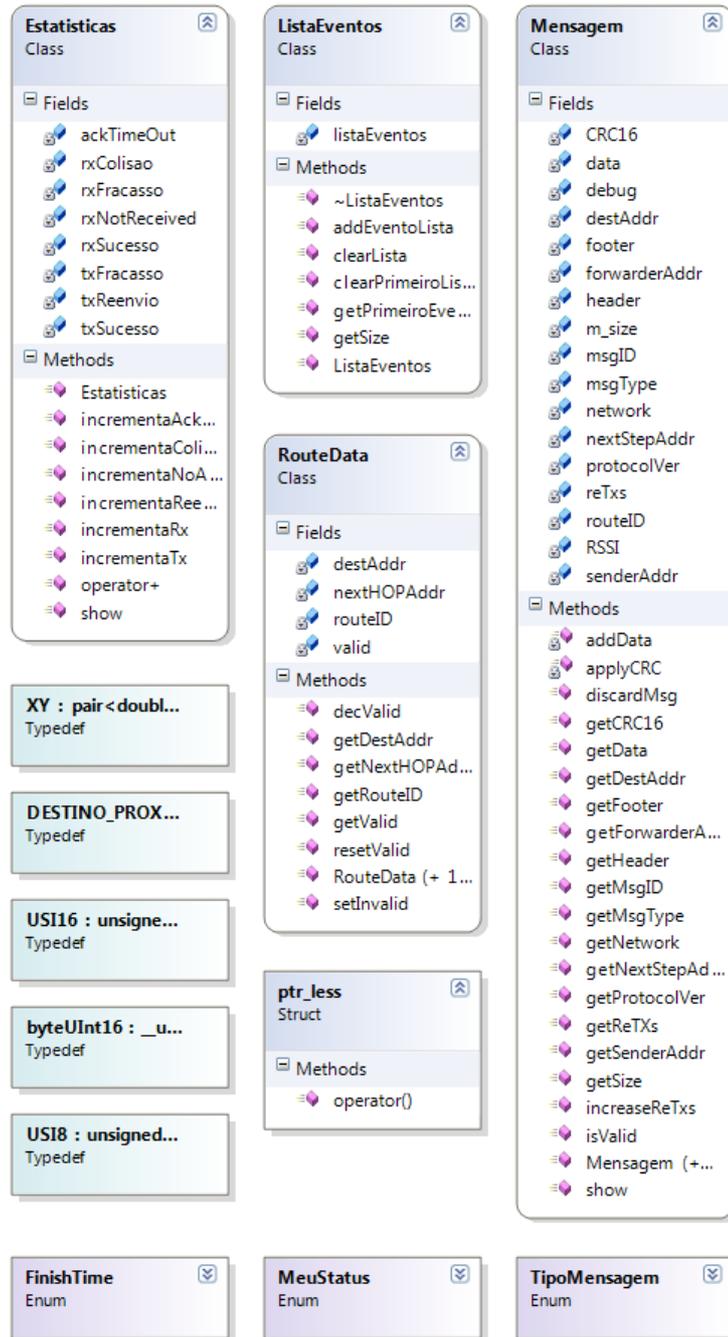


Figura 47 - Diagrama de outras classes e estruturas de dados do simulador.

Dentre os métodos mais significativos de cada classe, podemos citar:

- **Motor**

- *start* – Um *loop* responsável por buscar o primeiro evento da lista ordenada e executá-lo, até que não existam mais eventos na lista, ou que o tempo para execução do evento seja maior que o instante de tempo máximo definido para a simulação;
- *stop* – Método responsável por parar a execução do método *start*, através de uma variável disponível em memória compartilhada;
- *getEstatisticasFinal* – Responsável por compilar os dados de todos os sensores a serem apresentados ao usuário no final da simulação.

- **Meio**

- *adicionaSensor* – Método capaz de adicionar em um ambiente bidimensional, com posicionamento (X,Y), um novo sensor para a simulação;
- *calculaDistancia* – É responsável por computar a distância linear entre dois sensores dentro da área de trabalho;
- *calculaTempoPropagacaoMsg* – A partir da velocidade da luz, e da distância entre dois sensores, este método computa a duração do tempo de transmissão no ar de uma mensagem quando enviada de um sensor a outro;
- *calculaTempoTransmissaoDados* – Baseado na taxa de transmissão de dados e a quantidade de *bytes* a serem transmitidos, este método calcula em quanto tempo o sensor receptor receberá os dados;
- *clearChannel* – Retorna, para um determinado sensor, se há algum nó transmitindo dados dentro do alcance da rede. Para o sensor, seu retorno indica o funcionamento virtual do recurso de CCA, descrito na Seção 3.4 (envio de mensagens);
- *enviarMensagemEV* – É utilizado pelo sensor emissor ao transferir para o meio “físico” o pacote de dados, para que seja entregue ao destinatário. Todos sensores, que, porventura, possam estar próximos, receberão os dados depois do tempo *t*, calculado a partir dos métodos *calculaTempoPropagacaoMsg* e

calculaTempoTransmissaoDados, e com a inserção de eventos de recepção para os tempos calculados;

- *entregaMensagem* – Este método decide, baseado na distância d entre os sensores qual a probabilidade da mensagem ser entregue aos nós receptores, dada pela equação 4.3.1. Esta equação foi gerada a partir dos dados do fabricante do *transceiver* de rádio.

$$P(d) = \left(\frac{0.8}{1-ALCANCE} \right) * d + \left(1 - \left(\frac{0.8}{1-ALCANCE} \right) \right) \quad (4.3.1)$$

Com o alcance médio do rádio avaliado em 30m (especificações do fabricante), a curva $P(d)$ se comporta de acordo com a Figura 48, fornecendo, por exemplo, taxas de entrega de aproximadamente 60% à 15m de distância. Vale ressaltar que esta aproximação linear deve ser alterada para uma função exponencial em versões futuras do simulador, proporcionando maior realismo com métodos de desvanecimento de sinal, tais como FSPL (*Free Space Path Loss*), entre outros (JONES, 2003).

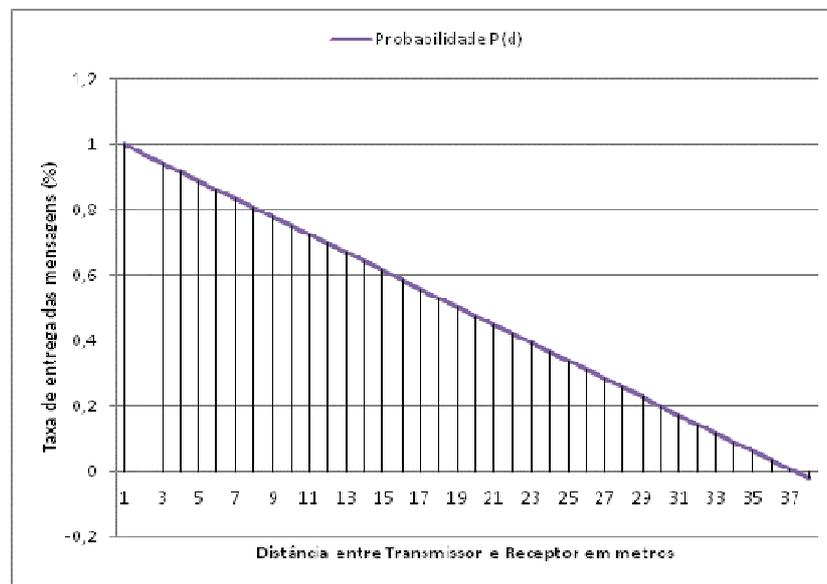


Figura 48 – Probabilidade $P(d)$ para um alcance máximo de 37m.

- **Sensor**

- *addRoute* – Ao receber uma mensagem, o sensor executa este método para adicionar o remetente à sua lista de caminhos diretos, caso ele ainda não esteja presente na lista;

- *calculaTempoBackoff* – Este método executa o algoritmo:

```
if (backOff ≤ MINTIME)
{backOff = MINTIME + Random()%MAXTIME}
else
{backOff = backOff - 1}
return (backOff * BACKOFFPERIOD)
```

Desta forma, é retornado o valor do *backOff*, em picossegundos, a ser adicionado ao próximo evento para a transmissão efetiva da mensagem no meio. Esta abordagem está presente no método de CSMA/CA (FEDERAL STANDARD 1037C, 1996), descrito na Seção 2.2, porém o algoritmo para escolha do valor de *backOff* é diferenciado para procurar reduzir o monopólio do meio, e consequente domínio dos critérios de concorrência na disputa pelo uso do recurso compartilhado (meio físico). Vale ressaltar que este procedimento não é o utilizado pelo IEEE 802.15.4;

- *clearChannel* – Este método busca no “Meio” se há algum dispositivo transmitindo dentro do alcance do seu transmissor;
- *enviarMensagemEV* – É utilizado pelo sensor no momento de transmitir ao meio os dados que serão enviados ao circuito simulado de rádio. Este método busca, através da função *clearChannel* se há como entregar a mensagem. Em caso positivo, usa o método *calculaTempoBackoff* e, após aguardar o valor de retorno, a mensagem é disparada, se o canal ainda estiver disponível;
- *findDestbyAddr* – Na iminência da transmissão de dados, o sensor deve verificar se o endereço do destinatário está presente em sua

tabela de rotas. Para isto, este método retorna a rota (com o valor do próximo salto) para que haja a garantia da entrega dos dados. No caso da rota não estar presente, o sensor emitirá uma mensagem para requisição de rotas, conforme descrito no fluxograma da Figura 39;

- *findNextFreeAddrInRouteTable* – Este método é o responsável por encontrar um endereço vazio na tabela de rotas e/ou remover endereços para que uma nova rota seja sobrescrita. Diversas técnicas para a escolha do item a ser removido poderiam ser utilizadas aqui, e, ainda podem ser aprimoradas em trabalhos futuros. Nesta primeira versão, no caso da “tabela de rotas” apresentar-se cheia, o dispositivo indica o primeiro elemento da lista como o próximo endereço a ser sobrescrito;
- *gerarMensagem* – A partir dos dados recebidos pela tabela de rotas, tipo de mensagem a ser enviada e os dados a serem transmitidos, este método constrói, baseado no diagrama de dados estipulado pelo protocolo descrito na Figura 33, um bloco de dados a ser transmitido pelo sensor. Este algoritmo cria um objeto da classe “Mensagem” que será utilizada por todo o processo de disparo de eventos de transmissão e recepção de dados entre os dispositivos;
- *setDoJob* – É executado quando uma mensagem recebida é do tipo atividade/tarefa, conforme descrito na Seção 3.3. Desta forma, a mensagem é gravada em um *buffer* interno que será utilizado assim que o evento de processamento for marcado (neste caso, a execução dar-se-á exatamente após o momento em que a mensagem for aceita pela camada de enlace).
- *testProtocol* – Este método é executado assim que há o recebimento de uma mensagem, validando se o pacote poderá subir para a camada superior do protocolo. Caso a mensagem seja para outro dispositivo, ou contenha algum campo fora de conformidade e/ou inválido, ela é então descartada por este método, cumprindo o papel das camadas inferiores de rede.

- **Evento**

- *executar* – Este método é virtual. Todas as classes que herdam de Evento, são responsáveis por sua implementação, visto que, *a priori*, podem existir diversos eventos com codificações distintas. Desta forma, este método é executado pelo motor a partir de uma chamada externa, por ponteiro de função, conforme se observa na Figura 49.
- *show* – Este método é virtual. Possui o descritivo e o tipo do evento, algo que o motor desconhece até o momento da execução, e retorna, através de uma estrutura do tipo *std::string*, o resultado da execução do trecho de código. Desta forma, é possível conhecer o estado atual dos sensores e mapear, passo a passo, o andamento da simulação, em modo de depuração.

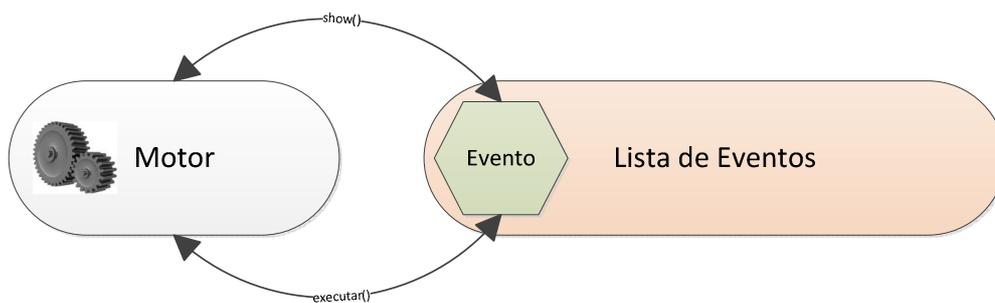


Figura 49 – Relacionamento entre Motor e Eventos do simulador.

Vale ressaltar que, após a execução, com sucesso ou fracasso, o evento é removido da lista dando lugar ao seu sucessor. Desta forma, o motor atualiza sua referência de tempo a partir da data e hora do evento programado, e garante a execução linear e o andamento correto de eventos a serem simulados. Esta arquitetura, infelizmente, não permite a sua paralelização em um *software multithread*, porém, para efeitos didáticos apresenta a funcionalidade, e requisitos compatíveis com o assunto deste trabalho.

4.4. Cenários

Diversas propostas e cenários podem servir de base para validar o funcionamento deste trabalho. Para o caso específico desta dissertação, foi pensado um cenário onde há um dispositivo mestre (*sink*), disposto no meio do *grid* de testes, e outros N sensores, igualmente espaçados angularmente, com um raio comum R, conforme ilustrado na Figura 50.

Desta forma, com um ângulo variando entre 0 e 2π , os sensores são dispostos no *grid* com $\frac{2\pi}{N}$ radianos a uma distancia R, no caso do primeiro cenário $R=8m$. Para o valor de N, consideram-se valores no intervalo de 1 a 15, para uma primeira avaliação, extrapolando o cenário adotado na Seção 3.2.1, e ilustrado na Figura 32, com uma margem de 50% de sensores extras convivendo em um único ambiente e compartilhando recursos de um mesmo canal de comunicação.

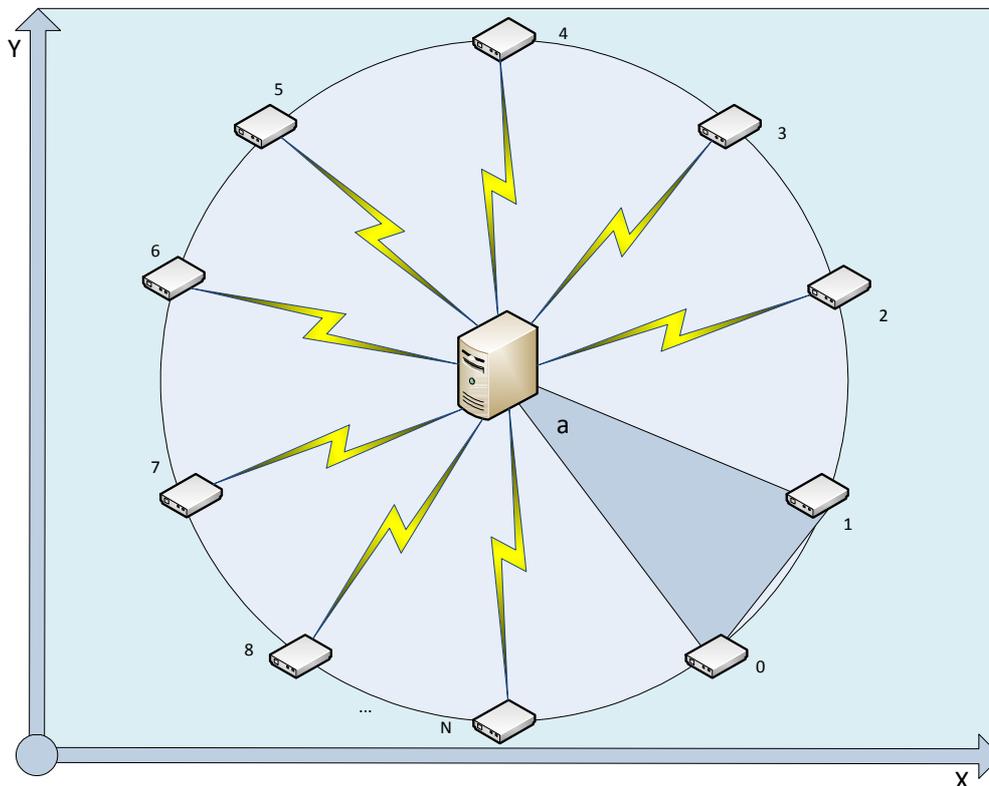


Figura 50 – Esquema proposto para o cenário de simulação com N sensores.

