

Universidade do Estado do Rio Grande do Norte – UERN

Irlan Arley Targino Moreira

**Uma Abordagem Experimental do Protocolo Zigbee em um
Ambiente Simulado de Comunicação Intraveicular**

Natal – RN

2012

Irlan Arley Targino Moreira

**Uma Abordagem Experimental do Protocolo Zigbee em um
Ambiente Simulado de Comunicação Intraveicular**

Trabalho de Conclusão de Curso
apresentado à Universidade do Estado do
Rio Grande do Norte como um dos
requisitos para obtenção do título de
bacharel em Ciência da Computação.

Orientador:

Karla Darlene Nepomuceno Ramos

Co-Orientador:

Felipe Denis Mendonça De Oliveira

NATAL – RN

2012

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Moreira, Irlan Arley Targino

Uma Abordagem Experimental do Protocolo Zigbee em um Ambiente Simulado de Comunicação Intraveicular / Irlan Arley Targino Moreira – Natal, RN, 2012.

68f.

Orientadores: Karla Darlene Nepomuceno Ramos; Felipe Denis Mendonça De Oliveira

Monografia (bacharelado). Universidade do Estado do Rio Grande do Norte. Curso de Ciência da Computação

1. Computação – Monografia. 2. Rede de sensores sem fio. 3. Zigbee. I. Universidade do Estado do Rio Grande do Norte. II. Título.

UERN/BC

CDD004

IRLAN ARLEY TARGINO MOREIRA

**Uma Abordagem Experimental do Protocolo Zigbee em um
Ambiente Simulado de Comunicação Intraveicular**

Trabalho de Conclusão de Curso
apresentado à Universidade do Estado do
Rio Grande do Norte como um dos
requisitos para obtenção do título de
bacharel em Ciência da Computação.

Aprovado em ____/____/____.

Banca Examinadora

Karla Darlene Nepomuceno Ramos
Universidade do Estado do Rio Grande do Norte

Felipe Denis Mendonça De Oliveira
Universidade do Estado do Rio Grande do Norte

Isaac de Lima Oliveira Filho
Universidade do Estado do Rio Grande do Norte

Rodrigo Soares Semente
Universidade Federal do Rio Grande do Norte

Dedico este trabalho aos meus pais, Francisco de Assis e Margarida Maria, e ao meu irmão, Ítalo Matheus, que contribuíram com todo o apoio para que este trabalho pudesse ser realizado.

Agradecimentos

Primeiramente agradeço a Deus, por me mostrar o caminho a ser trilhado e por abrir tantas portas em minha vida, principalmente nos momentos mais difíceis. Aos meus pais que são os alicerces da minha vida, que me ensinaram a ser tudo que sou, que sempre me apoiaram e me deram forças, que me incentivaram e não me deixaram desistir, quem foram os meus primeiros professores e continuam me ensinando muito ainda, e, se eu consegui chegar até aqui, foi por causa deles. Ao meu irmão que nunca negou ajuda quando precisei, que me descontraía nos momentos de stress, que sempre tinha tempo para me ouvir. Ao meu amigo Marcelo que praticamente me mostrou o mundo da computação pelo qual me apaixonei. Aos meus tios, Marta e Otaniel, e à minha prima, Lígia, que me acolheram em sua casa quando cheguei nesta cidade. À toda minha família pelo carinho e por sempre acreditar em mim. Aos meus amigos, Isaiane, Thomás, Marianna, Felipe, Thaise, Hércules, Helder (Dynho), Adriana, Rafael, Dayanne Escalé e a todos os demais, que sempre estiveram ao meu lado durante a minha caminhada acadêmica e que tive o prazer de conhecê-los aqui em Natal; espero que permaneçam na minha vida. À Daiane Saíze, que me ajudou muito no estágio e me deu muito apoio. Aos meus amigos de São Miguel, não citarei nomes porque são muitos, que sempre me propiciaram momentos de descontração sempre que eu estava na minha cidade natal. À minha orientadora, professora Karla, que me mostrou um norte para a elaboração deste trabalho e por todo o apoio durante a construção do mesmo. Ao meu Co-orientador, professor Felipe Denis, que colaborou bastante com este trabalho. Às professoras Bartira e Camila pela amizade construída dentro e fora da sala de aula. Ao Sport Club Corinthians Paulista, por ter me dado a alegria da conquista da Copa Libertadores da América. Por fim, à todos aqueles que estiveram comigo durante esta caminhada e que sempre acreditaram em mim.

“Cada sonho que você deixa pra trás, é um pedaço do seu futuro que deixa de existir.”

Steve Jobs

Resumo

Atualmente os automóveis utilizam fios para realizar a comunicação entre o computador central e os demais dispositivos, o que acarreta em custos, dificuldade de instalação de equipamentos, e limitação no número de dispositivos instalados devido aos *slots* para conexões físicas. Uma possível solução para este problema é a implantação de uma rede sem fio que utilize um protocolo de comunicação que atenda as necessidades apresentadas. Portanto, este trabalho tem como objetivo estudar a inserção do protocolo Zigbee e das redes de sensores sem fio no ambiente intraveicular. Para testar a utilização desse protocolo no ambiente intraveicular, foi desenvolvida uma aplicação de sensoriamento crepuscular veicular, na qual foi possível analisar a potência de sinal e a perda de pacotes considerando a coexistência de redes como *Wi-Fi* e *Bluetooth*, bem como, interferências eletromagnéticas causadas por um motor. Para o desenvolvimento da aplicação foi utilizado o kit OEM Design da MEMSIC e os resultados foram considerados satisfatórios para o ambiente analisado.

Palavras-chave: Rede de sensores sem fio, Zigbee, rede intraveicular, aplicação de sensoriamento crepuscular.

Abstract

Currently vehicles using wires to carry the communication between the central computer and other devices, which results in cost, difficulty of installation of equipment, and limiting the number of devices installed due to slots for physical connections. One possible solution to this problem is the implementation of a wireless network that uses a communication protocol that meets the needs presented. Therefore, this paper aims to study the insertion of protocol Zigbee and wireless sensor networks in intra-vehicular environment. To test the use of this protocol in the intra-vehicle environment, an application was developed sensing twilight vehicle in which it was possible to analyze the signal strength and packet loss considering the coexistence of networks such as Wi-Fi and *Bluetooth*, as well as interference caused by an electromagnetic motor. For application development kit was used MEMSIC OEM design and the results were satisfactory for examined environment.

Keywords: Wireless sensor network, Zigbee, in-vehicle network, twilight sensing application.

Sumário

1.Introdução	1
1.1. Problema	2
1.2. Solução.....	3
1.3. Objetivos.....	4
1.4. Organização do Trabalho	5
2.Rede Intraveicular	6
2.1. Arquitetura das Redes Intraveiculares.....	6
2.2. Classes de Protocolos de Comunicação Intraveicular	7
2.3. Protocolo CAN.....	11
3.Redes de Sensores Sem Fio no Ambiente Intraveicular	14
3.1. <i>Bluetooth</i> (IEEE 802.15.1).....	15
3.2. Zigbee (IEEE 802.15.4)	17
3.2.1. Dispositivos	18
3.2.2. Arquitetura da Pilha de Protocolos.....	19
3.2.3. Camada de Aplicação	19
3.2.4. Camada de Rede	20
3.2.5. Padrão IEEE 802.15.4.	22
3.3. Coexistência com outras redes.....	26

4.Desenvolvimento de Aplicação de Sensoriamento Crepuscular para Rede Intraveicular.....	28
4.1. Aplicação e Materiais Utilizados	29
4.1.1. Nó Iris XM2110	29
4.1.2. Placa Sensor MDA100CB	31
4.1.3. Pod de Programação	32
4.1.4. Sistema Operacional TinyOs 2.x.....	32
4.2. Metodologia	33
4.2.1. Nó Sensor	34
4.2.2. Nó Central.....	37
4.2.3. Computador Central.....	38
4.3. Testes.....	39
4.3.1. Ambiente 1: Coexistência com WI-FI.....	40
4.3.2. Ambiente 2: Coexistência com o <i>Bluetooth</i>	41
4.3.3. Ambiente 3: Interferência Causada Por Um Motor.....	42
4.3.4. Ambiente 4: Coexistência com WI-FI, <i>Bluetooth</i> e Motor.....	43
5.Resultados	45
6.Conclusão	53
6.1. Trabalhos Futuros.....	54
Referências Bibliográficas	55
Apêndice A – Códigos da Aplicação do Nó Sensor e do Computador Central.....	59
Apêndice B – Imagens do Ambiente de Testes.....	68

Lista de Figuras

Figura 1 – Arquitetura de hardware de um nó sensor. Fonte: Silva (2006)	14
Figura 2 – Pilha de camadas do protocolo Zigbee. Fonte: (BARONT et al, 2007)	19
Figura 3 – Topologia de rede Zigbee. Fonte: (BARONTI et al, 2007).	22
Figura 4 – Coexistência entre padrões 802.15.4 e 802.11 b. Fonte: OLIVEIRA (2009)	26
Figura 5 – Coexistência entre padrões IEEE 802.15.1 e IEEE 802.15.4. Fonte: CHEN, L. (2006).....	27
Figura 6 – Mote IRIS XM2110. Fonte: CROSSBOW (2007)	29
Figura 7 – diagrama de bloco da arquitetura de Hardware do IRIS. Fonte: CROSSBOW (2007).....	30
Figura 8 – Placa sensor MDA100CB. Fonte: CROSSBOW (2007)	31
Figura 9 – Pod de Programação com conector de 10 pinos. Fonte: MEMSIC	32
Figura 10 – Diagrama de blocos do sistema de sensoriamento crepuscular.	33
Figura 11 – fluxograma de aplicação instalada no nó sensor	36
Figura 12 – Diagrama de blocos do ambiente de teste de coexistência Wi-Fi.	41
Figura 13 – Diagrama de blocos do ambiente de teste de coexistência <i>Bluetooth</i> . ..	42
Figura 14 – Diagrama de blocos do ambiente de teste de interferência eletromagnética.....	43
Figura 15 – Diagrama de blocos do ambiente de teste de coexistência Wi-Fi, <i>Bluetooth</i> e Motor.	43

Figura 16 – Gráficos de desempenho da rede Zigbee com a presença de rede Wi-Fi.	47
Figura 17 – Gráficos de desempenho da rede Zigbee com a presença de rede <i>Bluetooth</i>	48
Figura 18 – Gráficos de desempenho da rede Zigbee com a presença de um motor.	50
Figura 19 – Gráficos de desempenho da rede Zigbee com coexistência <i>Wi-Fi</i> , <i>Bluetooth</i> e motor.	52
Figura 20 – Ambiente de teste de coexistência Wi-Fi. Dispositivos Zigbee com círculos amarelos e dispositivos Wi-Fi com círculos vermelhos.	68
Figura 21 – Ambiente de teste de coexistência Bluetooth. Dispositivos Zigbee com círculos amarelos e dispositivos Bluetooth com círculos Azuis.	68
Figura 22 – Ambiente de teste de interferência eletromagnética. <i>Dispositivos Zigbee</i> <i>com círculos amarelos e motor com círculo verde</i>	69
Figura 23 – Ambiente de teste de coexistência Wi-Fi, Bluetooth e eletromagnética causada por motor. Dispositivos Zigbee com círculos amarelos, dispositivos Wi-Fi com círculos vermelhos, dispositivos Bluetooth com círculos azuis e motor com círculo verde.....	69

Lista de Tabelas

Tabela 1 – Protocolos de comunicação Classe A.	8
Tabela 2 – Protocolos de comunicação Classe B.	8
Tabela 3 – Protocolos de comunicação Classe C.	9
Tabela 4 – Protocolos de comunicação de Diagnóstico.	9
Tabela 5 – Protocolos de comunicação para aplicações Mobile Media.	10
Tabela 6 – Protocolos de comunicação para sistemas Safety Bus.	10
Tabela 7 – Protocolos de comunicação para sistemas Drive By Wire.	11
Tabela 8 – Parâmetros da camada PHY.	24
Tabela 9 – Comparação entre os protocolos <i>Bluetooth</i> e <i>Zigbee</i>	25
Tabela 10 – Corrente necessária para operações dos componentes do nó.	31
Tabela 11 – Medidas dos Tipos de Veículos.	40
Tabela 12 – Coleta de dados coexistência Wi-Fi.	45
Tabela 13 – Coleta de dados coexistência <i>Bluetooth</i>	47
Tabela 14 – Coleta de dados coexistência com motor – interferência eletromagnética	49
Tabela 15 – Coleta de dados coexistência Wi-Fi, <i>Bluetooth</i> e motor.	51

1. Introdução

Os computadores hoje não são apenas para uso geral, mas também para uso específico e a cada dia surge a necessidade da utilização de sistemas menores, com baixo consumo de energia, baixo custo e que possam ser utilizados em qualquer lugar.

Além da evolução dos equipamentos, outro grande avanço foi a transmissão de dados por meio do ar (*Wireless*) facilitando a instalação e configuração de redes sem a necessidade do cabeamento para interligar os equipamentos conectados e permitindo o movimento destes sem que haja a desconexão em um raio definido.

Com o avanço das técnicas de microprocessadores, redes sem fio e de novos materiais de sensoriamento estimulou-se a utilização de sensores conectados por uma rede sem fio para resolver problemas como monitoramento de animais, processos físicos e químicos, entre outros (LOUREIRO, et al., 2003). Portanto, com o objetivo de monitorar um dado fenômeno, surgiram as Redes de Sensores sem Fio (RSSF).

As RSSFs são compostas por um conjunto de nós que trocam informações, e cada componente possui quantidade de energia limitada (AKYILDIZ, et. al., 2002). Podem ser implantadas nos mais diversos tipos de ambientes, como em terrenos inacessíveis para outros equipamentos e seres humanos. Podem ser usadas em controle de produção em uma linha de montagem, monitorar ambientes internos ou externos, monitorar o tráfego de carros em avenidas, segurança de edifícios, entre outras aplicações.

RSSF engloba algumas frentes de pesquisa como as redes pessoais (em inglês, *Wireless Personal Area Network - WPAN*), redes terrestres, redes subterrâneas, redes aquáticas, redes multimídia (YICK; et al., 2008) e as redes veiculares (em inglês, *Vehicle Area Network*), que é abordada nessa pesquisa.

Considerando as redes veiculares, atualmente, é possível ouvir música, falar ao celular utilizando o sistema de áudio do veículo por meio de conexão sem fio e ser

guiado por um sistema global de posicionamento, por exemplo. Além das tecnologias citadas, os veículos interagem cada vez mais com o ambiente. Hoje existem sistemas que captam o nível de iluminação da estrada e automaticamente acendem os faróis, sistemas de detecção de chuva que liga os limpadores de para-brisa, entre outros.

As redes veiculares, também conhecidas como *Intelligent VAN* são redes formadas principalmente por veículos que interagem (1) com outros veículos (em inglês, *Vehicle-to-Vehicle* – V2V), (2) veículo com infra-estrutura (dispositivos espalhados pelas estradas) para transmitir e receber dados (em inglês, *Vehicle-to-roadside-Infrastructure* - V2I), (3) veículo com a nuvem (em inglês, *Vehicle-to-Broadband cloud* - V2B) e (4) a comunicação entre os dispositivos internos (em inglês, *Intra-Vehicle* - *In-Vehicle*) (FAEZIPOUR; et. al., 2012).

A comunicação *In-Vehicle*, cenário motivador desta pesquisa, é formada por uma rede interna composta por diversos nós sensores espalhados dentro do veículo. O principal foco é coletar informações com relação ao comportamento do carro e do condutor para que estas sejam analisadas e classificadas para evitar acidentes. Essas informações podem ser as mais diversas. Do veículo, dados como velocidade, pressão exercida sobre os pedais de freio e acelerador, rotação do volante e GPS; e do lado do condutor, informações como expressão facial (com o intuito de perceber sinais de fadiga ou sonolência), taxas de variação cardíacas e sudorese (informação capturada quando o motorista está tocando o volante) podem ser analisadas em conjunto (FAEZIPOUR; et. al., 2012).

1.1. Problema

Normalmente os veículos utilizam fios para realizar a comunicação entre seus dispositivos e o controlador central. A tendência é que mais dispositivos sejam instalados nos veículos aumentando a quantidade de fios que deverão ser o meio de comunicação. Porém, este tipo de infraestrutura pode se tornar mais cara, aumentar o nível de complexidade e o peso, o que pode prejudicar o desempenho do veículo (ROCA, 2008).

Com o aumento do número de dispositivos no veículo, a complexidade e a necessidade de comunicação entre eles também aumenta. É necessário realizar requisições de dados aos sensores, e esses dados devem ser processados para que alguma decisão seja tomada ou alguma mensagem seja enviada até ao condutor. Com a conexão com fios, o número de dispositivos conectados estará limitado ao número de portas existentes para a conexão com a unidade de processamento (RIBEIRO, 2008) e a quantidade de cabos utilizados pode interferir no conforto e no desempenho do veículo.

1.2. Solução

Uma possível solução seria trocar o sistema elétrico que faz a conexão desses dispositivos com o controlador central por uma rede sem fio intraveicular, aumentando assim a possibilidade de instalação de novos dispositivos de maneira rápida e fácil por não necessitar de instalação de cabos e fios, reduzindo gastos e aumentando desempenho e conforto.

Atualmente existem diversos protocolos utilizados no ambiente intraveicular para a comunicação e controle de equipamentos. Esses protocolos estão divididos em classes de acordo com suas características e funcionalidades e, na sua maioria, são proprietários, ou seja, são desenvolvidos por empresas ou conjuntos de empresas que estabelecem regras sobre seu uso, ocasionando a criação de diversos protocolos desenvolvidos para desempenharem a mesma função, mas que são utilizados por montadoras diferentes.

Esta pesquisa estuda a utilização do protocolo Zigbee no ambiente veicular, mais precisamente no ambiente interno. A partir de resultados alcançados durante esta pesquisa, este protocolo pode vir a ser uma possível solução para os problemas aqui apresentados.

Em (TSAI *et al*, 2007), são apresentados relatos de experiências de um caso de estudo, onde foram realizadas medições que comprovam o uso do Zigbee como uma possível tecnologia que pode vir a ser utilizada na implementação de uma rede de sensores sem fio no ambiente intraveicular. Nós sensores foram espalhados em um veículo e parâmetros como o LQI (qualidade do sinal), RSSI (potência do

sinal recebido), quantidade de pacotes perdidos e a checagem de redundância cíclica (*Cyclic Redundance Check Field – CRC*) foram utilizados para cálculos de métricas e avaliação de desempenho. Outra característica analisada foi o impacto que dispositivos que utilizam a tecnologia *Bluetooth* pode causar no desempenho das redes Zigbee. Com a análise dos dados, chegou-se a conclusão de que o Zigbee apresentou resultados satisfatórios para o possível uso no ambiente intraveicular, observando claro, as características da camada física.

Em (ELBATT *et al*, 2006) foi proposto uma rede de sensores sem fio com a finalidade de facilitar a comunicação entre sensores, interruptores e unidades de controle presentes no veículo. O foco principal é a análise da camada MAC buscando-se a obtenção de parâmetros baseados em requisitos de redes e dados como analisar os requisitos de latência por meio de sua quantificação. Por fim, foi analisado o potencial das redes de sensores sem fio no ambiente intraveicular para fins de monitoramento, controle e comunicação entre os componentes.