Com tal abordagem, é possível aferir e mensurar as taxas e a capacidade de transmissão máximas dos sensores, a interferência e o número de colisões causadas pelo aumento no número de sensores, a taxa de entrega de mensagens por desvanecimento de sinal a partir do número de retransmissões, e, a partir destes resultados, computar métricas para validar se o modelo conceitual de protocolo deste trabalho consegue atender à demanda antes de se iniciar o processo de construção do *hardware*.

Como objeto deste trabalho, a métrica padrão adotada, além da taxa máxima de transmissão (*throughput*), denomina-se índice de *Raj Jain* (JAIN, 2012), também conhecido como *Fairness Index*. Desta forma, é possível verificar se, ao aditar novos sensores na simulação, o meio é utilizado de forma a compartilhar os recursos de forma justa, garantindo taxas de vazão máxima equivalentes entre os dispositivos. Tal índice pode ser obtido através da equação 4.4.1.

$$\mathfrak{J}(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n * (\sum_{i=1}^n x_i^2)} \quad (4.4.1)$$

Onde existem n nós, cada nó com sua respectiva taxa de vazão de dados máxima x_i (*throughput*), sendo que o resultado pode variar de $\frac{1}{n}$, até 1 (no melhor caso).

4.5. Metodologia de testes

Inicialmente, visando validar um cenário real de rede elétrica inteligente, procurou-se utilizar na metodologia de testes uma arquitetura de *polling* para a troca de mensagens de atividade/tarefa, partindo sempre a iniciativa do mestre, de forma sequencial para os outros sensores em sua tabela de rotas. Desta forma, o nó pai (*sink*) faz um pedido de dados para os nós filho, e recebe em seguida a resposta, todas estas com confirmações por ACK, conforme previamente descrito no fluxograma da Figura 38.

Da mesma forma, toda a sinalização da rede é feita por cada sensor de forma independente, com temporização pseudoaleatória, em *broadcast*, conferindo ao teste características de uma situação real, onde há risco de colisão e utilização do recurso de CCA, caso os sensores optem simultaneamente por um mesmo valor de *backoff* escolhido em uma troca de mensagens entre o *sink* e outro nó da rede.

Nota-se que sensores diametralmente opostos, possuirão entre si uma menor chance de detecção dos sinais de transmissão, aumentando-se também a probabilidade de colisão entre os seus pacotes nos respectivos receptores.

Para este experimento, após uma série de cálculos e premissas de escalonamento de tarefas, utilizaram-se os seguintes valores padrão para o cenário de simulação:

- Período de *backoff* = **150 μ s**
- Janelamento de *backoff*:
 - Prioritária mínima = **1**
 - Prioritária máxima = **3**
 - Normal mínima = **4**
 - Normal máxima = **9**
- Velocidade da luz = **299792458 m/s**
- Intervalo entre *beacons* de sinalização = **1s**
- Intervalo para decrementar o contador de uma rota = **500ms**
- Tempo máximo de espera de um *beacon* para remoção da tabela = **4s**
- Tempo de processamento de uma requisição no receptor = **625ns**
- Raio = **8m**

- Intervalo intrapedido de mensagens = **10ns**
- Taxa de transferência = **250Kbps**
- Número máximo de tentativas de retransmissão = **10**
- *Timeout* para retransmissão de mensagem (*no Ack*) = **28,5ms**
- *Timeout* para retransmissão do pedido (*no Answer*) = **85,5ms**
- Quantidade de dispositivos – **2 a 16 sensores**
- Tempo do teste – **30 minutos**
- Número de simulações por cenário – **3**

Ressalta-se que os valores de temporização (*timeout*) são baseados no tempo para se transmitir uma mensagem com tamanho médio de 64 *bytes* e receber uma resposta ACK válida. Para este cálculo tem-se, em média, que:

$$T_{tx} = \frac{64 \cdot 8}{250000} = 2048 \mu s$$

(para cada mensagem transmitida)

$$T_{backOffPrioriMax} = 3 * 150 \mu s = 450 \mu s$$

(tempo máximo para o ACK ser transmitido)

$$T_{backoffMax} = 9 * 150 \mu s = 1350 \mu s$$

(tempo máximo para o pedido ser transmitido)

Desta forma tem-se que:

$$T_{ackMax} = ReTx_{Max} * (2048 \mu s + 450 \mu s) = 10 * (2498) = 24,98ms$$

De forma análoga, tem-se para a mensagem a ser transmitida:

$$T_{msgMax} = ReTx_{Max} * (2048 \mu s + 1350 \mu s) = 10 * (3398) = 33,98ms$$

Obtém-se então o tempo total pela equação 4.5.1.

$$T_{total} = T_{ackMax} + T_{msgMax} \quad (4.5.1)$$

$$T_{total} = 58,96ms$$

De forma a evitar grandes probabilidades de descarte, estipulou-se então, um valor médio de 28,5ms (valor médio de T_{total}) para $T_{timeoutTx}$, propiciando assim, caso a mensagem de ACK não seja recebida a tempo, um valor suficiente para considerar falha na entrega da confirmação.

Analogamente, o valor determinado para retransmissão do pedido (*no Answer*) foi escolhido como três vezes o valor da temporização de retransmissão de uma mensagem, tendo em vista que o processo completo é definido pela equação 4.5.2, porém, como forma de garantir que possíveis atrasos não penalizem a transmissão de dados, foi adicionada uma margem de segurança ao processo.

$$T_{noAnswer} = 2 * T_{total} + T_{Processamento} \quad (4.5.2)$$

Vale ressaltar que, no protocolo base deste trabalho, o tempo de espera não segue o princípio do ‘*backoff* exponencial binário’, ou BEB, onde o valor da janela é calculado a partir de $(2^N * T_{backoff})$. O valor do intervalo de *backoff* é calculado por $(N * T_{backoff})$. Desta forma, propõe-se testar um algoritmo que visa aprimorar o tempo de espera de disputa, reduzindo o seu valor médio.

Outro ponto importante para este cenário de simulação é que, para esta primeira versão do simulador, não foram utilizadas configurações com múltiplos saltos. Esta abordagem deverá ser incluída em trabalhos futuros para que sejam testados, em maior nível de detalhe, sua confiabilidade e performance. Desta forma, a tabela de rotas passou a receber 15 endereços, preenchidos de acordo com o recebimento de sinalizações, o que limitou o valor máximo de dispositivos deste teste para 16 sensores.

4.6. Resultados obtidos

A partir da metodologia apresentada na Seção 4.5 (media de três testes completos com duração de 30 minutos), obteve-se os dados das simulações para a capacidade de transmissão e, conseqüentemente, o cálculo do *Fairness Index* com a metodologia de Raj Jain (JAIN, 2012).

Espera-se encontrar, inicialmente, uma taxa de transmissão de dados máxima próxima à metade da taxa de transmissão máxima de dados, visto que o processo de disputa do meio, com a utilização de CCA e *backoff*, possui a mesma ordem de grandeza do tempo de transmissão da mensagem. Ao adicionar um número maior de dispositivos, este valor deve ir se aproximando à taxa de transmissão máxima.

De forma análoga, espera-se que o índice de Raj Jain seja muito próximo de 1 (um), independentemente da quantidade de dispositivos a ser utilizada, dada a natureza sequencial adotada na metodologia de testes e à otimização do protocolo para ocupar pouca banda com sinalização.

4.6.1. Transmissão de dados

Após a execução dos testes, observa-se na Figura 51 o comportamento da taxa média de transmissão de dados, ao longo de todo o período de testes, para diferentes configurações e quantidade de dispositivos, seguindo a metodologia proposta em 4.5.

Conforme esperado, a taxa de dados total do sistema aumenta conforme são adicionados novos dispositivos, até um ponto onde, aparentemente, há uma desaceleração do crescimento da taxa de dados. Com o limite imposto pela metodologia, de 16 dispositivos, a taxa máxima média obtida foi de, aproximadamente, 135Kbps.

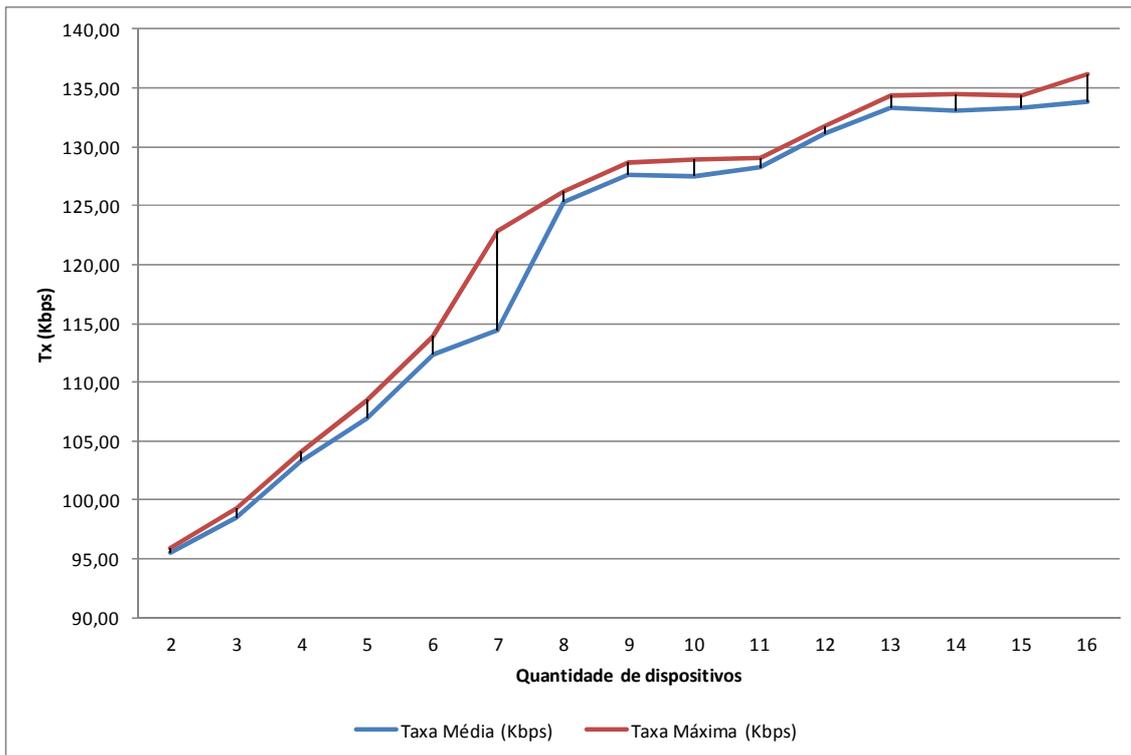


Figura 51- Capacidade de transmissão de dados do sistema, em Kbps, ao aumentar-se a quantidade de dispositivos da rede.

Conforme observado na Figura 52, a taxa de pacotes reenviados por ocupação do meio aumenta conforme são adicionados mais dispositivos ao sistema. Isto se deve, principalmente, ao fato de que quanto mais sensores compartilhando o mesmo recurso, maior será a disputa do mesmo. Os resultados mostram taxas de até 10% dos pacotes reenviados, em relação ao total de pacotes entregues ao destino de forma adequada.

Analogamente, na Figura 53, observa-se o número total de colisões observadas durante o tempo total de duração dos testes. Estes valores estão relacionados às situações de terminal escondido e/ou exposto devido ao desvanecimento de sinal, com maior probabilidade de ocorrer em sensores diametralmente opostos. Verifica-se que este número é muito baixo quando comparado ao total de mensagens enviadas, representando apenas 0,0037% do total trafegado pela rede.

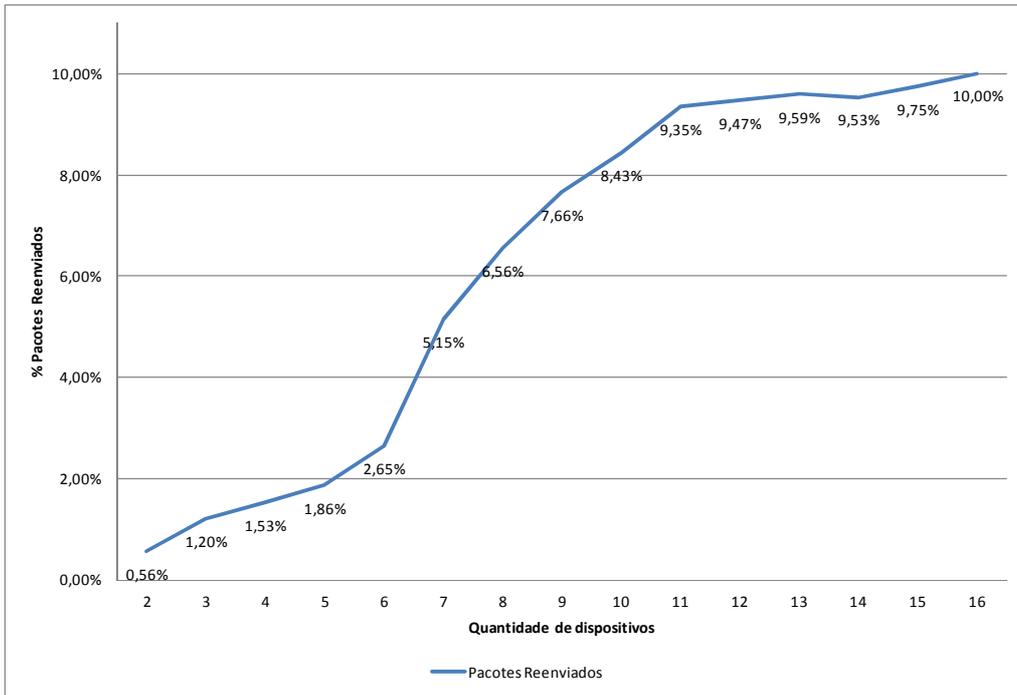


Figura 52 – Taxa percentual de pacotes reenviados.