1.3. Objetivos

Para que os dispositivos se comuniquem é necessário estabelecer um padrão de comunicação e transferência de dados. Esses padrões são estabelecidos por meio de protocolos. Os dispositivos que utilizam o meio sem fio para a troca de informações utilizam alguns padrões definidos pela IEEE, por exemplo, o IEEE 802.11, IEEE 802.15.1 e o 802.15.4.

O objetivo deste trabalho é estudar a utilização do protocolo Zigbee em um ambiente de redes veiculares, mais precisamente analisar a transmissão de dados em um ambiente com interferências de outros protocolos de comunicação sem fio por meio de testes, observando a potência do sinal e o número de pacotes perdidos.

O estudo e a análise do protocolo ZigBee compreende os seguintes objetivos específicos: pesquisar os protocolos utilizados pela indústria automotiva, estudar as semelhanças entre tais protocolos e o zigbee, desenvolver uma aplicação semelhante ao sistema de sensoriamento crepuscular utilizado atualmente em alguns automóveis.

1.4. Organização do Trabalho

Este trabalho está organizado da seguinte forma: o capítulo 2 aborda o conceito das redes intraveiculares e os principais protocolos utilizados bem como sua classificação em classes. O capítulo 3 apresenta uma visão geral sobre as redes de sensores sem fio aplicada no ambiente intraveicular, características e tecnologias existentes e detalha o protocolo em estudo. O capítulo 4 descreve a aplicação de sensoriamento crepuscular implementada, plataforma utilizada e seu funcionamento. Por fim, no capítulo 5 é descrito como os testes foram realizados, seus resultados, a conclusão do trabalho e trabalhos futuros.

2. Rede Intraveicular

As redes intraveiculares interligam as unidades de controle elétrico dos veículos aos diversos dispositivos como sensores, atuadores, etc. O sistema de bordo se tornou um equipamento necessário no meio veicular. As redes intraveiculares são responsáveis por toda a rede de comunicação de equipamentos de bordo que avaliam tanto desempenho e segurança do veículo, quanto o comportamento do condutor (FAEZIPOUR *et al*, 2012).

Acidentes veiculares são a maior causa de mortes no mundo. Só no Brasil o número de acidentes com veículos envolvidos chegam a mais de 1,5 milhão por ano, sendo 35.000 de mortes diretas (WOLF *et al*, 2011). No ambiente intraveicular existem duas técnicas de segurança em veículos, a ativa e passiva (FAEZIPOUR *et al*, 2012). A segurança passiva está ligada no desenvolvimento de dispositivos ou ferramentas que agem na ocorrência de um acidente. Por exemplo, pode-se citar o cinto de segurança e o *air bag* que ao haver um acidente, esses dois sistemas trabalham para evitar maiores lesões aos passageiros. Já na segurança ativa são abordadas técnicas para dar assistência ao condutor e evitar, ou reduzir as chances de ocorrer um acidente, por exemplo, detecção de fadiga do condutor, detector de objetos na via, acendimento automático dos faróis, etc.

Segundo (GUIMARÃES, 2007 *apud* RIBEIRO, 2008) as redes intraveiculares podem ser divididas em duas arquiteturas: centralizadas e distribuídas. Além das arquiteturas, outro aspecto inerente a essas redes são os protocolos utilizados na comunicação. Portanto, na subseção 2.1 será discutida a arquitetura deste tipo de rede e na subseção 2.2 serão abordados os protocolos utilizados na comunicação das redes intraveiculares e a sua organização em classes.

2.1. Arquitetura das Redes Intraveiculares

A arquitetura centralizada possui uma única Unidade Eletrônica Central (*Electronic Central Unit* - ECU) que é responsável por avaliar as informações oriundas dos sensores, processar a informação e tomar decisões. A vantagem de se utilizar este tipo de arquitetura é a simplicidade do hardware, onde este recebe todos os

dados e não precisa de técnicas mais sofisticadas para coletar as informações. Como limitações, esta arquitetura apresenta o uso de grande quantidade de cabos para conectar todos os dispositivos e o número limitado de portas reduz a capacidade de expansão do sistema (JUNIOR, 2010).

Já na arquitetura distribuída existe mais de uma ECU e estas podem compartilhar informações e tarefas. Nessa arquitetura a quantidade de cabos é bastante reduzida, o tempo de montagem do veículo é reduzido e se uma ECU falhar não ocorre a falha total do sistema. Porém, esta arquitetura necessita de um meio de comunicação entre as ECUs e um protocolo de comunicação que faça a previsão de taxas de transmissão que a aplicação necessita (JUNIOR, 2010).

2.2. Classes de Protocolos de Comunicação Intraveicular

No ambiente intraveicular existem diversos tipos de protocolos que são utilizados para as mais diversas funcionalidades. Com a tentativa de agrupar esses protocolos de acordo com suas características, a Sociedade dos Engenheiros Automotivos (SAE – *Society of Automotive Engineers*) criou a definição das seguintes classes: A, B, C, Diagnóstico, *Mobile Media*, *Safety Bus* e *Drive by-wire* (RIBEIRO, 2008).

Na classe A estão incluídos os protocolos com taxa de transmissão de até 10 Kbps. Geralmente os protocolos desta classe estão relacionados com aplicações de conforto de um veículo, como controle de áudio e comunicação de sensores (ROCA, 2008; JUNIOR, 2010). A tabela 1 informa os protocolos pertencentes a esta classe. A maioria destes protocolos são proprietários.

Tabela 1 – Protocolos de comunicação Classe A.

	SINEBUS	SAE J1780	CCD	ACP	BEAN	LIN
Instituição Relacionada	Delco Eletronics	SAE/TCM	Chrysler	Ford	Toyota	Motorola
Aplicação Principal	Sistema de Áudio	Comunicação e diagnóstico	Comunicação e diagnóstico	Sistema de áudio	Comunicação e diagnóstico	Sensores e Atuadores
Taxa de Transmissão	66,6 Kbps - 200 Kbps	9,6 Kbps	7,81 Kbps	9,6 Kbps	10 Kbps	20 Kbps
Comp. Máx. de Barramento	10 metros	40 metros	-	40 metros	-	40 metros
Quantidade Máx. de Nós	-	20	10	20	20	16

Fonte: Baseado em ROCA (2008).

Nesta tabela, o protocolo SAE J1780 é uma tentativa de padronização da Sociedade dos Engenheiros Automotivos, os demais são especificados por seus fabricantes e o LIN está sendo largamente utilizado nos últimos anos em sub-redes CAN (RIBEIRO, 2008).

Os protocolos da classe B têm como característica a taxa de transmissão entre 10 Kbps à 125 Kbps. Normalmente estão incluídos os sistemas de entretenimento do veículo (ROCA, 2008). A tabela 2 mostra informações sobre alguns protocolos da Classe B e suas características.

Tabela 2 – Protocolos de comunicação Classe B.

	CAN 2.0 ISO 11898 ISO 11519-2	CAN 2.0 SAE J1780	J1859 Class 2	J1850 SCP	J1850 PCI
Instituição Relacionada	SAE ISO	SAE	GM	Ford	Chrysler
Aplicação Principal	Controle e diagnóstico	Controle e diagnóstico	Controle e diagnóstico	Controle e diagnóstico	Controle e diagnóstico
Taxa de Transmissão	10 Kbps - 1 Mbps	250 Kbps	10,4 Kbps	41 Kbps	10,4 Kbps
Comp. Máx. de Barramento	40 metros	40 metros	35 metros	35 metros	35 metros
Quantidade Máx. de Nós	32	32	32	32	32

Fonte: Baseado em ROCA (2008).

Todos os protocolos são proprietários desenvolvidos pela GM, Ford e Chrysler, menos o CAN (RIBEIRO, 2008).

Os protocolos da classe C são aqueles com taxa de transmissão entre 125 Kbps a 1 Mbps. Normalmente estão relacionados à sistemas de segurança de um veículo e controle em tempo real (RIBEIRO, 2008). O protocolo CAN também aparece nesta classe, como é mostrado na tabela 3.

Tabela 3 – Protocolos de comunicação Classe C.

	CAN 2.0 ISO 11898 ISO 11519-2	CAN 2.0 SAE J139
Instituição Relacionada	SAE & ISO	SAE
Aplicação Principal	Controle e diagnóstico	Controle e diagnóstico
Taxa de Transmissão	10 Kbps - 1 Mbps	250 Kbps
Comp. Máx. de Barramento	40 metros	40 metros
Quantidade Máx. de Nós	32	32

Fonte: Baseado em ROCA (2008).

A classe diagnóstico é composta pelos protocolos utilizados em sistema de diagnóstico embarcado (JUNIOR, 2010). A tabela 4 mostra os protocolos desta classe.

Tabela 4 – Protocolos de comunicação de Diagnóstico.

	J1850 Class 2	J1850	J1850 PCI	ISO 9141	Keyword
Instituição Relacionada	GM	Ford	Chrysler	ISO	ISO
Aplicação Principal	Controle e diagnóstico	Controle e diagnóstico	Controle e diagnóstico	Diagnóstico	Diagnóstico
Taxa de Transmissão	10,4 Kbps	41,6 Kbps	10,4 Kbps	10,4 Kbps	5 bps - 10,4 Kbps
Comp. Máx. de Barramento	35 metros	35 metros	35 metros	-	-
Quantidade Máx. de Nós	32	32	32	-	10

Fonte: Baseado em JUNIOR (2010).

Na *Mobile Media* estão incluídos os protocolos que permitem a comunicação de dispositivos com os computadores dos veículos proporcionando funções de entretenimento (RIBEIRO, 2008).

Tabela 5 – Protocolos de comunicação para aplicações *Mobile Media*.