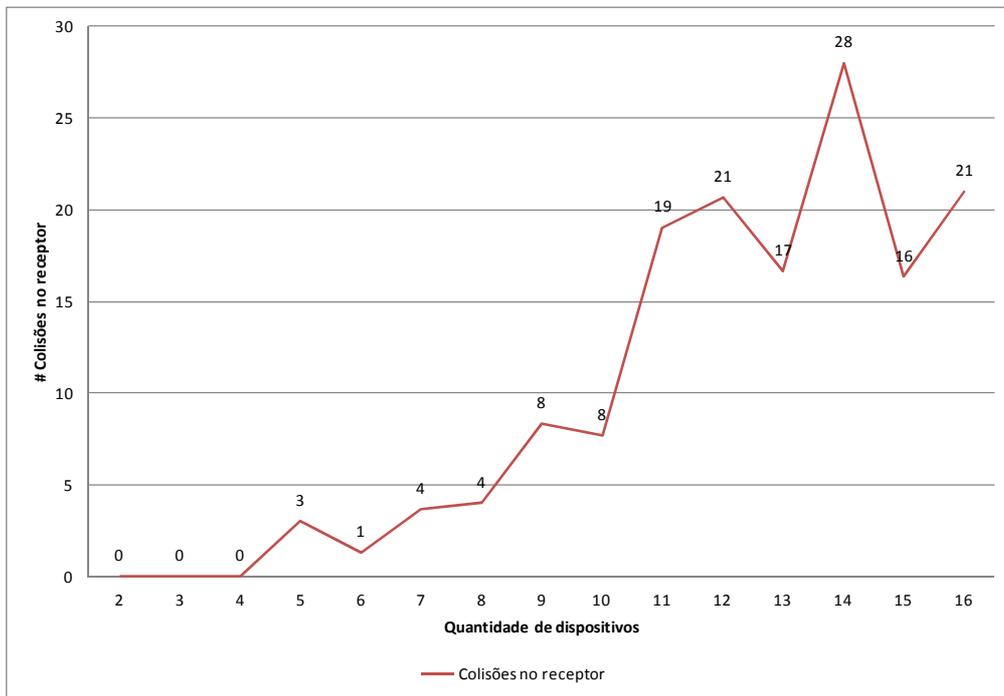


Figura 53 – Somatório do número total de colisões no receptor dos dispositivos.

A fim de quantificar qual o grau de confiabilidade da camada de rede na busca pelas informações desejadas dos sensores filho, recolheram-se também dados sobre a taxa de tarefas/atividades que não conseguiram receber os dados referentes ao pedido, ou seja, não houve o recebimento de dados compatível com o pedido executado.

Conforme observado na Figura 54, a camada de rede não conseguiu concluir a tarefa de buscar os dados em **0,017%** dos pedidos, no pior caso, obtendo um resultado bastante satisfatório. Isto implica que todas as tentativas para o reenvio do pedido de dados foram esgotadas e a operação foi então abortada, seguindo para a operação subsequente, ou, entrando em modo de espera.

Nestes casos, a camada de aplicação pode assumir os seguintes modos:

- **Refazer o pedido** – Espera-se que a aplicação jamais deva perder um pacote de dados. Assim, a aplicação, que utiliza os serviços de rede para receber os dados, ou o usuário, devem então refazer o pedido após a falha indicada no recebimento;
- **Descartar o pacote** – Para operações em *smart grid*, acredita-se que esta opção seja a mais adequada, visto que há uma grande quantidade de dados a ser analisada. Desta forma, o descarte de poucos pacotes não trará impactos para a operação como um todo.

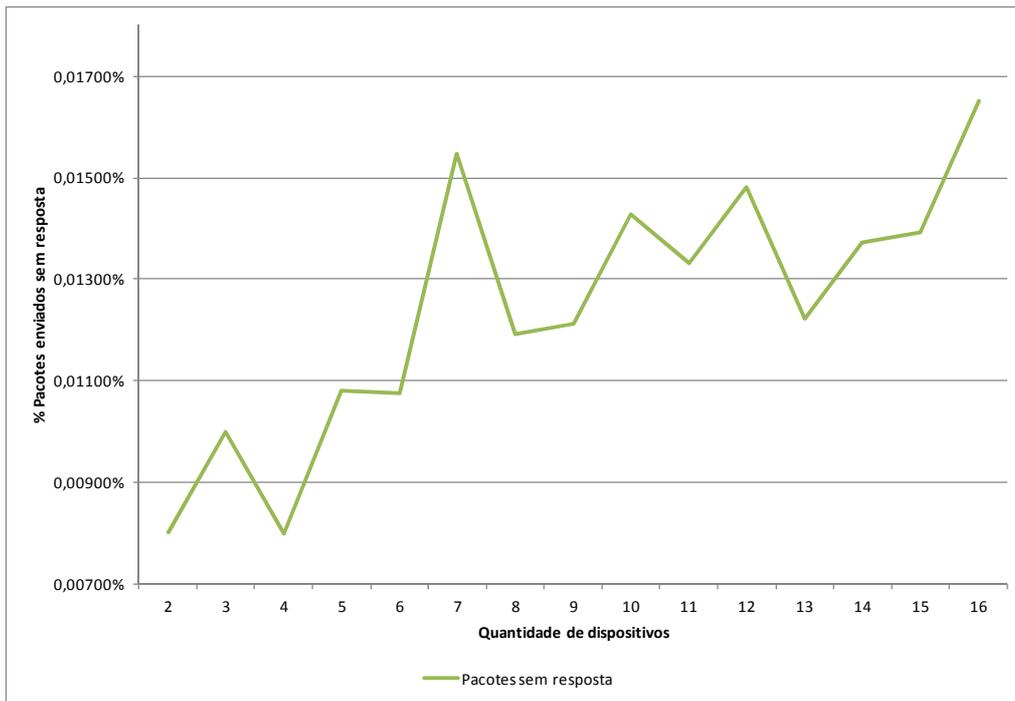


Figura 54 – Percentual dos pacotes enviados que não obtiveram resposta dos transmissores.

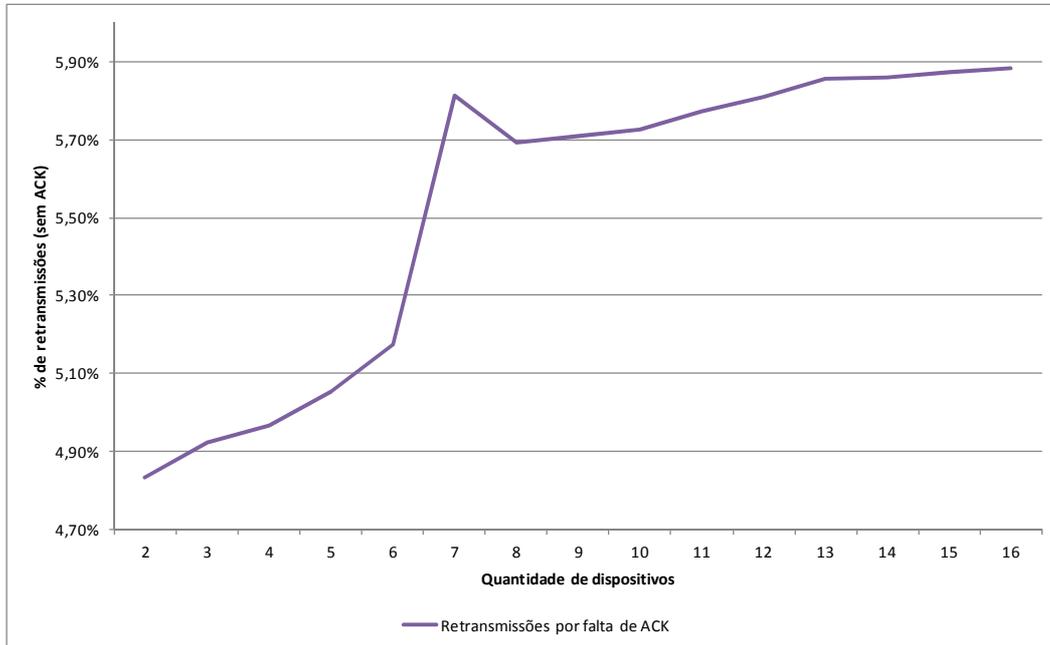


Figura 55 – Percentual de retransmissões por falta de reconhecimento (*no ACK*).

De forma a explicar mais profundamente a análise das simulações, observou-se também o percentual de retransmissões por falta de reconhecimento. Ou seja, o número de pacotes que foram reenviados por uma falha na transmissão, ou recebimento, de uma mensagem de *acknowledgment* (ACK). Assim, é acionado o *timeout* e o pedido é reenviado pela camada de rede. Vale ressaltar que, nestes casos, podem ser descritas algumas situações:

- A mensagem de pedido falhou e o receptor não enviou o ACK;
- A mensagem de pedido chegou e o receptor não conseguiu enviar o ACK após múltiplas tentativas;
- A mensagem de pedido chegou e o receptor conseguiu enviar o ACK, porém o mesmo não chegou ao receptor do emissor do pedido original.

Assim, analisando a Figura 55, percebe-se que o número de retransmissões ocorridas somente por este tipo de *timeout* é significativo, ocorrendo em até 5.9% dos casos, ou seja, mais da metade do total de retransmissões (avaliado na Figura 52).

4.6.2. Índice de Raj Jain

Após analisar todos os dados relativos à performance da rede, verifica-se, na Figura 56, os valores para o *Fairness Index* obtidos a partir da equação 4.4.1. Por se tratar de um teste sequencial, era esperado que índice se mantivesse próximo de 100%, ou 1. De fato, observa-se que, no pior dos casos analisados, o índice ficou maior que 99,91% em todas as medições realizadas.

Este tipo de comportamento mostra que o protocolo está, de fato, sendo justo com seus membros, garantindo que todos possam “falar” na rede sem que outros sensores queiram monopolizar os recursos compartilhados, neste caso o meio físico, ar.

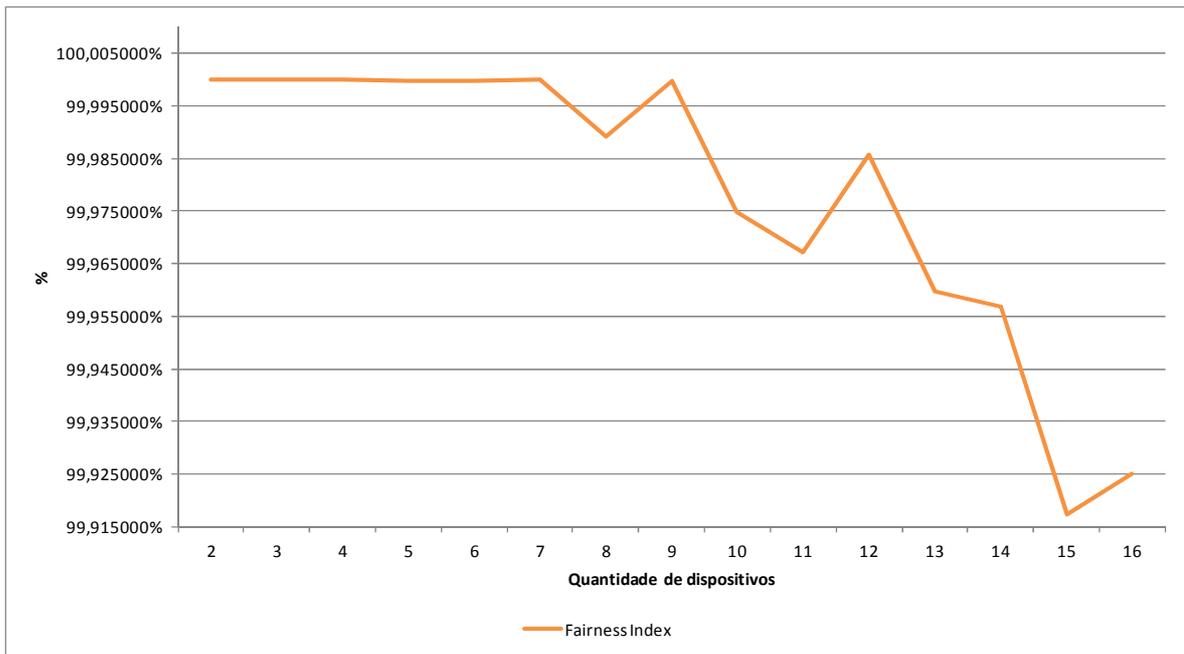


Figura 56 – Valores de *Fairness Index* quando se aumenta a quantidade de dispositivos.

Desta forma, acredita-se que o sistema, a partir deste momento, possa vir a ser implementado em microcontrolador e seus resultados, demonstrados no Capítulo 5, comparados com os obtidos por simulação para uma avaliação mais correta dos dados coletados em ambas as circunstâncias.

5. Resultados experimentais

Diante dos resultados obtidos nos Capítulos 3 e 4, faz-se necessário o desenvolvimento de um equipamento capaz de suportar o protocolo embarcado. Desta forma, utilizou-se a plataforma de hardware desenvolvida pela empresa *Powersave*, visando acelerar o processo de depuração e desenvolvimento de *hardware*. Assim, o trabalho pôde se especializar na implementação da camada de enlace e rede do protocolo em questão.

5.1. Equipamento

Todo o arranjo experimental foi baseado na plataforma, previamente descrita na Seção 3.2.1, com microcontroladores MSP430 e *transceivers* de rádio CC2500, ambos da *Texas Instruments*. A placa de circuito eletrônico, com *layout* mostrado na Figura 57, possui espaços delimitados (A e B) para a soldagem SMD dos componentes que receberão e processarão o código do protocolo de rede.

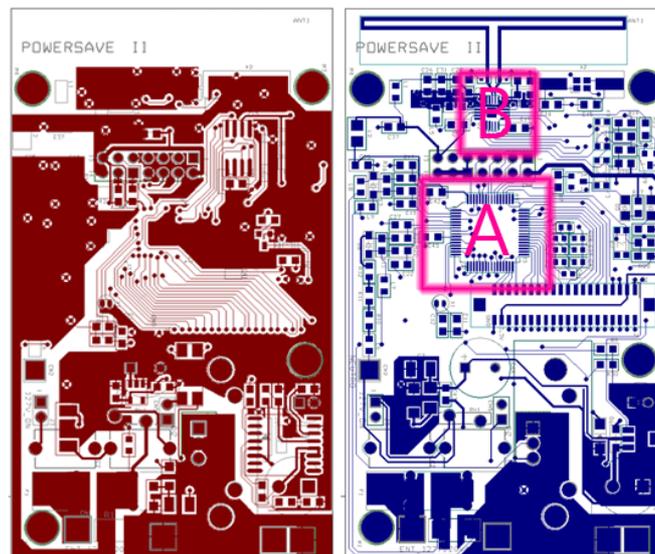


Figura 57 – Microcontrolador MSP430 (A) e CC2500 (B) na placa desenvolvida para testes (*layout* cedido pela empresa *Powersave*).

De posse dos layouts, foram fabricadas cinco placas completas, intituladas de *POWERSAVE II*, como pode ser observado na Figura 58. Estas placas serviram de base para a plataforma proposta neste trabalho. Além de baixo custo, elas apresentam interfaces para expansão das funcionalidades, como a ligação de *display* LCD, botões externos, interface de comunicação, chaves de potência, entre outros.

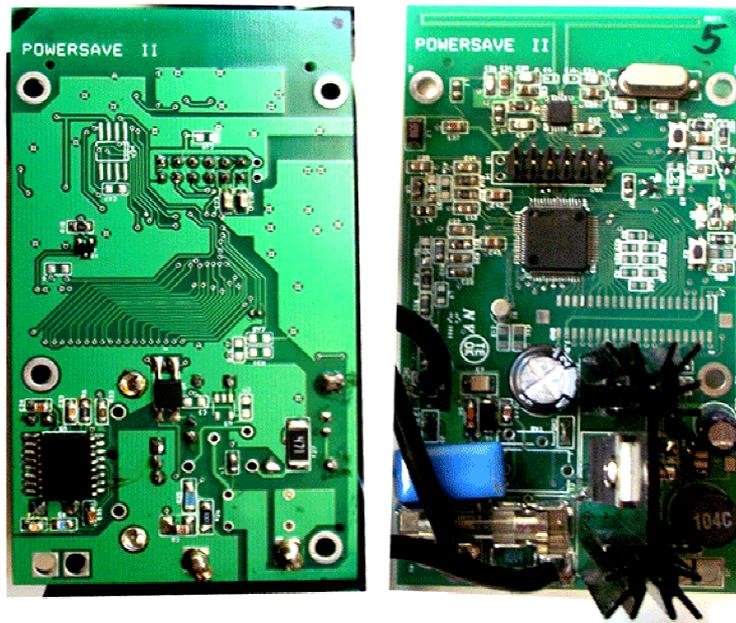


Figura 58 – Placas prontas a serem embarcadas em tomadas, disjuntores e outros dispositivos eletroeletrônicos.

Como citado anteriormente na Seção 3.2, toda a medição de energia fica a cargo de um *hardware* dedicado, chamado ESP (*Embedded Signal Processor*), que executa todo o processamento dos dados digitalizados de tensão e corrente para encontrar os valores *RMS* de tensão e corrente, potência, fator de potência e o consumo de energia acumulado. Este periférico é encapsulado dentro de um mesmo *chip*, podendo ser acionado e configurado por comandos de *firmware*.

Com relação à instrumentação, utilizou-se o circuito recomendado (SCHAUER, et al., 2009) pelo fabricante, com filtragem e adequação dos valores nominais de tensão e corrente, para o nível de sinal (3.3V) na entrada dos conversores analógico-digital.

5.2. Arranjo Experimental

De forma a buscar reproduzir os resultados obtidos por simulação, descritos na metodologia da Seção 4.5, as cinco unidades construídas foram dispostas conforme ilustrado na Figura 59. Com um raio aproximado de 8m, em um ambiente aberto (*outdoor*), sem obstáculos, a aproximadamente 1m do chão, cada sensor possuía um interruptor que era ligado após o primeiro intervalo de testes.

No sensor pai, representado por um *notebook* na Figura 60, observa-se um *hardware* especial dedicado exclusivamente a fazer o papel de *sink*. O *firmware* embarcado neste dispositivo é diferenciado e foi desenvolvido para fazer pedidos sequenciais aos nós que estão em sua tabela de rotas.

Sua construção é baseada no kit de desenvolvimento eZ430-RF2500 (EZ430-RF2500, 2009) que fundamenta-se no *transceiver* CC2500, o mesmo utilizado na placa *POWERSAVE II* e escolhido para a plataforma deste trabalho. Desta forma, compatibiliza-se a comunicação entre diferentes dispositivos microcontroladores. Após o período de um minuto, é liberado um *feedback* ao computador, através de uma porta de comunicação serial USART/USB, contendo informações sobre a quantidade de mensagens trocadas e entre quais dispositivos.

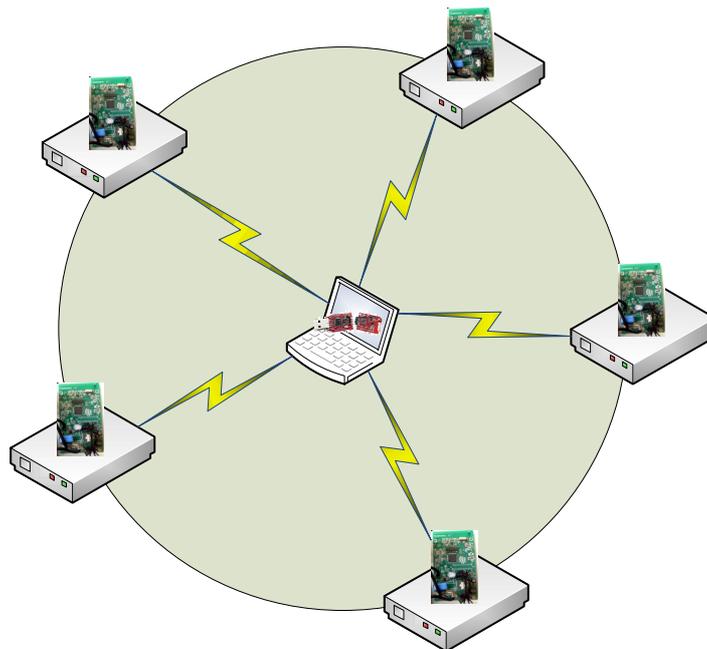


Figura 59 – Esquema experimental de comunicação.

Observa-se também na Figura 60, que o dispositivo *sink* possui duas taxas distintas de comunicação. Entre os dispositivos da rede sem fio, é utilizada a taxa padrão de 250Kbps. Porém, entre o microcontrolador do nó pai e o computador a taxa utilizada é de 9600bps, significando que, dependendo da metodologia utilizada pelo teste prático, a performance da rede poderá nivelar-se por baixo, atingindo um valor menor que o desejado.

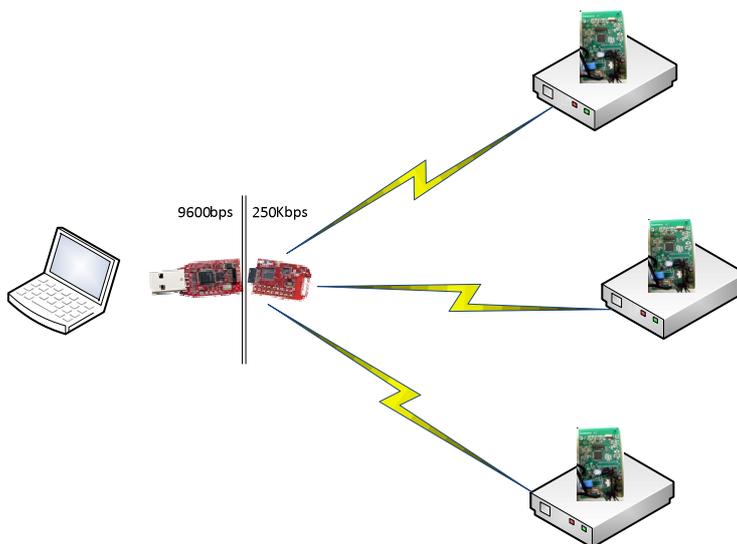


Figura 60 – Comunicação pela porta USB a partir de um módulo TI.

De forma a evitar este comportamento indesejado, priorizou-se o desenvolvimento de um *firmware* que utilizasse apenas a interface de comunicação USART/USB para notificar o andamento do processo do teste. Assim, somente em um curto intervalo de tempo o processo lento é disparado. Para cada entrada na tabela de rotas, definiu-se então um contador, em memória, capaz de armazenar toda a quantidade de interações existentes entre um nó pai e filho.

Em um momento inicial, onde somente o pai e um nó filho eram ligados, era esperado, após o recebimento de um minuto de teste válido, que o nó pai enviasse uma mensagem contendo os valores medidos no teste. O processo estaria completo até que pudesse ser determinada uma média. A partir de então, era ligado um segundo nó, descartado o minuto inicial (evitando-se efeitos transitórios indesejados), e coletado os dados do segundo minuto válido em diante. Este processo foi repetido, até que todos os cinco nós fossem ligados e o último dado válido fosse coletado para análise.

5.3. Resultados obtidos

Após a aplicação da metodologia citada na Seção 5.2, obteve-se os dados resultantes do processo, dispostos na Tabela 8. A duração de cada medição do teste foi de dez minutos, sendo o primeiro minuto sempre descartado. Desta forma, apenas nove minutos foram computados para se chegar ao resultado final apresentado.

Tabela 8 – Resultados obtidos após a coleta de dados do experimento.

Sensores	2	3	4	5	6
<i>Sensor A (Kbps)</i>	84,68	38,31	32,03	28,80	23,33
<i>Sensor B (Kbps)</i>	x	47,41	38,56	32,21	28,92
<i>Sensor C (Kbps)</i>	x	x	28,89	23,61	25,71
<i>Sensor D (Kbps)</i>	x	x	x	8,26	4,14
<i>Sensor E (Kbps)</i>	x	x	x	x	26,63
VazãoTotal (Kbps)	84,68	85,72	99,49	92,87	108,73
Vazão Média (Kbps)	84,68	42,86	33,16	23,22	21,75
Jains FI	1,000	0,989	0,985	0,865	0,854

Observa-se que o sensor com endereço D, possui claramente uma taxa de transmissão de dados menor que seus pares de teste. Este comportamento anômalo pode indicar um descasamento das capacitâncias (problema na montagem do *hardware*) para o atingimento da frequência no canal de transmissão, o que gera uma portadora levemente fora da faixa de 2,4GHz. Desta forma, o transmissor não envia nem recebe mensagens com o ganho desejado, gerando perdas nos sinais de transmissão e, consequentemente, menores taxas de transmissão de dados.

5.3.1. Capacidade de transmissão

Após uma análise dos dados apresentados, é possível descrever que o seu comportamento, em termos de velocidade máxima e ocupação do canal ilustrado na Figura 61, está sinérgico em relação aos obtidos em simulação na Seção 4.6.1. De fato, observa-se uma queda inesperada da taxa quando o sistema possui cinco sensores simultâneos, porém tal fato se deve ao possível descasamento de impedâncias, conforme citado na Seção 5.3, entre um *transceiver* de rádio e a antena, e/ou à geração de uma frequência de portadora fora da faixa de ganho máximo de sinal.

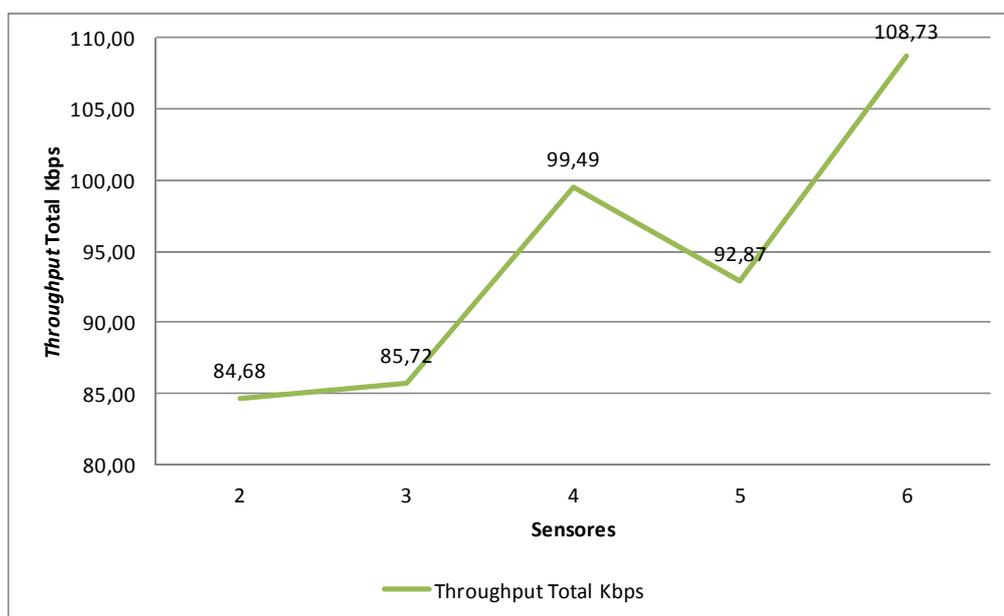


Figura 61 – Capacidade conjunta de transmissão de dados média (após três rodadas).

Avaliando o mesmo teste em um cenário individualizado, ilustrado no gráfico da Figura 62, onde são mostrados os valores médios de taxa de transmissão para cada sensor, é possível verificar que as taxas individuais se estabilizaram em torno de 20Kbps por sensor, o que, de acordo com os cálculos de taxas de transmissão teóricas para o *smart grid*, discutidos na Seção 3.2, garantem, com folgas, a transmissão de todos os dados necessários para medição intervalar, teleautomação e configuração.

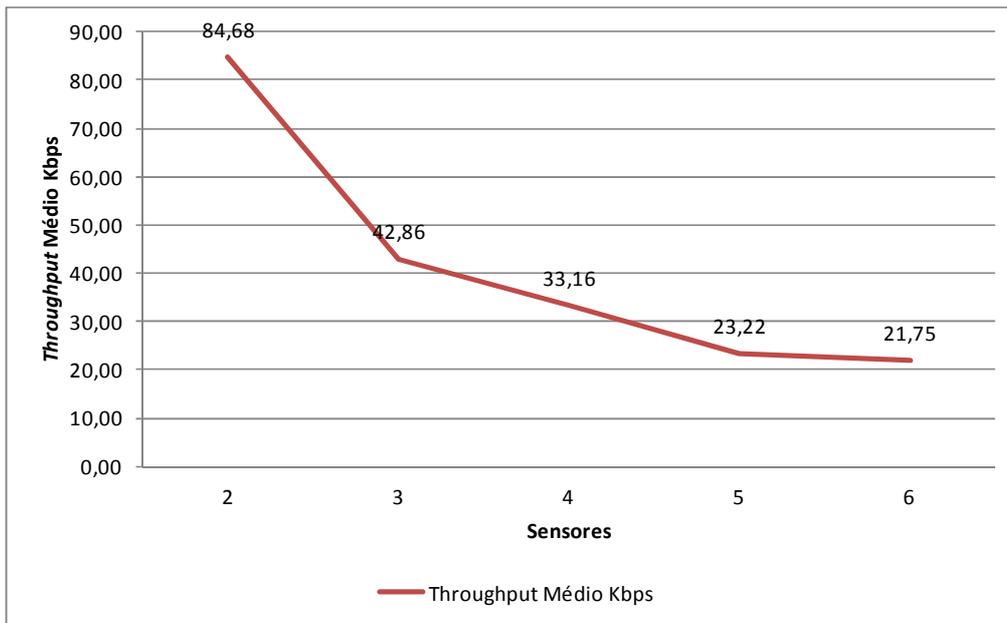


Figura 62 – Capacidade média individual de transmissão de dados.

5.3.2. Índice de Raj Jain

Como forma de avaliar o índice de justiça da rede, avaliando se há conformidade na distribuição dos recursos, assim como proposto no teste por simulação disponível na Seção 4.6.2, computou-se o índice de Raj Jain, para a validação dos resultados experimentais. Vale ressaltar que, assim como nos resultados da simulação, o índice esperado é de aproximadamente 100% (ou 1) pela natureza sequencial do método aplicado.

Conforme observado na Figura 63, o índice calculado é próximo de 100% até a entrada do sensor D, cujo comportamento anômalo, já explicitado anteriormente, interferiu negativamente para o atingimento das metas esperadas. Desta forma, ao atingir cinco sensores, o índice caiu para 86%.

Este comportamento indesejado observado deve trazer consigo fatores sensíveis ao processo antes não observados em testes por simulação. De fato, comprova-se que metas de qualidade no processo de fabricação e montagem de dispositivos em larga escala devem ser levados em consideração para que o equipamento apresente o

comportamento desejado em campo. Desta forma, pode-se mitigar o risco de que equipamentos defeituosos entrem em utilização e precisem de reparo antes de sua ida ao mercado.