	IDB-C	MOST	MML	USB	IEEE 1394
Instituição Relacionada	SAE	Oasis	Delco	Commercial	IEEE
Aplicação Principal	Entretenimento em Geral	Controle e transmissão de voz e imagem	Controle e transmissão de voz e imagem	Dispositivos Baseados em Computadores Pessoais	Dispositivos Baseados em Computadores Pessoais
Taxa de Transmissão	250 Kbps	25 Mbps	110 Mbps	12 Mbps	90 - 393 Mbps
Comp. Máx. de Barramento	35 metros	35 metros	35 metros	-	-
Quantidade Máx. de Nós	16	24	16	127	16

Fonte: Baseado em JUNIOR (2010)

Na classe *Safety Bus* estão incluídos os protocolos de comunicação que atuam em sistemas de segurança para condutor e passageiro. Tem como característica a comunicação em alta taxa de amostragem, porém de pequenos pacotes de 1 a 2 bytes e a maioria destes protocolos são proprietários.

Tabela 6 – Protocolos de comunicação para sistemas *Safety Bus*.

	BST	Safe By Wire	DSI	Byteflight
Instituição Relacionada	Bosh - Siemens Temic	Delphi - Phillips - TRW - Autoliv - SDI	Motorola	BMW
Aplicação Principal	Airbag	Airbag	Airbag	Airbag
Taxa de Transmissão	31,25 Kbps ou 250 Kbps	150 Kbps	5 Kbps - 150 Kbps	10 Mbps
Comp. Máx. de Barramento	-	25 - 40 Metros	-	-
Quantidade Máx. de Nós	12 squibs + 62 slaves	64	16	-

Fonte: Baseado em JUNIOR (2010)

Na classe By-Wire estão incluídos os protocolos que trabalham em sistemas by-wire. Sistemas *drive by-wire* vieram para substituir acionamentos mecânicos por acionamentos elétricos, como freio, acelerador, direção, etc. (RIBEIRO, 2008). A tabela 7 lista os protocolos de comunicação desta classe.

Tabela 7 – Protocolos de comunicação para sistemas Drive By Wire.

	TTP	FlexRay	TTCAII
Instituição Relacionada	U-VIENNA	Motorola	CAN in Automation
Aplicação Principal	Controle de Segurança	Controle de Segurança	Controle de Segurança
Taxa de Transmissão	-	10 Mbps	1 - 2 Mbps
Comp. Máx. de Barramento	-	-	40 metros
Quantidade Máx. de Nós	-	-	32

Fonte: Baseado em JUNIOR (2010)

Considerando o uso do protocolo CAN nas classes B e C, definidas pelo SAE, e ainda, que esta pesquisa investiga o protocolo Zigbee, a próxima seção detalha o protocolo CAN com a finalidade de identificar as características que se aproximam do protocolo investigado.

2.3. Protocolo CAN

O CAN (*Controller Area Network*) foi desenvolvido por Robert Bosch Gmb na década de 80 com a finalidade de ser utilizado para fornecer a conexão entre dispositivos de controle em automóveis. Hoje pode encontrar vários padrões derivados do CAN em diversas outras máquinas como satélites, caminhões, máquinas agrícolas, etc. (SOUZA, 2002).

Tornou-se um padrão ISO 11898 para aplicações de alta velocidade (125 Kbps à 1 Mbps) e ISO 11519-2 para aplicações de baixa velocidade (10 Kbps à 125 Kbps) e foi desenvolvido para rápida troca de dados (ROCA, 2008). Estes padrões apresentam camada física e enlace de acordo com a camada OSI. As camadas superiores não são definidas permitindo que o desenvolvedor crie o protocolo que melhor atenda as necessidades da aplicação (SOUZA, 2002).

Segundo (SOUZA, 2002; ROCA, 2008; RIBEIRO, 2008, GUIMARÃES E SARAIVA, 2002) Este protocolo possui as seguintes características:

- **Arquitetura multi-mestre:** módulos possuem direitos iguais, ou seja, todos podem assumir função de mestre ou escravo possibilitando acesso ao meio sempre que necessitar enviar uma mensagem.
- **Configuração da taxa de transmissão:** pode ser configurável para transmitir poucos Kbps até 1 Mbps.
- **Barramento com 1, 2 ou 4 fios:** o barramento com apenas um fio terá apenas o fio de dados, conhecido como *CAN*. Já os outros dois barramentos trabalham com sinais *CAN H (high)* e *CAN L (low)* e deverão ser trançados, não necessariamente blindados e os dados são analisados pela diferença de potencial entre os fios *CAN H* e *CAN L*. Esta técnica é conhecida como par trançado diferencial e elimina as interferências eletro-magnéticas, pois se um dos fios sofrer alteração, o outro também sofrerá e o que importa neste caso é a diferença de potencial.
- **Mecanismo de controle de colisões:** utiliza o *CSMA/CD with NDA (Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration)*. Trabalha semelhante ao *CSMA/CD* utilizado no *Ethernet*, sendo que este possui prioridade de tempo definida, ou seja, o módulo com menor prioridade esperará a transmissão da mensagem pelo módulo com maior prioridade.
- **Flexibilidade de instalação:** novos dispositivos podem ser instalados ou removidos sem necessitar de configuração prévia de *hardware* ou *software*.
- **Possibilidades de implementação de rede:** permite a implementação de redes ponto-a-ponto, por multidifusão (de um dispositivo para um grupo de dispositivos) ou por difusão (de um para todos os dispositivos).

Considerando as informações sobre a arquitetura e os protocolos relacionados à infraestrutura das redes intraveiculares apresentados nesse capítulo, percebe-se a importância e o interesse, tanto da academia como da indústria em identificar novos modelos e protocolos de comunicação, que reduzam a quantidade de fios.

Uma das características observadas no CAN que é semelhante ao Zigbee, é a sua utilização em aplicações de baixa velocidade, onde as taxas variam de 10 Kbps

à 125 Kbps. Além da taxa de transmissão, os dois protocolos permitem flexibilidade de instalação, e ambos permitem diversas possibilidades de implementação de rede. Essas características sobre o Zigbee podem ser encontradas no capítulo 3.

(ROCA, 2008) apresenta no seu trabalho uma rede de sensores para aquisição de dados para veículos como uma alternativa de comunicação neste ambiente possibilitando a diminuição da instalação de cabos para a instalação de equipamentos. (JUNIOR, 2010), por sua vez, apresenta diversas tecnologias sem fio, inclusive o Zigbee com o propósito de inovação na comunicação entre dispositivos, principalmente sensores, e a ECU.

No próximo capítulo serão abordados os principais conceitos das redes sem fio e do protocolo *Bluetooth* e Zigbee, com ênfase nesse último detalhando suas características, tipos de dispositivos, camadas da pilha de protocolos e coexistência com outras redes.

3. Redes de Sensores Sem Fio no Ambiente Intraveicular

As RSSF aplicadas no ambiente intraveicular é uma proposta nova e promissora que trás novas ideias para minimizar e resolver problemas das redes atuais que possuem os fios como principal meio de comunicação entre os dispositivos.

As redes sem fio podem ajudar na diminuição dos custos e tempo de montagem, além de contribuir com o desempenho do veículo, com a eliminação da fiação que segundo (AHMED *et al*, 2007), dependendo da estrutura de cabeamento do veículo, poderá *conter* mais de 4.000 peças, pesando aproximadamente 40 Kg em um total de 4 Km de fios.

No geral, as redes de sensores possuem elementos que são de suma importância para o seu funcionamento, são eles (ZHAO *et al*, 2004; SILVA, 2006):

- **Sensor:** elemento que transforma um fenômeno físico em um sinal elétrico para ser manipulado por outros equipamentos.
- **Nó sensor, nó ou mote:** unidade básica de uma RSSF. Possui elementos como microprocessador, memória, bateria, unidade de rádio e um ou mais sensores conectados ou integrados. A figura 1 ilustra a arquitetura de hardware de um nó sensor:



Figura 1 – Arquitetura de hardware de um nó sensor. Fonte: Silva (2006)

- **Sink:** é o nó coordenador que concentra as informações recebidas dos demais nós sensores e a envia para o gateway. Pode ter um sensor ou não.
- **Gateway:** Equipamento utilizado para receber as informações coletadas pelos nós sensores ou do estado da rede. Também pode realizar requisições ou enviar comandos à rede.

- **Estação base:** computador que se comunica diretamente com o *gateway* e pode armazenar os resultados em um banco de dados e disponibilizá-los ao usuário.
- **Atuador:** dispositivo capaz de modificar valores no ambiente para corrigir falhas ou controlar o objeto monitorado.

As RSSF estão sujeitas a limitações energéticas, nós com baixa capacidade de processamento, memória e largura de banda limitadas. Usualmente, os recursos de processamento e armazenamento são utilizados apenas para processamento de sinal e agendamento de tarefas (ZHAO *et al*, 2004).

As RSSF são diferentes das redes convencionais devido aos tipos de aplicações que são desenvolvidas e ao ambiente onde são implantadas. Segundo (LOUREIRO *et al*, 2003), possuem como principais características o endereçamento dos nós onde cada nó pode ter mais de um endereço dependendo da aplicação; agregação de dados que colabora com a diminuição das transferências realizadas dentro da rede; mobilidade dos sensores; quantidade de sensores; limite de energia; auto-organização da rede; capacidade de realizar tarefas colaborativas e coexistência de outras redes.

Considerando a utilização das RSSF em um ambiente intraveicular, as próximas seções apresentam os protocolos utilizados nesse ambiente. Na subseção 3.1 serão abordadas as principais características do protocolo *Bluetooth*. Em seguida, na subseção 3.2 será discutido o protocolo Zigbee, suas características, tipos de dispositivos, arquitetura do protocolo, o padrão IEEE 802.15.4 e uma breve comparação entre os protocolos *Bluetooth* e Zigbee. Na subseção 3.3 será discutido a coexistência de outras redes.