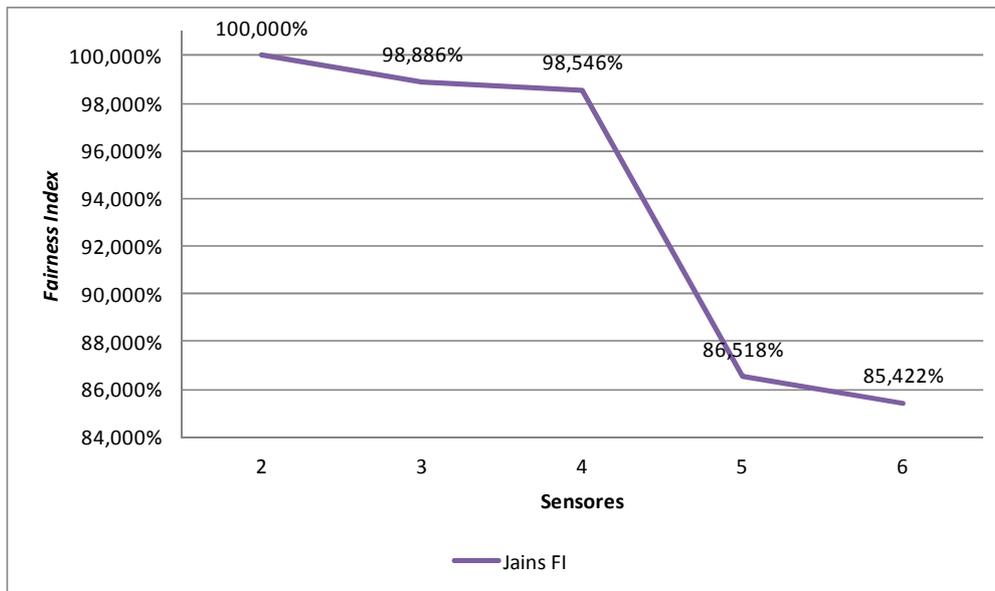


Figura 63 - *Fairness Index* para múltiplos sensores.

Porém, para o caso específico deste trabalho, o processo de fabricação e montagem não será estudado, devendo somente salientar-se que após a obtenção destes resultados observa-se que há uma sensibilidade ao risco de defeito por montagem inadequada dos componentes do circuito de rádio.

5.4. Comentários sobre os resultados

Após o desenvolvimento e implementação de uma série de testes com o protocolo objeto deste trabalho observou-se que o sistema, para uma versão preliminar, apresentou funcionamento suficiente e aceitável para funcionamento em um ambiente *smart grid*. Vale ressaltar que, após uma série de simulações e testes em bancada, a comparação dos resultados demonstra total sinergia entre os processos.

Houve pequenas discrepâncias com relação aos dados de um sensor em bancada, porém, de forma global, pode-se considerar o resultado do desenvolvimento como positivo.

A plataforma de envio de mensagens em múltiplos saltos, explicitando-se uma rede em malha, ainda carece de ajustes e aperfeiçoamentos com relação à implementação do código já pode ser vislumbrada em uma versão atual de *firmware*. Porém, sua avaliação ainda não possui metodologia científica para comparação de resultados, apresentando-se funcional em uma rede de protótipos. Para validar seu funcionamento em laboratório, executou-se a implementação de filtros na camada de enlace, onde somente eram entendidas as mensagens de dois vizinhos.

Desta forma, as mensagens eram transmitidas entre os sensores até que o último laço da rede fosse atingido e retornasse com os dados solicitados pela mesma rota. Com isto, validou-se em campo, ainda que de forma primitiva, um esboço do funcionamento teórico do algoritmo previsto e estudado na Seção 3.2.

Ressalta-se, entretanto, que de forma a garantir o funcionamento correto, todo o processo de simulação deve ser ainda estudado e simulado, garantindo temporização adequada a aplicações de tempo real, necessárias a algumas áreas de automação e controle em um ambiente *smart grid*.

5.5. Medições reais

Como forma de validar os dados medidos pelo equipamento desenvolvido, foi feita uma comparação de medição com um equipamento comercial, disponível no mercado, da empresa *Landis+Gyr*, chamado SAGA2000. Todo o processo de medição para comparação durou aproximadamente dois dias e as curvas de dados foram sobrepostas, conforme ilustrado na Figura 64.

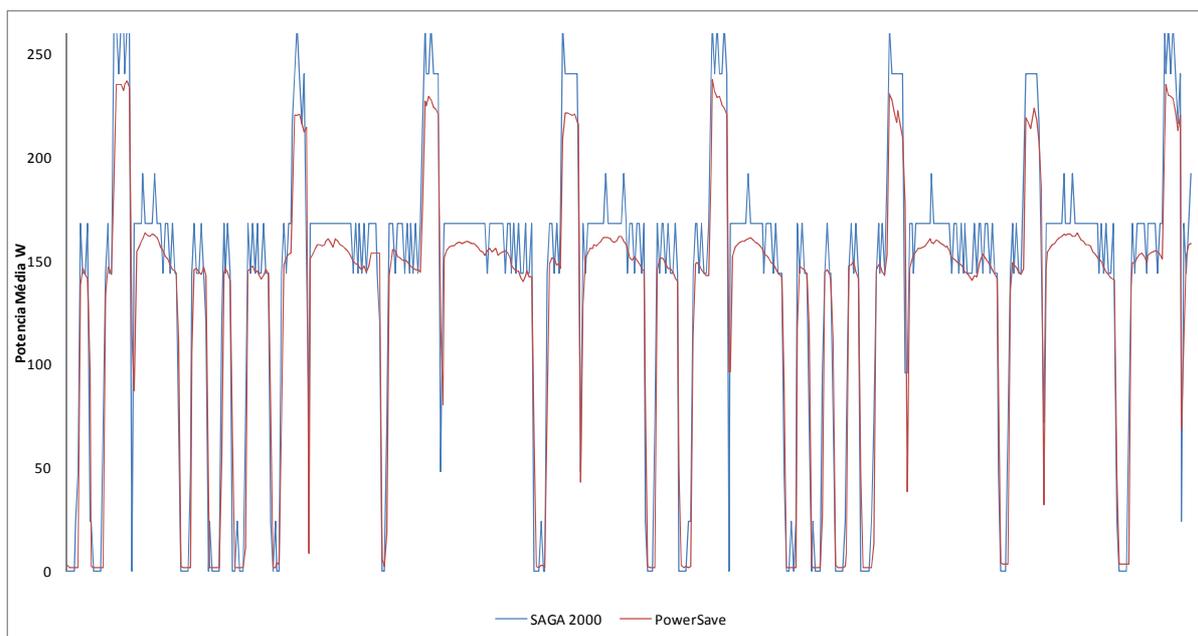


Figura 64 – Comparação das medições do equipamento desenvolvido com um medidor SAGA2000 da *Landis+Gyr* para uma geladeira.

Ao analisar os dados e compará-los, percebe-se que o equipamento desenvolvido, objeto deste trabalho, possui maior sensibilidade de medição em relação às nuances de consumo de um equipamento de pequeno porte, neste caso uma geladeira. Porém, percebe-se também que, no mesmo período, há um desvio no acumulador de consumo, o que de fato, contribui para aumentar a diferença ao longo do tempo, como se observa na Figura 65.

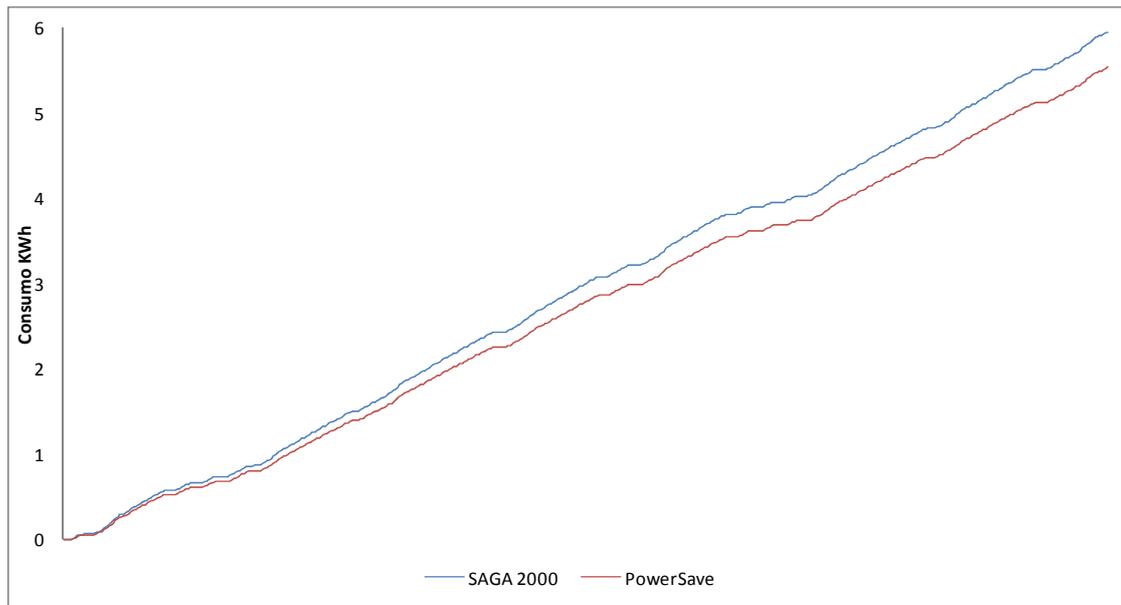


Figura 65 – Curva de consumo de energia para calibração do medidor *Powersave*.

A comparação destes dados é utilizada para recalibrar o medidor embarcado no dispositivo, e, com isto, reduzir a diferença entre um equipamento certificado, garantindo sua acurácia e precisão nas grandezas medidas.

Após o domínio do processo de calibração e sua consequente aplicação do equipamento, foram executadas diversas medições em equipamentos de uso comum em escritórios e residências, tais como televisores, ventiladores, bebedouros e aparelhos de ar condicionado, durante o período de uma semana, para traçar um perfil de consumo. As curvas obtidas com estes testes estão ilustradas na Figura 66.

Observa-se que, desta forma, a arquitetura proposta neste trabalho para transmissão de dados, em um ambiente real, conseguiu obter de forma precisa os dados, sem o comprometimento da operação. Assim, espera-se que sua adoção em massa seja viável, ainda que a tecnologia presente, neste estágio inicial, baixo grau de maturidade, visando um cenário de utilização em redes elétricas inteligentes.

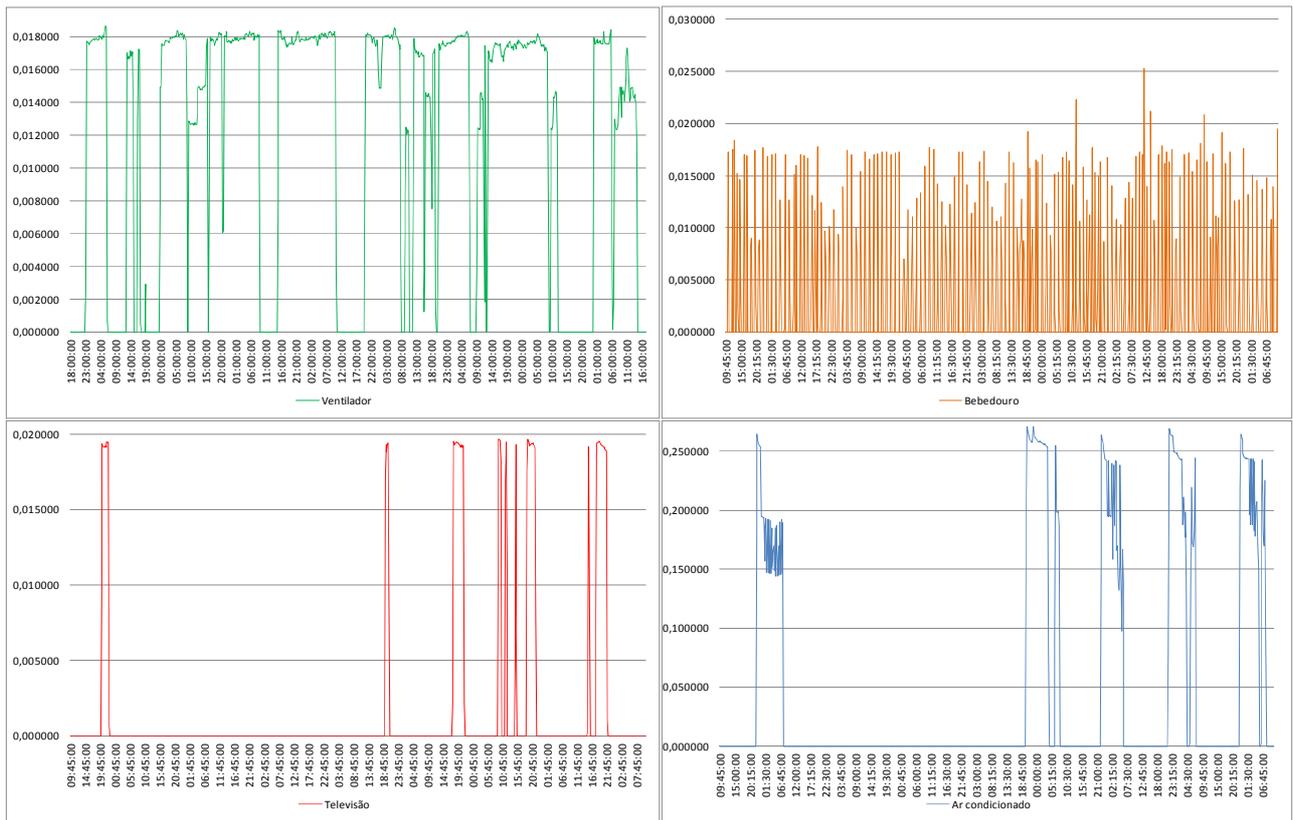


Figura 66 – Dados de medições reais coletadas pelo equipamento (em kW).

6. Conclusões

6.1. Epílogo

O presente trabalho propôs-se a analisar, inicialmente, as referências bibliográficas do estado da arte existentes sobre o tema. Foram avaliadas arquiteturas e tecnologias de comunicação que podem vir a se tornar, no futuro, padrão da indústria tais como o IEEE802.15.4, *ZigBee*, 6LoWPAN, IEEE802.11 (*WiFi*), PLC, entre outros.

Dentre as diversas publicações analisadas, grande parte dos autores propôs uma arquitetura decompondo o problema da comunicação bidirecional, voltada para o *smart grid*, em três grandes grupos, cada um com suas particularidades e características.

No universo compreendido dentro das residências e estabelecimentos, propôs-se a criação de uma rede denominada HAN (*Home Area Network*), composta por todos os dispositivos eletroeletrônicos, tomadas, disjuntores e o medidor central. Para estas redes observou-se que tecnologias de baixo consumo de energia e curto alcance devem ser utilizadas, com autores propondo o uso de *ZigBee*, ou *WiFi*, e até mesmo PLC *narrowband* como soluções viáveis para esta aplicação.

Em uma análise voltada para a comunicação destas redes com as concessionárias de energia, propôs-se então, a criação de dois outros agentes de comunicação chamados BAN (*Building Area Network*) e NAN (*Network Area Network*). Desta forma, os dados de diversas redes NAN podem ser agrupados em *clusters* que organizam a informação e distribuídos por tecnologias PLC *broadband*, 3G, 4G LTE, ou até mesmo fibra ótica, até o centro de controle da concessionária de energia.