3.1. *Bluetooth* (IEEE 802.15.1)

O *Bluetooth* é uma tecnologia de comunicação sem fio muito utilizada atualmente com o propósito de facilitar a conexão e o compartilhamento de arquivos multimídia. Está classificado dentro das redes pessoais (*Wireless Personal Area Network* – WPAN) e pode ser encontrado principalmente em dispositivos como

celulares, vídeo games, computadores portáteis, tablets, impressoras, mouses, teclados, aparelhos de som, etc.

Foi publicado por um consorcio de empresas composto inicialmente por 3COM, Ericsson, IBM, Intel, Lucent, Microsoft, Motorola, Nokia e Toshiba (OLIVEIRA, 2009) e surgiu com intuito de substituir os cabos de comunicação de dispositivos fixos e móveis (IEEE 802.15.1, 2002).

O padrão 802.15.1 implementa as camadas de acesso ao meio (*Medium Access Control - MAC*) e física (*Physical - PHY*). O *Bluetooth* opera na banda de frequência não licenciada ISM (Industry, Science, Medical) de 2,4 GHz. Utiliza a técnica de *Frequency Hopping Spread Spectrum (FHSS)* que realiza saltos de frequência para evitar interferências (IEEE 802.15.1, 2002). Um transceptor com uma frequência de 1600 saltos/s é utilizado para combater a fadiga e interferência desta faixa. A distância entre equipamentos que utilizam o *Bluetooth* pode chegar a cerca de 10 metros de distância (OLIVEIRA, 2009).

A frequência da banda é de 2.400 MHz à 2483.5 MHz, sendo esta dividida em 79 canais (enumerados de 0 a 78) e o espaçamento entre eles é de 1 MHz. A taxa de transmissão pode chegar à 1 Mbps (IEEE 802.15.1, 2002).

A rede é formada por um conjunto de dispositivos que compartilham o mesmo canal e são sincronizados por um relógio e um padrão de saltos de frequência em comum. O dispositivo responsável por fornecer essa sincronização é conhecido como mestre (*master*). Os demais dispositivos são conhecidos como escravos (*slaves*). Dessa forma, a rede criada é conhecida como *piconet*. Um *piconet* pode ser formado por até 7 dispositivos, sendo um nó mestre e os demais escravos (IEEE 802.15.1, 2002).

O padrão de saltos de frequência dos dispositivos em um *piconet* é determinado algoritmicamente pelos campos de endereço e pelo relógio do mestre. O padrão obedece a uma ordem pseudo-aleatória de 79 frequências. Os saltos podem ser adaptados para excluir parte das frequências que possam estar sendo utilizadas por outros dispositivos e que possam causar interferência (IEEE 802.15.1, 2002).

O canal é dividido em unidades de tempo (*slots*). Os dados são transmitidos por meio de pacotes que são posicionados nos *slots*. Cada pacote é transmitido em diferentes frequências na sequência de saltos (OLIVEIRA, 2009). O salto de frequência ocorre no início da transmissão de cada pacote (IEEE 802.15.1, 2002).

Há uma maneira de expandir o número de dispositivos *Bluetooth* conectados. Esta técnica é conhecida por Scatternet. A scatternet é a união entre uma ou mais piconet. Esta união acontece quando um dos dispositivos de uma piconet participa como escravo de outra rede. O dispositivo pode participar de duas redes como escravo, ou como mestre de uma e escravo de outra (ERICSSON Technology, 2004).

Porém, o protocolo *Bluetooth* não implementa o Scatternet, sendo necessário a sua implementação na pilha de host, o que pode comprometer a interoperabilidade entre os seus subsistemas, tornando as suas entidades dependentes uma das outras, indo contra um dos objetivos deste protocolo (ERICSSON Technology, 2004).

Segundo (CHEN et al, 2005) o *Bluetooth* pode ser abordado no ambiente veicular em dois níveis, sendo o nível de sistema onde pode ser utilizado para substituir os fios em uma rede de controle não crítica facilitando o diagnóstico e reparação de equipamentos, e o nível de dispositivos de usuário fornecendo interface de conexão amigável entre dispositivos pessoais.

3.2. Zigbee (IEEE 802.15.4)

O Zigbee é um consórcio de software, hardware e companhias de serviços que desenvolveram um padrão comum para gestão de redes de transmissão de dados sem fio de sensores e controladores. (JUNIOR, 2010). A Zigbee Alliance, nome dado ao consórcio, criou um protocolo de comunicação, nomeado por Zigbee, baseado no padrão IEEE 802.15.4 e foi validado em dezembro de 2004. O nome, por sua vez, se deu a partir do seu funcionamento em malha e da analogia com as abelhas, pois as abelhas voam em zig-zag trocando informações entre si sobre distância, direção e localização de alimentos (OLIVEIRA, 2009).

Foi desenvolvido para ser um padrão de comunicação sem fio de baixo custo, baixo consumo de energia e baixa potência onde não sejam necessárias altas taxas de transferência (ZHAO et al 2004). É utilizado em sistemas embarcados como

produtos eletrônicos, automatização, controle industrial, brinquedos e aplicações que utilizam sensores médicos.

O Zigbee é implementado sobre o padrão IEEE 802.15.4, que por sua vez, define as camadas física e de enlace. As demais camadas são definidas pelo próprio Zigbee.

3.2.1. Dispositivos

Os dispositivos podem ser tratados de forma física e lógica. Como dispositivo físico, eles podem ser dispositivos com funções completas (*Full Function Device – FFD*) e dispositivos com funções reduzidas (*Reduced Function Device – RFD*).

Os FFDs são dispositivos que podem exercer qualquer tipo de função e podem fazer parte de qualquer topologia de rede. Além disso, estes podem se comunicar com qualquer tipo de nó e consomem mais energia que os dispositivos mais simples devido a sua complexidade ou função exercida na rede.

Já os RFDs, ao contrário dos FFDs, são dispositivos limitados que não podem exercer a função de coordenador e roteador, apenas se comunicam com os FFDs e consomem menos energia.

Como dispositivos lógicos podem assumir a função de coordenador (*Coordinator*), roteador (*Router*) e dispositivo final (*End Divece*). Os coordenadores têm como principal função fornecer o canal de operação e o número lógico para formar a rede, serve como ligação entre redes Zigbee distintas e permite que roteadores e dispositivos finais se integrem à rede. O único dispositivo que pode realizar a função de coordenador é o FFD.

Os roteadores podem expandir a rede sem a necessidade de um coordenador. Isso é possível, pois este tipo de dispositivo guarda as informações da rede e a partir dos dados pode traçar rotas para o tráfego de dados. Também é um dispositivo FFD.

Por último, o dispositivo final pode tanto transmitir como receber informações, contudo, não pode redirecioná-las. Comunica-se apenas com o nó coordenador ou

com o roteador. Este tipo de nó pode ser tanto um FFD, como um RFD, apesar deste último ser mais aconselhável por consumir menos energia.

3.2.2. Arquitetura da Pilha de Protocolos

A pilha é constituída de quatro camadas, onde as duas últimas (abordagem top-down) são definidas pelo padrão 802.15.4. Cada camada especifica um conjunto de funções para a camada mais a cima. Uma entidade de dados fornece serviços para a transmissão dos dados e a entidade de gerenciamento fornece todos os demais serviços. Cada entidade de serviço possui uma interface para a comunicação com a camada superior por meio de um ponto de acesso de serviço (*Service Point Access - SAP*) e cada interface possui uma quantidade se serviços primitivos que podem ser suportados (Zigbee Specification, 2008). A pilha de camadas pode ser vista na figura 2. O Zigbee é responsável pelas camadas de aplicação e rede.

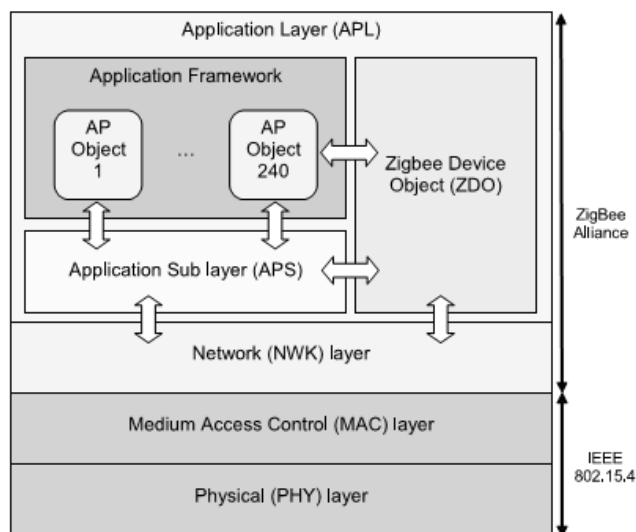


Figura 2 – Pilha de camadas do protocolo Zigbee. Fonte: (BARONT et al, 2007)

3.2.3. Camada de Aplicação

Como pode ser visualizada na figura 3, a camada de aplicação é composta por uma APS (*Application Suport Sub-layer*), *Application Framework* e o ZDO (*Zigbee Device Object*).

A APS fornece uma interface entre a camada de rede (*Network - NWK*) e a camada de aplicação por meio de um conjunto de serviços que são usados pelo ZDO e pelos objetos da aplicação (*Application Object – AP Object*). Estes serviços

são fornecidos pela entidade de dados da aplicação (APSDE) que fornece o serviço de transmissão de dados entre as entidades de aplicações que estão em uma mesma rede, e a entidade de gerenciamento de serviços (APSME) que fornece uma série de serviços, tais como serviço de segurança e de ligação de dispositivos (ZIGBEE Specification, 2008).

O *Application Framework* é responsável por hospedar os *AP Object*. Cada aplicação pode possuir 240 objetos. Cada objeto é endereçado com valores que variam de 1 a 240. O endereço zero é reservado para a interface de dados do ZDO e o 255 para uma interface para a comunicação com todos os objetos. Os endereços entre 241 e 254 são reservados para uso futuro (ZIGBEE Specification, 2008).

E por último, o ZDO é responsável por fornecer serviços que permitam um *AP Object* descobrir outros *AP Objects* para que estes se organizarem em uma aplicação distribuída.

3.2.4. Camada de Rede

A camada de rede fornece funcionalidades para o bom funcionamento da camada MAC e fornece serviços para a interface que faz a sua ligação com a camada de aplicação. É responsável por oferecer mecanismos para que dispositivos possam (ZIGBEE Specification, 2008):

- **Inicializar a rede:** capacidade de estabelecer uma nova rede;
- **Configuração de um novo dispositivo:** permitir que novos dispositivos possam se conectar à rede ou permitir que um dispositivo comece como coordenador;
- **Segurança:** garantir autenticidade e confiabilidade de uma transmissão;
- **Endereçamento:** permitir que coordenadores Zigbee ou roteadores distribuam endereços aos dispositivos pertencentes à rede;
- **Descoberta de vizinhos:** permitir que dispositivos descubram, relatem e registrem informações de vizinhos que estão à um salto de distância;
- **Descoberta de rota:** descobrir caminhos na rede por onde as mensagens possam ser transmitidas de forma mais eficiente;

- **Controle de recepção:** permitir que dispositivos realizem a sincronização para recepção de dados;
- **Roteamento:** realizar diferentes mecanismos de roteamento de dados, tais como unicast, multicast e broadcast para troca eficiente de dados;

A camada de rede do Zigbee suporta três tipos de topologias: estrela, malha e árvore (ZIGBEE Specification, 2008).

Na topologia em estrela, a rede é controlada por apenas um nó central, ou seja, um nó coordenador que é responsável por inicializar, manter e gerenciar toda a rede. Os demais componentes são conhecidos como nós terminais e se comunicam diretamente com o nó coordenador.

As topologias árvore e malha possuem uma característica em comum: as redes podem ser expandidas utilizando roteadores, mas o nó coordenador continua sendo o responsável por iniciar a rede e definir os seus parâmetros. A principal diferença entre a topologia em malha e a de árvore é que nessa última, os roteadores trafegam os dados e controlam as mensagens de acordo com uma estratégia de roteamento hierárquico, podendo a rota de transmissão ser baseada no mapa de energia da rede. Também pode implementar a comunicação orientada a beacons (ZIGBEE Specification, 2008).

Na topologia em árvore, a rede é composta na sua maioria por dispositivos FFDs que podem realizar o papel de coordenador. Os dispositivos RFDs são os dispositivos folhas e se comunicam apenas com dispositivos FFDs. É baseada na comunicação ponto-a-ponto (JUNIOR, 2010).

Na topologia em malha a comunicação entre os dispositivos ocorrem ponto-a-ponto. Devido aos roteadores permitirem a expansão da rede, esta pode chegar a ter 65.000 dispositivos conectados. Esta topologia é indicada para aplicações que necessitem ser extendidas (OLIVEIRA, 2009). A figura 3 ilustra as três topologias de rede:

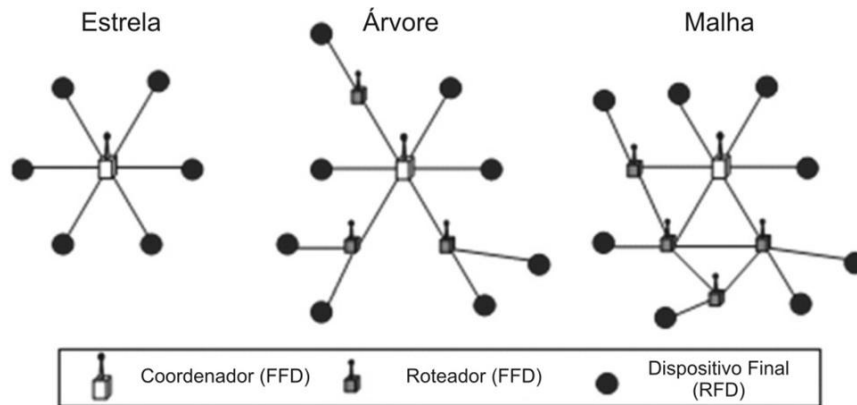


Figura 3 – Topologia de rede Zigbee. Fonte: (BARONTI et al, 2007).

As camadas de controle de acesso ao meio e a camada física serão descritas na próxima sessão que tratará do padrão 802.15.4.

3.2.5. Padrão IEEE 802.15.4.

Como já foram mencionadas anteriormente, as camadas de controle de acesso ao meio (*Medium Access Control – MAC*) e física (*Physical – PHY*) são definidas pelo padrão IEEE 802.15.4. As principais características desse padrão são (IEEE 802.15.4, 2003):

- Transferência de dados de 250 Kbps, 40 Kbps e 20 Kbps;
- Topologia estrela e ponto a ponto;
- Endereçamento curto de 16 bits ou estendido de 64 bits;
- Alocação de slots garantido no tempo (*Guaranteed Time Slot - GTS*);
- Acesso ao canal com o CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance);
- Baixo consumo de energia;
- Detecção de energia (*Energy Detection - ED*);
- Indicador de qualidade de conexão (*Link Quality Indication - LQI*);
- 16 canais na banda de 2,4 GHz, 10 canais na banda de 915 MHz e 1 canal na banda de 868 MHz;

A. Camada MAC

A camada MAC é responsável por todo o acesso ao canal de rádio e é responsável por gerar redes *beacons* se o dispositivo for um coordenador,

sincronizar os *beacons*, dá suporte à associação e dissociação, dá suporte à segurança dos dispositivos, implementar o mecanismo de CSMA-CA para acesso ao canal, manusear e realizar manutenção no mecanismo de GTS e fornecer conexão confiável entre duas entidades MAC. Esta camada destina-se basicamente a formatar as mensagens recebidas pela camada física, processá-las e em seguida repassar para a camada de rede (IEEE 802.15.4, 2003).

O protocolo CSMA/CA é um protocolo de acesso ao meio utilizado para evitar colisões entre pacotes transmitidos pelos dispositivos conectados à rede. Antes de transmitir o pacote, o dispositivo escuta o meio de comunicação para averiguar se o meio está livre ou não. Caso esteja, o pacote será enviado, caso contrário, o dispositivo irá esperar um tempo aleatório e logo após verificará o canal novamente. Cada nó deve informar a intenção de transmissão aos demais nós. (OLIVEIRA, 2009).

Segundo (SILVA, 2011) existem dois mecanismos utilizados para o acesso ao meio: beacon¹ habilitado e beacon não habilitado.

No modo beacon habilitado, existe uma estrutura chama de superframe que fornece sincronização entre dispositivos. Esta sincronização evita que outros dispositivos tentem transferir dados ao mesmo tempo evitando assim as colisões. Para que um nó tenha acesso ao meio de transmissão ele deve então receber um quadro beacon. Estes quadros não são endereçados previamente a nenhum dispositivo, ou seja, são transmitidos para todos os dispositivos que estejam escutando o canal. Finalmente o dispositivo é sincronizado quando um coordenador beacon liga o seu receptor ao mesmo tempo em que um quadro é esperado (OLIVEIRA, 2009).

Já o modo beacon desabilitado o nó acessa um canal por meio do mecanismo CSMA/CA para realizar a transmissão não havendo assim a sincronização entre os dispositivos (OLIVEIRA, 2009).

¹ Sinal emitido para sincronização de dispositivos.

B. Camada PHY

A camada PHY é responsável por ativar e desativar o transceptor de rádio, medição da energia dentro do canal, LQI (Link Quality Indication) para recebimento dos pacotes, avaliação de canal livre (*Clear Channel Assessment - CCA*) para o CSMA/CA e recepção e transmissão de dados.

São disponibilizados dois padrões de camada física, onde ambos operam na faixa não licenciada (*ISM – Industrial, Scientific, Medical*). Um dos padrões trabalha com duas faixas: 868 e 915 MHz utilizadas na América do Norte, Europa, Nova Zelândia e Austrália. O outro padrão é utilizado no mundo todo e trabalha na faixa de 2,4 GHz (IEEE 802.15.4, 2003).

No total, existem 27 canais disponíveis, numerados de 0 a 26, entre as bandas de frequência citadas. Na banda de 868 MHz existe apenas 1 canal disponível (canal 0), 10 canais (canais de 1 a 10) na banda de 915 MHz e por último, 16 canais (canais de 11 a 26) na banda de 2,4 GHz (IEEE 802.15.4, 2003). O centro de frequência de cada canal pode ser calculado com a seguinte fórmula:

$$F_c = 868.3 \text{ em MHz, para } k = 0$$

$$F_c = 906 + 2 (k - 1) \text{ em MHz, para } k = 1, 2, \dots, 10$$

$$F_c = 2405 + 5 (k - 11) \text{ em MHz, para } k = 11, 12, \dots, 26$$

Onde k é o número do canal.

Este padrão utiliza o modo de acesso espalhamento de espectro direto (*Direct Sequence Spread Spectrum – DSSS*) que gera um padrão redundante de bits para cada bit transmitido (CAMPOS, 2006). A banda de 2,4 GHz utiliza o chaveamento por deslocamento de fase em quadratura (*Offset Quadrature Phase Shift Keying – O-QPSK*) e as outras duas bandas utilizam o chaveamento binário de fase (*Binary Phase Shift Keying – BPSK*). A banda passante entre os canais é de 2 MHz e o espaçamento entre a frequência central é de 5 MHz. A tabela 8 trás um breve resumo dos principais parâmetros da camada física.