A proposta deste trabalho conduziu-se então para uma arquitetura de comunicação para redes do tipo HAN, propondo uma alternativa às soluções fechadas de mercado existentes. Desta forma, esperava-se ser possível baixar o custo total da solução sem comprometer a qualidade e confiabilidade necessárias a uma arquitetura HAN em *smart grid*.

Desta forma, apresentou-se uma série de plataformas de *hardware* que poderiam servir ao trabalho como base para a implementação de um protocolo a ser proposto.

Após uma análise detalhada dos requisitos impostos às tarefas a serem executadas, optou-se pela de menor custo e recursos suficientes ao atendimento dos serviços, com microcontroladores de baixo consumo MSP430 e *transceivers* CC2500, ambos fabricados pela *Texas Instruments*.

Após a definição da plataforma de *hardware*, o trabalho voltou-se para o desenvolvimento da camada de enlace e interface lógica dos dados, além dos possíveis serviços a serem providos pela camada de rede para as aplicações. Percebeu-se, neste estágio, que a definição de serviços especializados ao *smart grid* seriam de grande valia à plataforma, otimizando o uso dos recursos computacionais disponíveis e garantindo tarefas implementadas em *hardware*, melhorando o tempo de resposta para a execução das atividades.

Diversos recursos foram adicionados ao desenho inicial da arquitetura e o sistema passou a oferecer funcionalidades de uma rede em malha sem fio, com a utilização conjunta de abordagens para descobertas de rotas em modos proativos e reativos, atrelando um *hardware* de limitada capacidade computacional com a utilização eficiente do uso de memória.

Após a validação do diagrama de estados, partiu-se para o desafio de simular a plataforma inovadora de rede. Para isto, desenvolveu-se com sucesso um *software* capaz de embarcar toda a tecnologia necessária ao projeto de redes e simular, na forma de eventos discretos, toda a complexidade encontrada em uma rede *smart grid* com diversos sensores. Desta forma, aprimoraram-se de forma recursiva, antes da montagem em bancada do produto, diversos parâmetros da rede relativos à temporização, escalonamento de tarefas, disputa do meio e controle de tráfego.

Todo o processo de modelagem do simulador foi primordial para depurar a concepção das estruturas de dados envolvidas no processo de troca de mensagens entre os nós da rede. Desta forma, foi possível observar seu comportamento passo a passo, de forma a entender o relacionamento entre todos os nós envolvidos no processo e como minimizar os problemas relacionados a colisões em um ambiente de alta complexidade.

Tal processo de depuração seria impossível de se reproduzir em um ambiente de bancada, pois as interações internas e a troca de dados ocorrem em processos complexos, dificultando sua observação, e, mesmo com o uso de equipamentos modernos, seria complexo distinguir a troca de sinais em uma rede com mais de dois

nós (com os equipamentos disponíveis em bancada dentro do laboratório) em tempo de execução, com altas taxas e maior volume de dados.

De posse dos dados e resultados obtidos com a simulação, voltaram-se, então, os trabalhos para a implementação da arquitetura em *hardware* de forma a validar e confrontar os resultados obtidos em simulação com os obtidos em bancada. Assim, torna-se possível a comparação e validação do funcionamento do equipamento, além de obter-se, de fato, uma plataforma física viável para a utilização em *smart grid*.

Utilizando-se uma metodologia padrão para os dois cenários foi possível obter e comparar dados de vazão e índice de justiça (*Fairness Index*) entre diferentes cenários envolvendo até 16 nós em uma rede. As características observadas e medidas dentro do conteúdo deste trabalho cumprem satisfatoriamente os objetivos previamente estipulados.

Desta forma, considerando-se que há uma grande base de estudos formada com este trabalho, é possível afirmar que há argumentos suficientes para a continuação do mesmo, em uma contribuição acadêmica aprimorada, de forma a buscar formas inovadoras de estudos em sistemas *smart grid*, visando seu aprimoramento científico e formação de base tecnológica que possam melhorar e aprimorar a qualidade dos serviços de energia elétrica providos no Brasil e em todo o mundo.

6.2. Trabalhos futuros

No atual estado em que se encontra este trabalho foram citados alguns aprimoramentos que poderiam ser incluídos em versões futuras do projeto. De fato, um dos pontos principais que devem ser incluídos em breve no sistema de simulação é a possibilidade de se utilizar múltiplos saltos. Desta forma, cenários mais elaborados, com múltiplos percursos, poderiam ser incluídos, como forma de validar o algoritmo de descoberta de rotas proposto.

Ainda no que diz respeito ao algoritmo, deve ser incluído dentro do protocolo, em um futuro próximo, a criptografia das mensagens. A introdução desta funcionalidade deverá incluir *overhead* na quantidade de dados trafegados pelos nós e, pode, de fato, impactar no resultado final e na taxa de transferência de dados. Porém, estudos relacionados a uma criptografia eficiente ainda devem ser executados, visando garantir que a probabilidade de decodificação dos dados seja próxima de zero, provendo maior segurança aos dados sensíveis dos usuários que, porventura, venha a ser transmitidos baseados na plataforma proposta por este trabalho.

Procurando também o aprimoramento no algoritmo de inserção e remoção de rotas na memória dos dispositivos, pode-se alterar, futuramente, o processo de escolha dos endereços que serão removidos (no caso de uma tabela cheia). Desta forma, campos de dados trafegados que hoje são utilizados para depuração do código, poderiam servir de base para inserir novos critérios, tais como: nível de sinal, frequência de troca de mensagens, número de transferências com sucesso etc.. Desta forma, a escolha para remoção poderia se basear em uma rota com pouca utilização, ou com menor nível de sinal, por exemplo.

Buscando uma maior redução de custos, conforme comentado na Seção 3.2.1, deverá ser aprimorada a plataforma de *hardware* para um SoC (*System on Chip*), onde em um único encapsulamento estão microcontrolador e *transceiver* de rádio, compatíveis com a atual plataforma. Desta forma, o custo total da solução poderá sofrer redução de até 54%, de acordo com a pesquisa de preços executada, e ainda se comunicar com toda a infraestrutura existente e desenvolvida na atual plataforma, objeto deste trabalho.

Com todos estes novos recursos embarcados, torna-se possível também simular uma rede com um maior número de sensores. Assim, é possível criar sub-redes distintas, e avaliar a interferência causada quando diversos dispositivos *sink* buscam informações dos nós atrelados a eles. Cria-se, então, um modelo de distribuição dos equipamentos bastante similar ao que será encontrado quando, após sua massificação, a plataforma for instalada em diversas residências e/ou escritórios em prédios, em um cenário urbano, validando-se, por simulação, o que de fato pretende-se que aconteça em um cenário futuro.

Adiciona-se a este fato, a possibilidade de um modelamento matemático mais sofisticado, no que diz respeito ao desvanecimento de sinal. O atual sistema propõe que uma função linear seja utilizada como padrão. A inserção de modelos do tipo FSPL (*Free Space Path Loss*), com obstáculos para redução do nível de sinal e a adição de reflexões em multipercurso *indoor* para aplicação de interferência intersimbólica, são adendos que podem ser incluídos em versões futuras da plataforma de simulação.

Embora se trate de uma proposta para “trabalhos futuros” bastante ambiciosa, a construção da plataforma deste simulador tornou-se tão modular que é possível idealizar, em um futuro não tão distante, a simulação de uma rede elétrica em *smart grid* completa, desde as redes residenciais (HAN), passando por novos agentes *gateway* (BAN e NAN), com diversos aspectos possíveis, em meios físicos distintos e modulações sendo simuladas, até que os dados sejam armazenados em uma central de controle, dentro da concessionária de energia, simulada por um banco de dados.

Desta forma, torna-se possível vislumbrar um cenário onde há um *ranking* de performance, baseado em parâmetros de requisitos dos projetos, que auxilie de forma técnica, estratégica e econômica, quais plataformas são mais adequadas para cada tipo de cenário, aplicação e topologia. Define-se então um novo desafio tecnológico para a implantação de redes inteligentes no Brasil, visando o aperfeiçoamento da metodologia para sua implantação.

Referências Bibliográficas

ABREME. 2006. Pesquisa de Mercado 2006. *Associação Brasileira dos Revendedores e Distribuidores de Material Elétrico*. [Online] 2006. [Citado em: 25 de Fevereiro de 2012.] http://www.abreme.com.br/pdf/Pesquisa_mercado_Abreme_2006.pdf.

AGÊNCIA BRASIL. 2012. Brasil estuda redes inteligentes para diminuir desperdício de energia elétrica. *Agência Brasil EBC*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://agenciabrasil.ebc.com.br/noticia/2012-02-01/brasil-estuda-redes-inteligentes-para-diminuir-desperdicio-de-energia-eletrica>.

ARDUINO. 2012. Home Website. *Arduino*. [Online] 2012. [Citado em: 25 de Fevereiro de 2012.] <http://www.arduino.cc>.

ATMEL. 2012. Wireless for Home Appliances. *ATMEL*. [Online] 2012. [Citado em: 25 de Fevereiro de 2012.] <http://www.atmel.com/>.

BANCO MUNDIAL. 2012. Energy & Mining. *WorldBank*. [Online] 2012. [Citado em: 09 de Fevereiro de 2012.] <http://data.worldbank.org/topic/energy-and-mining>.

CAO, LITING, JIANG, WEI e ZHANG, ZHAOLI. 2008. Networked wireless meter reading system based on ZigBee technology. *Control and Decision Conference*. Agosto de 2008, pp. 3455-3460.

CAO, ZHONGYU e LU, GANG. 2010. S-AODV: Sink Routing Table over AODV Routing Protocol for 6LoWPAN. *Networks Security Wireless Communications and Trusted Computing*. 2010, pp. 340-343.

CC1101. 2009. Low-Power Sub-1GHz RF Transceiver. *Texas Instruments*. [Online] 2009. [Citado em: 14 de Fevereiro de 2012.] <http://www.ti.com/product/cc1101>.

CC2420. 2008. Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee™ Ready RF Transceiver. *Texas Instruments*. [Online] 2008. [Citado em: 25 de Fevereiro de 2012.] <http://www.ti.com/product/cc2420>.

CC2500. 2007. Low Cost, Low-Power 2.4 GHz RF Transceiver. *Texas Instruments*. [Online] 2007. [Citado em: 14 de Fevereiro de 2012.] <http://www.ti.com/product/cc2500>.

CHOI, HAKSOO, KIM, NAKYOUNG e CHA, HOJUNG. 2009. 6LoWPAN-SNMP: Simple Network Management Protocol for 6LoWPAN. *High Performance Computing and Communications*. Julho de 2009, pp. 305-313.

DA SILVA CAMPOS, B., et al. 2011. Design and construction of a wireless sensor and actuator network gateway based on 6LoWPAN. *EUROCON - International Conference on Computer as a Tool*. Junho de 2011, pp. 1-4.

DIGI. 2012. ZigBee and RF Modules. *DIGI*. [Online] 2012. [Citado em: 25 de Fevereiro de 2012.] <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/>.

DIGIKEY. 2012. Home Website Price Search. *Digikey Corporation*. [Online] 2012. [Citado em: 25 de Fevereiro de 2012.] <http://www.digikey.com>.

EPE BEN. 2011. Balanço Energético Nacional. *EPE*. [Online] 2011. [Citado em: 09 de Fevereiro de 2012.]

ESS MESH. 2006. Draft Amendment to Standard for Information Technology - Telecommunications and Information Exchange Between Systems. [ed.] IEEE 802.11s Task Group. *LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Novembro de 2006. Amendment: ESS Mesh Networking.

EZ430-RF2500. 2009. MSP430 Wireless Development Tool. *eZ430-RF2500 Development Tool User's Guide*. [Online] 2009. [Citado em: 22 de Outubro de 2012.] <http://www.ti.com/tool/ez430-rf2500>.

FADLULLAH, Z.M., et al. 2011. Toward intelligent machine-to-machine communications in smart grid. *Communications Magazine, IEEE*. Abril de 2011, pp. 60-65.

FEDERAL STANDARD 1037C. 1996. Carrier sense multiple access with collision avoidance (CSMA/CA). *Institute for Telecommunication Sciences*. [Online] National Telecommunications and Information Administration, 1996. [Citado em: 13 de Fevereiro de 2012.] http://www.its.bldrdoc.gov/fs-1037/dir-006/_0859.htm.

FOUDA, M.M., FADLULLAH, Z.M. e KATO, N. 2010. Assessing attack threat against ZigBee-based home area network for Smart Grid communications. *Computer Engineering and Systems (ICCES), 2010 International Conference on*. Dezembro de 2010, pp. 245-250.

GHARAVI, H. e HU, BIN. 2011. Multigate Communication Network for Smart Grid. *Proceedings of the IEEE*. Junho de 2011, pp. 1028-1045.

HAN, DAE-MAN e LIM, JAE-HYUN. 2010. Design and implementation of smart home energy management systems based on zigbee. *Consumer Electronics, IEEE Transactions on*. Outubro de 2010, pp. 1417-1425.

HASAN, M., RAHMAN, S. e DINCER, H. 2011. Design of wireless smart metering system based on MSP430 MCU and ZigBee for residential application. *Electrical and Electronics Engineering (ELECO), 7th International Conference on*. Dezembro de 2011, pp. 255-258.

HASBOLLAH, A.A., ARIFFIN, S.H.S. e HAMINI, M.I.A. 2010. Performance analysis for 6loWPAN IEEE 802.15.4 with IPv6 network. *TENCON IEEE Region 10 Conference*. Janeiro de 2010, pp. 1-5.

HEINZELMAN, W.B., CHANDRAKASAN, A.P. e BALAKRISHNAN, H. 2002. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*. Dezembro de 2002, pp. 660-670.

IBM. 2009. Estudo da IBM revela aumento da preocupação dos consumidores com o consumo de energia. *IBM Press*. [Online] 2009. [Citado em: 10 de Fevereiro de 2012.] <http://www-03.ibm.com/press/br/pt/pressrelease/27165.wss>.

IEEE 802.11. 2012. IEEE 802.11™ The Working Group for WLAN Standards. *IEEE 802.11™*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://www.ieee802.org/11/>.