Tabela 8 – Parâmetros da camada PHY.

Faixa de Frequência (MHz)	Modulação	Taxa de Transmissão (Kbps)	Número de Canais
868 - 868,6	BPSK	20	1
902 - 928	BPSK	40	10
2400 - 2483,5	O-QPSK	250	16

Fonte: Campos (2006)

Com base nas características dos protocolos *Bluetooth* e *Zigbee*, a tabela 9 mostra uma comparação entre os mesmos. Nessa tabela estão descritas a banda de atuação, canais utilizados, taxa de transferência, área de cobertura, aplicações que geralmente utilizam esses protocolos, número de dispositivos em uma rede e custos de tecnologia.

Tabela 9 – Comparação entre os protocolos *Bluetooth* e *Zigbee*

Características	<i>Bluetooth</i>	<i>Zigbee</i>
Banda de atuação	2,4 GHz	868 MHz, 915 MHz e 2,4 GHz.
Canal	79 canais	16 canais nos 2,4 GHz, 10 canais nos 915 MHz e 1 canal nos 868 MHz.
Taxa de transferência	Até 1 Mbps.	Até 250 Kbps.
Cobertura	10 metros aprox.	70 – 300 metros.
Aplicações típicas	Voz e Dados	Dispositivos de baixa potência e baixa velocidade, sensores, controladores e monitoramento remoto.
N.º máximo de dispositivos	Até 7 em um piconet, mas pode ser estendida por meio do Scatternet	Até 65.000
Custo	Médio	Baixo

Analisando a tabela 9, nota-se que o *Zigbee* pode ser utilizado em diversas áreas e tipos de aplicações se comparado ao *Bluetooth*. Mesmo o *Bluetooth* podendo ter a sua quantidade de dispositivos estendidos, o Scatternet não é padrão do protocolo, devendo-se ser implementada por terceiros. Já o *Zigbee* possui a possibilidade de grandes quantidades de nós em uma rede como característica do próprio protocolo, é mais adequado para aplicações que exijam baixa taxa de transferência e para aplicações que utilizam sensores. Outra Vantagem do *Zigbee* é o seu baixo custo.

3.3. Coexistência com outras redes

Com a grande propagação de aparelhos que utilizam a tecnologia *Wi-Fi* e *Bluetooth*, é importante observar como essas redes se comportam com o Zigbee. Para a coexistência entre redes *Wi-Fi* 802.11 b/g/n é aconselhável o uso dos canais 15, 20, 25 e 26 para o Zigbee, pois tais canais não se sobrepõem aos canais do padrão IEEE 802.11 (PETERSEN, 2007).

As redes *Wi-Fi* 802.11 b/g/n e 802.11 p (padrão utilizado para comunicações de redes veiculares do tipo V2I e V2B) também podem trabalhar na faixa de 5 GHz, mas não interferem no protocolo aqui estudado. A figura 4 ilustra a coexistência entre os canais utilizados pelos padrões 802.11 b e 802.15.4.

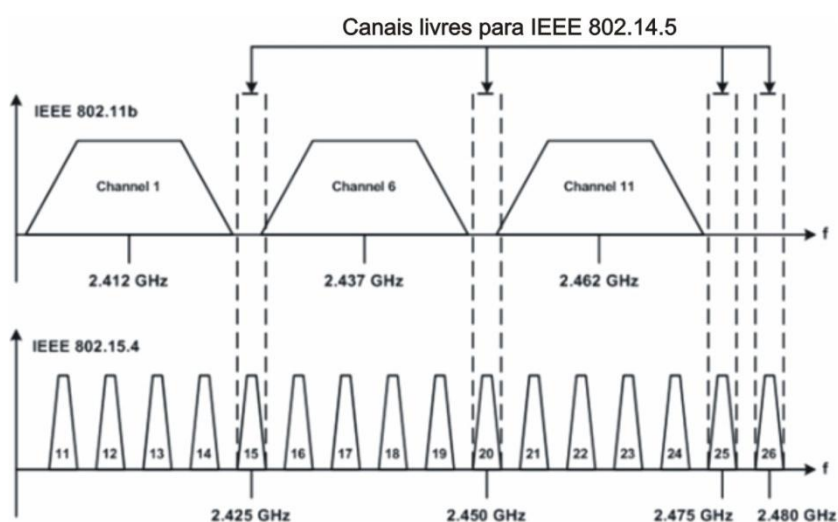


Figura 4 – Coexistência entre padrões 802.15.4 e 802.11 b. Fonte: OLIVEIRA (2009)

O *Bluetooth* também trabalha na faixa de frequência de 2,4 GHz e, diferentemente da comparação anterior, há sobreposição de todos os canais IEEE 802.15.4 sobre os canais IEEE 802.15.1. Porém, com a técnica de FHSS empregada no padrão IEEE 802.15.1, onde são realizados saltos de canais quando está recebendo ou transmitindo pacotes, a interferência entre as redes pode ser mínima. A figura 5 ilustra a coexistência entre os canais desses dois padrões.

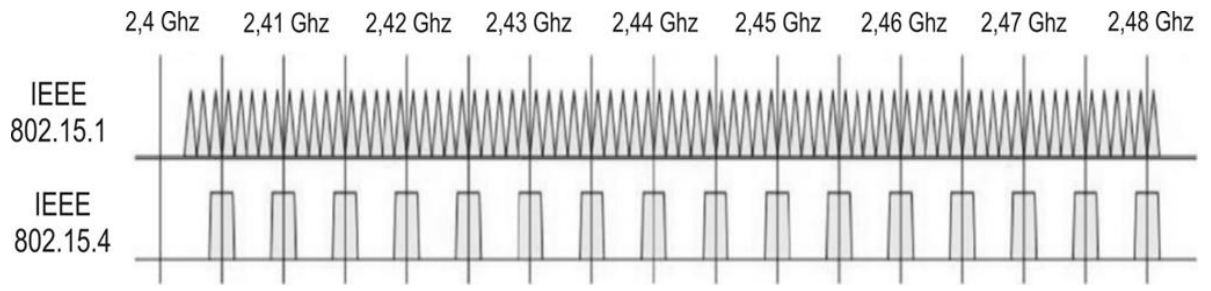


Figura 5 – Coexistência entre padrões IEEE 802.15.1 e IEEE 802.15.4.
Fonte: CHEN, L. (2006).

4. Desenvolvimento de Aplicação de Sensoriamento Crepuscular para Rede Intraveicular

O ambiente veicular está utilizando cada vez mais uma grande quantidade de dispositivos, provocando uma demanda por protocolos que suportem a construção de redes com um número considerável de dispositivos. O Zigbee, como pode ser constatado no capítulo anterior, é um protocolo com grande capacidade de expansão de rede. Dependendo da topologia escolhida, pode chegar a 65.000 dispositivos. O *Bluetooth*, no entanto, tem capacidade padrão para apenas 7 dispositivos, podendo ter este total aumentado utilizando Scatternet. Porém, esta técnica não é padrão do protocolo, devendo-se ser implementada, o que pode causar a interoperabilidade entre os subsistemas do protocolo.

O Zigbee não fornece altas taxas de transferência de dados, mas com taxas de até 250 Kbps, pode ser utilizado nas classes A, B, Diagnóstico e Safety Bus, conforme explicitado na seção 2.2. O Zigbee opera nas bandas de 868 MHz, 915 MHz e 2,4 GHz e dependendo da taxa de transferência exigida e do tipo de função desempenhada, as três bandas de operação poderão ser utilizadas.

Apesar de o ambiente veicular, atualmente, não possuir restrições quanto à fonte de alimentação, pois os automóveis geram energia, o Zigbee possui mecanismos para gerenciar o consumo permitindo aos dispositivos a possibilidade de hibernação. Esta característica é interessante para as pesquisas sobre veículos ecologicamente corretos.

Com base nos aspectos citados, e considerando ainda que a área de cobertura do Zigbee é suficiente para o ambiente intraveicular, pois os nós podem ficar separados por uma distância de até 300 metros (essa distância depende do tipo de ambiente), bem como, o custo dos equipamentos são considerados baixo, foi desenvolvida uma aplicação para analisar a transmissão dos dados em relação à interferências que podem ocasionar perda de dados. As seções que seguem apresentam a aplicação desenvolvida, os materiais e a metodologia utilizada, bem como os testes realizados.

4.1. Aplicação e Materiais Utilizados

A aplicação desenvolvida consiste no sensoriamento crepuscular, que capta o nível de luz presente no ambiente. O nó sensor captura a informação do nível de luz, envia um sinal para o computador central que irá avaliar o dado recebido e tomar a decisão se deverá apagar ou acender os faróis.

Para o desenvolvimento da aplicação foi utilizado o kit de OEM Design da MEMSIC. Este kit fornece dispositivos que trabalham na frequência de 2,4 GHz e é composto por módulos de programação, sensores, e módulos de rádio. Na aplicação de sensoriamento crepuscular foram utilizados os equipamentos listados abaixo, cujos detalhes são apresentados em seguida:

- Dois nós Iris XM2110;
- Uma placa sensor MDA100CB com uma fotocélula;
- Um gateway MIB520 (pod de programação);
- TinyOS 2.1.1;
- Um notebook com sistema operacional Ubuntu 12.04;

4.1.1. Nó Iris XM2110

Nó desenvolvido pela Crossbow, criado para ser utilizado em redes de sensores sem fio de baixo consumo de energia. Utiliza o transceptor de rádio frequência de baixa potência AT86RF230, integrado com o microcontrolador Atmega 1281 e um conector de 51 pinos (CROSSBOW, 2007). Cada componente será descrito a seguir. A Figura 6 exhibe o mote Iris, e a Figura 7 exhibe o diagrama de arquitetura de hardware.