- IEEE 802.15.4. 2012.** IEEE 802.15 WPAN™ Task Group 4 (TG4). *IEEE 802.15*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://www.ieee802.org/15/pub/TG4.html>.
- IEEE SG. 2012.** Smart Grid Conceptual Model. *IEEE*. [Online] 2012. [Citado em: 10 de Fevereiro de 2012.] <http://smartgrid.ieee.org/ieee-smart-grid/smart->
- IEEE Std 802.15.4. 2003.** Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal. *Johns Hopkins University*. [Online] 2003. [Citado em: 13 de Fevereiro de 2012.] <http://www.cs.jhu.edu/~cliang4/public/datasheets/802.15.4-2003.pdf>.
- IETF 6lowpan. 2012.** IPv6 over Low power WPAN (6lowpan). *6LoWPan*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://datatracker.ietf.org/wg/6lowpan/>.
- IETF MANET. 2012.** Mobile Ad-hoc Networks. *IETF*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://datatracker.ietf.org/wg/manet/>.
- IPV6 CISCO. 2008.** IPv6 - White Paper. *CISCO*. [Online] 2008. [Citado em: 25 de 09 de 2012.] http://www.cisco.com/web/strategy/docs/gov/IPv6_WP.pdf.
- JAIN, R. 2012.** Curriculum Vitae. [Online] 2012. [Citado em: 11 de Outubro de 2012.] http://www.cse.wustl.edu/~jain/cv_jain.htm.
- JAMIAN, J.J., et al. 2011.** Conceptual data management and communication for smart distribution system. *IEEE First Conference on Clean Energy and Technology (CET)*. Outubro de 2011, pp. 364-368.
- JIAN, HUANG, LI, LI e XIAOGUANG, HU. 2011.** Design of new architecture of AMR system in Smart Grid. *Industrial Electronics and Applications (ICIEA), 6th IEEE Conference on*. Agosto de 2011, pp. 2025-2029.
- JONES, V. 2003.** Radio Propagation Models. *Wireless Communication*. [Online] 2003. [Citado em: 09 de Outubro de 2012.] http://people.seas.harvard.edu/~jones/es151/prop_models/propagation.html#fsl.
- KODIALAM, M. e NANDAGOPAL, T. 2005.** Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks. *In Proc. ACM MobiCom*. 2005.
- KYASANUR, P. e VAIDYA, N. H. 2004.** Routing in Multi-Channel Multi-Interface Ad Hoc Wireless Networks. Dezembro de 2004.
- LANGHAMMER, N. e KAYS, R. 2012.** Evaluation of wireless smart metering technologies in realistic indoor channels. *Wireless Communication Systems (ISWCS), 8th International Symposium on*. Janeiro de 2012, pp. 387-391.
- LATR'É, BENOIT, et al. 2005.** Maximum Throughput and Minimum Delay in IEEE 802.15.4. *1st International conference on Mobile Ad-hoc and Sensor Networks*. Dezembro de 2005, pp. 866-876.
- LEE, TZUNG-JE, et al. 2012.** 2.45 GHz ZigBee receiver frontend for HAN with smart meter. *Integrated Circuits (ISIC), 13th International Symposium on*. Janeiro de 2012, pp. 480-483.
- LI, LI, XIAOGUANG, HU e KE, CHEN. 2011.** The applications of WiFi-based Wireless Sensor Network in Internet of Things and Smart Grid. *Industrial Electronics and Applications (ICIEA), 6th IEEE Conference on*. Agosto de 2011, pp. 789-793.

- LISBOA, C. 2003.** Introdução às Redes de Petri. *Prof. Dr. Carlos Renato Lisboa UFPA*. [Online] Agosto de 2003. [Citado em: 5 de Março de 2012.] http://www.dca.ufrn.br/~affonso/DCA0409/pdf/redes_de_petri.pdf.
- LIU, JIANMING, et al. 2010.** Application of power line communication in smart power consumption. *Power Line Communications and Its Applications (ISPLC), IEEE International Symposium on*. Junho de 2010, pp. 303-307.
- LUAN, SHANG-WEN, et al. 2010.** Development of a smart power meter for AMI based on ZigBee communication. *Power Electronics and Drive Systems*. Janeiro de 2010, pp. 661-665.
- MAZIERO, C. 1990.** THE ARP TOOL. *Prof. Carlos A. Maziero*. [Online] 1990. [Citado em: 5 de Março de 2012.] http://dainf.ct.utfpr.edu.br/~maziero/doku.php/software:arp_tool.
- MIT BLUWARE. 2002.** Blueware: Bluetooth Simulator for ns. *Blueware: Bluetooth Simulator for ns*. [Online] MIT, 2002. [Citado em: 08 de Outubro de 2012.] <http://nms.lcs.mit.edu/projects/blueware/software/>.
- MSP430FE427A. 2008.** 16-Bit Ultra-Low-Power Microcontroller for Energy Meters. *Texas Instruments*. [Online] 2008. [Citado em: 25 de Fevereiro de 2012.] <http://www.ti.com/product/msp430fe427a>.
- MUKHTAR, H., et al. 2008.** LNMP- Management architecture for IPv6 based low-power wireless Personal Area Networks (6LoWPAN). *Network Operations and Management Symposium*. Agosto de 2008, pp. 417-424.
- OPERA. 2004.** OPERA Open PLC European Research Alliance. *OPERA*. [Online] 2004. [Citado em: 08 de Fevereiro de 2012.] <http://www.ist-bread.org/pdf/SBS%202004%20Workshop/OPERA.pdf>.
- QINGYANG, L., et al. 2009.** Experience of AMR systems based on BPL in China. *Power Line Communications and Its Applications. ISPLC. IEEE International Symposium on*. Maio de 2009, pp. 280-284.
- RAJAGOPAL, V., SINGH, B. e KASAL, G.K. 2011.** Electronic load controller with power quality improvement of isolated induction generator for small hydro power generation. *Renewable Power Generation, IET*. Março de 2011, Vol. 5, 2, pp. 201-213.
- REVISTA FATOR. 2010.** Pesquisa comprova desperdício de energia no brasil de R\$ 15 bilhões ao ano. *Portal Fator Brasil*. [Online] 2010. [Citado em: 08 de Fevereiro de 2012.] http://www.revistafator.com.br/ver_noticia.php?not=121208.
- RFC3561. 2003.** IETF Network Working Group. *Ad hoc On-Demand Distance Vector (AODV) Routing*. [Online] Julho de 2003. [Citado em: 27 de Fevereiro de 2012.] <http://tools.ietf.org/pdf/rfc3561.pdf>.
- RICCI, A., et al. 2011.** Wr@p: A “last-meter” technology for energy-aware networked appliances. *Power Line Communications and Its Applications (ISPLC), IEEE International Symposium on*. Maio de 2011, pp. 193-198.
- ROCHOL, J., et al. 2003.** Plataformas de Simulação de Software Livre para Redes Fixas e Móveis: Características, Suporte, Instalação e Validação. 2003.

SAAKSJARVI, M. 2002. Software application platforms: from product architecture to integrated application strategy. *Computer Software and Applications Conference*. Dezembro de 2002, pp. 435-443.

SALMAN, N., RASOOL, I. e KEMP, A.H. 2010. Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks. *Wireless Communication Systems (ISWCS)*. Novembro de 2010, 7th International Symposium, pp. 701-705.

SAUTER, T. e LOBASHOV, M. 2011. End-to-End Communication Architecture for Smart Grids. *Industrial Electronics, IEEE Transactions on*. Março de 2011, pp. 1218-1228.

SCHAUER, S. e VENKAT, K. 2009. Implementing An Electronic Watt-Hour Meter With MSP430FE42x(A)/FE42x2. *Literature for implementing Electricity Metering*. [Online] 2009. [Citado em: 22 de Outubro de 2012.] <http://www.ti.com/lit/an/slaa203c/slaa203c.pdf>.

SEUNGHYUN, P., et al. 2010. Concurrent simulation platform for energy-aware smart metering systems. *Consumer Electronics, IEEE Transactions on*. Outubro de 2010, pp. 1918-1926.

SMART HOME USA. 2012. How X10 Works. *X10*. [Online] 2012. [Citado em: 12 de 02 de 2012.] <http://www.smarthomeusa.com/info/x10theory/x10theory/#theory>.

SROECO SOLAR. 2012. Most Efficient Solar Panels. *SRoeCo Solar*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://sroeco.com/solar/most-efficient-solar-panels>.

STARSINIC, M. 2010. System architecture challenges in the home M2M network. *Long Island Systems Applications and Technology Conference (LISAT)*. Junho de 2010, pp. 1-7.

TANAKA, M., et al. 2011. New throughput analysis of long-distance IEEE 802.11 wireless communication system for smart grid. *Smart Grid Communications (SmartGridComm), IEEE International Conference on*. Dezembro de 2011, pp. 90-95.

TANENBAUM, S.A. 2003. *Redes de Computadores*. s.l. : Prentice Hall, 2003. Vol. 4o. ISBN 970-26-0162-2.

TEXAS INSTRUMENTS. 2012. SimpliciTI Compliant Protocol Stack. *SimpliciTI*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://www.ti.com/tool/simpliciti>.

THE ECONOMIST. 2008. The elusive negawatt. *Energy efficiency*. [Online] 2008. [Citado em: 08 de Fevereiro de 2012.] <http://www.economist.com/node/11326549>.

TINGRUI, PEI, et al. 2009. An Improved Hierarchical AODV Routing Protocol for Hybrid Wireless Mesh Network. *Networks Security, Wireless Communications and Trusted Computing*. Maio de 2009, pp. 588-593.

TUDOSE, D.S., et al. 2011. Home automation design using 6LoWPAN wireless sensor networks. *Distributed Computing in Sensor Systems and Workshops (DCOSS)*. Agosto de 2011, pp. 1-6.

VERMA, V., et al. 2010. Power Conditioner for Variable-Frequency Drives in Offshore Oil Fields. *Industry Applications, IEEE Transactions on*. Abril de 2010, pp. 731-739.

- WANG, F., LI, D. e ZHAO, Y. 2011.** Analysis of CSMA/CA in IEEE 802.15.4. *Communications, IET*. Novembro de 2011, pp. 2187-2195.
- WANG, QIHE e AGRAWAL, DHARMA. 2003.** UCBT - Bluetooth extension for NS2 at the University of Cincinnati. *UCBT - University of Cincinnati - BlueTooth*. [Online] 2003. [Citado em: 08 de Outubro de 2012.] <http://www.cs.uc.edu/~cdmc/ucbt/>.
- YIBO, CHEN, et al. 2011.** 6LoWPAN Stacks: A Survey. *Wireless Communications, Networking and Mobile Computing (WiCOM)*. Outubro de 2011, pp. 1-4.
- ZHANG, YICHI, et al. 2011.** A multi-level communication architecture of smart grid based on congestion aware wireless mesh network. *North American Power Symposium (NAPS)*. Setembro de 2011, pp. 1-6.
- ZIGBEE ALLIANCE. 2012.** Zigbee Specifications. *Zigbee*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://www.zigbee.org/Specifications.aspx>.
- Z-WAVE. 2012.** Energy Conservation. *Z-Wave*. [Online] 2012. [Citado em: 08 de Fevereiro de 2012.] <http://www.z-wave.com/modules/Z-WaveSolutions/?id=24&chk=d1801e23b9ec09c91dfaf4a0ec1ba603>.

Anexo A

Resultado da Rede de Petri

*-----
-----*

Net S_R_1 ;

Nodes

```
CLIENT : place (1) ;
MSG : place ;
WAIT_ACK : place ;
SERV : place (1) ;
PROC : place ;
ACK : place ;
WAITDATA : place ;
DATA : place ;
WAITACK : place ;
ACK_2 : place ;
CHOOSE : place ;
CHOOSE2 : place ;
choose_to_serv : place ;
choose_to_client : place ;
CSMA_CA : place (1) ;
MSG2 : place ;
ACK2 : place ;
DATA2 : place ;
ACK_22 : place ;
MSG_Lost : place ;
MSG_Lost2 : place ;
WAIT_STATE : place ;
Ack_arrived : place ;

SEND : transition ;
RECV : transition ;
RECV1 : transition ;
SENDDATA : transition ;
RECVDT : transition ;
ACKOK : transition ;
CLIENTCH : transition ;
SERVCH : transition ;
SERVCH2 : transition ;
CLIENTCH2 : transition ;
transito1 : transition ;
transito2 : transition ;
transito3 : transition ;
transito4 : transition ;
```

```

perda1 : transition ;
Re_TX1 : transition ;
perda2 : transition ;
perda3 : transition ;
Re_TX2 : transition ;
perda4 : transition ;
noloss : transition ;

```

Structure

```

SEND : (CLIENT, CSMA_CA) , (MSG, WAIT_ACK) ;
RECV : (SERV, CSMA_CA, MSG2) , (PROC, ACK) ;
RECV1 : (WAIT_ACK, ACK2) , (WAITDATA) ;
SENDDATA : (PROC, CSMA_CA) , (DATA, WAITACK) ;
RECVDT : (WAITDATA, CSMA_CA, DATA2) , (ACK_2,
WAIT_STATE) ;
ACKOK : (WAITACK, ACK_22) , (CHOOSE2, choose_to_serv) ;
CLIENTCH : (CHOOSE, choose_to_client) , (CLIENT) ;
SERVCH : (CHOOSE, choose_to_serv) , (SERV) ;
SERVCH2 : (CHOOSE2, choose_to_serv) , (SERV) ;
CLIENTCH2 : (CHOOSE2, choose_to_client) , (CLIENT) ;
transito1 : (MSG) , (MSG2, CSMA_CA) ;
transito2 : (ACK) , (CSMA_CA, ACK2) ;
transito3 : (DATA) , (DATA2, CSMA_CA) ;
transito4 : (ACK_2) , (ACK_22, CSMA_CA, Ack_arrived) ;
perda1 : (MSG, WAIT_ACK) , (MSG_Lost) ;
Re_TX1 : (MSG_Lost) , (MSG, WAIT_ACK) ;
perda2 : (ACK, PROC, WAIT_ACK) , (MSG_Lost, SERV) ;
perda3 : (DATA, WAITACK) , (MSG_Lost2) ;
Re_TX2 : (MSG_Lost2) , (DATA, WAITACK) ;
perda4 : (ACK_2, WAITACK, WAIT_STATE) , (MSG_Lost2,
WAITDATA) ;
noloss : (WAIT_STATE, Ack_arrived) , (CHOOSE,
choose_to_client) ;

```

endNet.

```

*-----*
-----*

```

States Enumeration : net S_R_1.

Reachability graph for this net:

```

*-----*
-----*

```

```

M0   :(SEND: M1)
M1   :(transito1: M2) (perda1: M18)
M2   :(RECV: M3)
M3   :(transito2: M4) (perda2: M18)
M4   :(RECV1: M5) (SENDDATA: M30)
M5   :(SENDDATA: M6)
M6   :(transito3: M7) (perda3: M29)

```

```

M7      :(RECVDT: M8)
M8      :(transito4: M9) (perda4: M29)
M9      :(ACKOK: M10) (noloss: M24)
M10     :(SERVCH2: M11) (noloss: M13)
M11     :(noloss: M12)
M12     :(CLIENTCH: M0)
M13     :(CLIENTCH: M14) (SERVCH: M19) (SERVCH2: M12)
        (CLIENTCH2: M20)
M14     :(SEND: M15) (SERVCH2: M0)
M15     :(SERVCH2: M1) (transito1: M16) (perda1: M17)
M16     :(SERVCH2: M2)
M17     :(SERVCH2: M18) (Re_TX1: M15)
M18     :(Re_TX1: M1)
M19     :(CLIENTCH2: M0)
M20     :(SEND: M21) (SERVCH: M0)
M21     :(SERVCH: M1) (transito1: M22) (perda1: M23)
M22     :(SERVCH: M2)
M23     :(SERVCH: M18) (Re_TX1: M21)
M24     :(ACKOK: M13) (CLIENTCH: M25)
M25     :(SEND: M26) (ACKOK: M14)
M26     :(ACKOK: M15) (transito1: M27) (perda1: M28)
M27     :(ACKOK: M16)
M28     :(ACKOK: M17) (Re_TX1: M26)
M29     :(Re_TX2: M6)
M30     :(RECV1: M6) (transito3: M31) (perda3: M32)
M31     :(RECV1: M7)
M32     :(RECV1: M29) (Re_TX2: M30)

```

```

*-----*
-----*

```

State Enumeration : net S_R_1.
Reachable states for this net :

```

*-----*
-----*

```

```

M0      :{CLIENT, SERV, CSMA_CA}
M1      :{MSG, WAIT_ACK, SERV}
M2      :{WAIT_ACK, SERV, CSMA_CA, MSG2}
M3      :{WAIT_ACK, PROC, ACK}
M4      :{WAIT_ACK, PROC, CSMA_CA, ACK2}
M5      :{PROC, WAITDATA, CSMA_CA}
M6      :{WAITDATA, DATA, WAITACK}
M7      :{WAITDATA, WAITACK, CSMA_CA, DATA2}
M8      :{WAITACK, ACK_2, WAIT_STATE}
M9      :{WAITACK, CSMA_CA, ACK_22, WAIT_STATE, Ack_arrived}
M10     :{CHOOSE2, choose_to_serv, CSMA_CA, WAIT_STATE,
Ack_arrived}
M11     :{SERV, CSMA_CA, WAIT_STATE, Ack_arrived}
M12     :{SERV, CHOOSE, choose_to_client, CSMA_CA}
M13     :{CHOOSE, CHOOSE2, choose_to_serv, choose_to_client,

```

```

CSMA_CA}
M14  :{CLIENT, CHOOSE2, choose_to_serv, CSMA_CA}
M15  :{MSG, WAIT_ACK, CHOOSE2, choose_to_serv}
M16  :{WAIT_ACK, CHOOSE2, choose_to_serv, CSMA_CA, MSG2}
M17  :{CHOOSE2, choose_to_serv, MSG_Lost}
M18  :{SERV, MSG_Lost}
M19  :{SERV, CHOOSE2, choose_to_client, CSMA_CA}
M20  :{CLIENT, CHOOSE, choose_to_serv, CSMA_CA}
M21  :{MSG, WAIT_ACK, CHOOSE, choose_to_serv}
M22  :{WAIT_ACK, CHOOSE, choose_to_serv, CSMA_CA, MSG2}
M23  :{CHOOSE, choose_to_serv, MSG_Lost}
M24  :{WAITACK, CHOOSE, choose_to_client, CSMA_CA, ACK_22}
M25  :{CLIENT, WAITACK, CSMA_CA, ACK_22}
M26  :{MSG, WAIT_ACK, WAITACK, ACK_22}
M27  :{WAIT_ACK, WAITACK, CSMA_CA, MSG2, ACK_22}
M28  :{WAITACK, ACK_22, MSG_Lost}
M29  :{WAITDATA, MSG_Lost2}
M30  :{WAIT_ACK, DATA, WAITACK, ACK2}
M31  :{WAIT_ACK, WAITACK, CSMA_CA, ACK2, DATA2}
M32  :{WAIT_ACK, ACK2, MSG_Lost2}

```

```

*-----*
-----*

```

State Enumeration : net S_R_1 (33 reachable states).
Verified properties:

```

*-----*
-----*

```

Net under analysis is binary.

```

Null places (M = 0): {}
Binary places      : {all}
k-Bounded places  : {}
Unbounded places  : {}

```

Net under analysis is not strictly conservative.

Net under analysis is live.

```

Live Tr.           : {all}
"Almost-live" Tr.: {all}
Non-fired Tr.     : {}

```

Net can always go back to M0.

No live-locks detected.

No deadlocks detected.

```

*-----*
-----*

```

Performance Evaluation Oriented to STATES of Net S_R_1.

```

Initial Marking : {CLIENT, SERV, CSMA_CA}
Desired Precision : 0.10 %

```

Max. of Fires : 210
Num. of Interactions : 2110
Improductive Interac.: 1 (0.05 %)

*-----
-----*

Average number of fires from cycle transitions :

(SEND: 1.47) (RECV: 2.98) (RECV1: 1.47) (SENDDATA:
1.47) (RECVDT: 2.92)
(ACKOK: 1.47) (CLIENTCH: 1.09) (SERVCH: 0.38) (SERVCH2:
1.09)
(CLIENTCH2: 0.38) (transito1: 2.98) (transito2: 1.47)
(transito3: 2.92)
(transito4: 1.47) (perda1: 2.95) (Re_TX1: 4.47) (perda2:
1.52)
(perda3: 2.97) (Re_TX2: 4.41) (perda4: 1.45) (noloss:
1.47)

*-----
-----*

Md 1: {CLIENT, SERV, CSMA_CA}

Average Time: 0.00 Deviation: 0.00 Probab: 99.95
% Reach: 2109

Minimum time of reaching : 0.00 Maximum time of
reaching : 0.00

*-----
-----*

Invariants Analysis for the net S_R_1.

*-----
-----*

Inhibitions used in this analysis:

Obligatory Transitions : {}

Prohibited Transitions : {}

*-----
-----*

Transition invariant basis:

BI1 :{perda1, Re_TX1}
BI2 :{RECV, transito1, -perda1, perda2}
BI3 :{CLIENTCH, -SERVCH, SERVCH2, -CLIENTCH2}
BI4 :{perda3, Re_TX2}
BI5 :{RECVDT, transito3, -perda3, perda4}

BI6 :{SEND, RECV, RECV1, SENDDATA, RECVDT, ACKOK, SERVCH, CLIENTCH2, transitol, transito2, transito3, transito4, noloss}

*-----
-----*

Minimum positive transition invariants for this net:

IT1 :{SEND, RECV, RECV1, SENDDATA, RECVDT, ACKOK, SERVCH, CLIENTCH2, transitol, transito2, transito3, transito4, noloss}

IT2 :{RECV, transitol, Re_TX1, perda2}

IT3 :{RECVDT, transito3, Re_TX2, perda4}

IT4 :{SEND, RECV, RECV1, SENDDATA, RECVDT, ACKOK, CLIENTCH, SERVCH2, transitol, transito2, transito3, transito4, noloss}

IT5 :{perda1, Re_TX1}

IT6 :{perda3, Re_TX2}

*-----
-----*

Transitions in all invariants:

T = {}

Transitions at none invariant:

T = {}

*-----
-----*

Sub-Net existency test (events graph):

Following invariants are events graph:

IT2 IT3 IT5 IT6

*-----
-----*

Invariants Analysis for the net S_R_1.

*-----
-----*

Inhibitions used in this analysis:

Obligatory Places : {}

Prohibited Places : {}

*-----
-----*

Place invariant basis:

BI1 :{CHOOSE, CHOOSE2, -choose_to_serv, -choose_to_client}
BI2 :{MSG, -WAIT_ACK, ACK, MSG2, ACK2}
BI3 :{MSG, -WAIT_ACK, -SERV, -WAITDATA, DATA, -WAITACK, -
choose_to_serv, MSG2, DATA2}
BI4 :{MSG, -WAIT_ACK, -SERV, -WAITDATA, -ACK_2, -
choose_to_serv, MSG2, -ACK_22}
BI5 :{MSG, -SERV, -PROC, ACK, DATA, -WAITACK, ACK_2, -
choose_to_serv, CSMA_CA, MSG_Lost}
BI6 :{SERV, PROC, WAITACK, choose_to_serv, MSG_Lost2}
BI7 :{CLIENT, -MSG, WAIT_ACK, SERV, PROC, -ACK, WAITDATA,
-DATA, WAITACK, -ACK_2, choose_to_serv, choose_to_client, -
CSMA_CA, WAIT_STATE}
BI8 :{CLIENT, -MSG, WAIT_ACK, SERV, PROC, -ACK, WAITDATA,
-DATA, WAITACK, choose_to_serv, choose_to_client, -CSMA_CA,
Ack_arrived}

*-----
-----*

Minimum positive place invariants for this net:

IL1 :{CLIENT, MSG, PROC, DATA, choose_to_client, MSG2,
DATA2, MSG_Lost, MSG_Lost2, WAIT_STATE}
IL2 :{CLIENT, MSG, SERV, PROC, ACK, WAITDATA, WAITACK,
CHOOSE, CHOOSE2, MSG2, ACK2, MSG_Lost, MSG_Lost2,
WAIT_STATE}
IL3 :{CLIENT, WAIT_ACK, SERV, PROC, WAITDATA, DATA, ACK_2,
CHOOSE, CHOOSE2, DATA2, ACK_22, MSG_Lost, MSG_Lost2,
WAIT_STATE}
IL4 :{CLIENT, MSG, SERV, 2* ACK, 2* WAITDATA, ACK_2,
CHOOSE, CHOOSE2, MSG2, 2* ACK2, ACK_22, MSG_Lost,
WAIT_STATE}
IL5 :{SERV, PROC, WAITACK, choose_to_serv, MSG_Lost2}
IL6 :{CLIENT, MSG, SERV, 2* PROC, 2* DATA, ACK_2, CHOOSE,
CHOOSE2, MSG2, 2* DATA2, ACK_22, MSG_Lost, 2* MSG_Lost2,
WAIT_STATE}
IL7 :{CLIENT, WAIT_ACK, SERV, PROC, WAITDATA, DATA, 2*
ACK_2, CHOOSE, CHOOSE2, DATA2, ACK_22, MSG_Lost, MSG_Lost2,
Ack_arrived}
IL8 :{CLIENT, WAIT_ACK, SERV, PROC, WAITDATA, WAITACK,
CHOOSE, CHOOSE2, MSG_Lost, MSG_Lost2, WAIT_STATE}
IL9 :{2* CLIENT, MSG, WAIT_ACK, 2* SERV, 3* PROC,
WAITDATA, 3* DATA, 2* ACK_2, 2* CHOOSE, 2* CHOOSE2, MSG2,

```

3* DATA2, 2* ACK_22,          2* MSG_Lost, 3* MSG_Lost2, 2*
WAIT_STATE}

IL10 :{CLIENT, MSG, ACK, WAITDATA, choose_to_client, MSG2,
ACK2, MSG_Lost, WAIT_STATE}

IL11 :{CLIENT, WAIT_ACK, WAITDATA, choose_to_client,
MSG_Lost, WAIT_STATE}

IL12 :{CLIENT, WAIT_ACK, SERV, ACK, 2* WAITDATA, ACK_2,
CHOOSE, CHOOSE2, ACK2, ACK_22, MSG_Lost, WAIT_STATE}

IL13 :{CLIENT, MSG, SERV, 2* PROC, DATA, WAITACK, CHOOSE,
CHOOSE2, MSG2, DATA2, MSG_Lost, 2* MSG_Lost2, WAIT_STATE}

IL14 :{2* CLIENT, 2* WAIT_ACK, 2* SERV, PROC, ACK, 3*
WAITDATA, DATA, 2* ACK_2, 2* CHOOSE, 2* CHOOSE2, ACK2,
DATA2, 2* ACK_22, 2* MSG_Lost, MSG_Lost2, 2* WAIT_STATE}

IL15 :{CLIENT, MSG, SERV, PROC, ACK, WAITDATA, DATA, ACK_2,
CHOOSE, CHOOSE2, MSG2, ACK2, DATA2, ACK_22, MSG_Lost,
MSG_Lost2, WAIT_STATE}

IL16 :{2* CLIENT, MSG, WAIT_ACK, 2* SERV, 2* PROC, ACK, 2*
WAITDATA, 2* DATA, 2* ACK_2, 2* CHOOSE, 2* CHOOSE2, MSG2,
ACK2, 2* DATA2,          2* ACK_22, 2* MSG_Lost, 2*
MSG_Lost2, 2* WAIT_STATE}

IL17 :{CLIENT, MSG, PROC, DATA, ACK_2, choose_to_client,
MSG2, DATA2, MSG_Lost, MSG_Lost2, Ack_arrived}

IL18 :{CLIENT, MSG, SERV, PROC, ACK, WAITDATA, WAITACK,
ACK_2, CHOOSE, CHOOSE2, MSG2, ACK2, MSG_Lost, MSG_Lost2,
Ack_arrived}

IL19 :{CLIENT, MSG, SERV, PROC, ACK, WAITDATA, DATA, 2*
ACK_2, CHOOSE, CHOOSE2, MSG2, ACK2, DATA2, ACK_22,
MSG_Lost, MSG_Lost2, Ack_arrived, }

IL20 :{SERV, ACK, WAITDATA, ACK_2, choose_to_serv, ACK2,
ACK_22}

IL21 :{SERV, PROC, DATA, ACK_2, choose_to_serv, DATA2,
ACK_22, MSG_Lost2}

IL22 :{CLIENT, WAIT_ACK, SERV, PROC, WAITDATA, WAITACK,
ACK_2, CHOOSE, CHOOSE2, MSG_Lost, MSG_Lost2, Ack_arrived}

IL23 :{CLIENT, WAIT_ACK, SERV, ACK, 2* WAITDATA, 2* ACK_2,
CHOOSE, CHOOSE2, ACK2, ACK_22, MSG_Lost, Ack_arrived}

IL24 :{CLIENT, MSG, ACK, WAITDATA, ACK_2, choose_to_client,
MSG2, ACK2, MSG_Lost, Ack_arrived}

IL25 :{CLIENT, WAIT_ACK, WAITDATA, ACK_2, choose_to_client,
MSG_Lost, Ack_arrived}

IL26 :{CLIENT, MSG, SERV, 2* PROC, 2* DATA, 2* ACK_2,
CHOOSE, CHOOSE2, MSG2, 2* DATA2, ACK_22, MSG_Lost, 2*
MSG_Lost2, Ack_arrived}

```

IL27 :{CLIENT, MSG, SERV, 2* PROC, DATA, WAITACK, ACK_2,
CHOOSE, CHOOSE2, MSG2, DATA2, MSG_Lost, 2* MSG_Lost2,
Ack_arrived}

IL28 :{MSG, ACK, DATA, ACK_2, CSMA_CA, MSG_Lost, MSG_Lost2}

IL29 :{CLIENT, MSG, SERV, 2* ACK, 2* WAITDATA, 2* ACK_2,
CHOOSE, CHOOSE2, MSG2, 2* ACK2, ACK_22, MSG_Lost,
Ack_arrived}

*-----
-----*

Place in all invariants:

P = {}

Places at none invariant:

P = {}

*-----
-----*

Sub-Net existency test (state machine):

Following invariants are state machine:

IL1 IL5 IL10 IL11 IL17 IL20 IL21 IL24 IL25 IL28

*-----
-----*

Performance Evaluation Oriented to STATES of Net S_R_1.

Inicial Marking : {CLIENT, SERV, CSMA_CA}

Desired Precision : 0.10 %

Max. of Fires : 210

Num. of Interactions : 2110

Improductive Interac.: 1 (0.05 %)

*-----
-----*

Average number of fires from cycle transitions :

(SEND: 1.47) (RECV: 2.98) (RECV1: 1.47) (SENDDATA:
1.47) (RECVDT: 2.92)

(ACKOK: 1.47) (CLIENTCH: 1.09) (SERVCH: 0.38) (SERVCH2:
1.09)

(CLIENTCH2: 0.38) (transito1: 2.98) (transito2: 1.47)
(transito3: 2.92)

(transito4: 1.47) (perda1: 2.95) (Re_TX1: 4.47) (perda2:
1.52)

(perda3: 2.97) (Re_TX2: 4.41) (perda4: 1.45) (noloss: 1.47)

*-----
-----*

Md 1: {CLIENT, SERV, CSMA_CA}

Average Time: 0.00 Deviation: 0.00 Probab: 99.95
% Reach: 2109

Minimum time of reaching : 0.00 Maximum time of
reaching : 0.00

*-----
-----*

Net Analysis S_R_1.

Visible Tr. : {}

Invisible Tr. : {all}

Inicial Mark. : {CLIENT, SERV, CSMA_CA}

*-----
-----*

Reduced graph was found :

S0 :

*-----
-----*