Figura 6 – Mote IRIS XM2110. Fonte: CROSSBOW (2007)

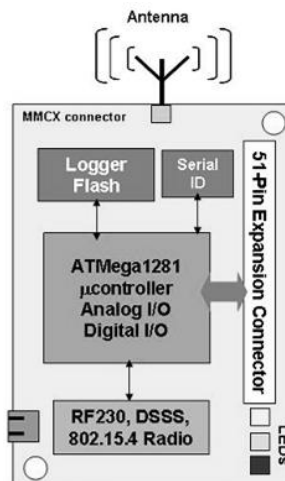


Figura 7 – diagrama de bloco da arquitetura de Hardware do IRIS. Fonte: CROSSBOW (2007)

OAT86RF230 é um transceptor que foi criado para aplicações Zigbee/802.15.4. Inclui modulação DSSS e taxa de transmissão de dados efetiva de 250 Kbps. A potência é programável de -17,2 dBm a 3 dBm e a sua sensibilidade de recepção é de -101 dBm (ATMEL, 2009).

O AT45DB041 (Logger Flash) é a memória flash utilizada no mote Iris. Seu tamanho é de 4 Mbits que é dividida em 2048 páginas de 264 bytes cada. Além da memória principal, essa memória possui dois *buffers* SRAM de 264 bytes que permitem a recepção de dados enquanto uma página da memória principal está sendo reprogramada (ATMEL, 2001).

A alimentação do nó é fornecida por duas pilhas alcalinas, tamanho AA. Entretanto, qualquer tipo de bateria também pode ser adaptado desde que a saída esteja entre 2,7 VDC e 3,6 VDC (CROSSBOW, 2007). A tabela 10 informa o consumo de alguns componentes que compõe o mote.

Tabela 10 – Corrente necessária para operações dos componentes do nó.

Atividade em funcionamento	mA
Processador em pleno funcionamento	8 (7,73 MHz)
Processador inativo	0,008
Rádio – Recepção	16
Rádio – Transmissão (1mW)	17
Rádio inativo	0.001
Memória flash serial – Escrita	15
Memória flash serial – Leitura	4
Memória flash serial – Inativa	0.002

Fonte: CROSSBOW (2007)

O conector de 51 pinos é utilizado para a agregação de sensores, programação dos motes e conexão de *gateway*.

4.1.2. Placa Sensor MDA100CB

O MDA 100CB é uma placa que possui os sensores de temperatura e de luminosidade. O sensor de temperatura possui alta precisão (precisão de 0,2 °C pode ser alcançada) e é altamente estável. A resistência do termistor varia com a temperatura, sendo a resistência base de 10 KΩ a 25 °C (CROSSBOW, 2007).

O sensor de luminosidade é uma fotocélula CdSe (seleneto de cádmio) simples. A sensibilidade máxima da fotocélula é no comprimento de onda de luz de 690nm. Sua potência típica, quando exposta à luz é de 2 KW e em ambientes escuros é de 520 KW (CROSSBOW, 2007). A placa sensor MDA100CB, apresentada na figura 8, possui área de prototipagem para a ligação de outros sensores e dispositivos no nó.

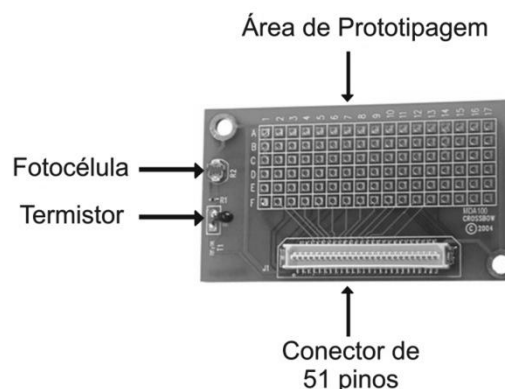


Figura 8 – Placa sensor MDA100CB. Fonte: CROSSBOW (2007)

4.1.3. Pod de Programação

O pod de programação é um *gateway* baseado no MIB520 e fornece comunicação USB entre a estação base e a rede de sensores sem fio, ou seja, realiza o papel do *gateway*. Também é utilizado para inserir nos nós a aplicação que se deseja utilizar. A aplicação é transferida do computador por meio da porta USB e chega ao nó por meio do conector de 10 pinos. Para a família de motes Iris, porém, este conector possui uma placa conversora para o conector de 51 pinos (MEMSIC).

Sua alimentação é realizada por meio do barramento USB. O pod de programação disponibiliza dois tipos de portas virtuais, uma para a programação dos motes e a outra para comunicação entre a rede e a estação base. Para a comunicação entre o computador e a RSSF é necessário que um mote fique conectado ao pod para a recepção do sinal sem fio. Este mote roda a aplicação da estação base (MEMSIC). A figura 9 mostra o pod de programação com o conector de 10 pinos.



Figura 9 – Pod de Programação com conector de 10 pinos. Fonte: MEMSIC

4.1.4. Sistema Operacional TinyOs 2.x

O TinyOs é um sistema operacional, desenvolvido pela Universidade de Berkeley com a colaboração da Intel, simples, de baixa complexidade desenvolvido para ser utilizado em dispositivos de baixa potência em uma rede de sensores sem fio oferecendo um conjunto de funcionalidades e abstrações que facilitam o desenvolvimento dessas redes como sensoriamento, comunicação, armazenamento e temporizadores (BERNARDO, 2011).

As aplicações e o próprio sistema é escrito na linguagem NesC. O NesC é um dialeto derivado da linguagem C, orientado à eventos, com recursos para diminuição

de memória RAM e tamanho do código, ajuda a reduzir os riscos de erros de baixo nível como condições de corrida.

4.2. Metodologia

A aplicação está dividida basicamente em três módulos: o nó sensor (escravo), o nó central (mestre ou *sink*), e o computador central (estação base). O diagrama de blocos da figura 10 ilustra os módulos utilizados.

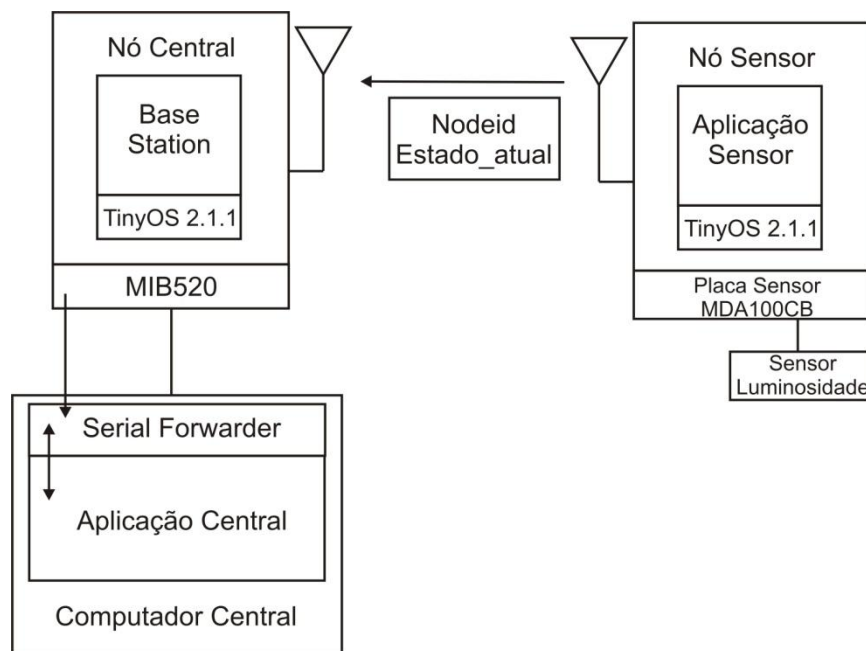


Figura 10 – Diagrama de blocos do sistema de sensoriamento crepuscular.

O nó sensor é responsável por capturar as informações e repassá-las para o nó central. A mensagem transmitida do nó sensor contém informações como o id do nó transmissor e a informação que representa o estado do ambiente (claro ou escuro). O nó central é responsável por receber as informações do nó sensor e encaminhá-las para o computador central por meio do *gateway* (MIB520); e o computador central faz a análise dos dados recebidos e toma decisões.

No nó central foi utilizado uma aplicação disponível nos diretórios do TinyOS 2.1.1. Ambas as aplicações dos motes foram escritas na linguagem NesC e a aplicação do computador foi escrita em Java.

A versão do TinyOS utilizada é a 2.1.1. Este sistema operacional foi escolhido por auxiliar no desenvolvimento de aplicações devido a sua baixa complexidade,

gerenciamento de concorrência de tarefas e facilidade de aprendizado propiciada pela semelhança entre a linguagem C e NesC.

A topologia escolhida foi a estrela, onde um nó possui a placa sensor conectada a ele, por meio do conector de 51 pinos, e esse transmite os dados coletados para o nó central. O nó central é conectado ao MIB520, também pelo conector de 51 pinos, que por sua vez, é conectado ao computador pelo barramento USB. Portanto, a aplicação completa está dividida em três partes que são descritas a seguir.

4.2.1. Nó Sensor

Responsável por capturar as informações da luminosidade ambiente e transmiti-las ao nó central. Existem duas variáveis que tem como papel fundamental identificar se o ambiente está escuro ou claro. A variável denominada *estado_atual* é responsável por armazenar os valores que indicam os dois estados que o ambiente pode assumir, e a outra variável é denominada *estado_anterior*. Esta é responsável por guardar a informação do estado do ambiente na coleta realizada anteriormente. O entendimento destas duas variáveis é necessário para a compreensão do funcionamento da aplicação.

A coleta da informação do nível de iluminação é realizada a cada 250 milissegundos. Após a amostragem, o valor obtido é comparado com o valor que foi estabelecido como limite para a definição de claro ou escuro. Se a amostra possuir o valor inferior à 700^2 unidades, então a variável *estado_atual* armazena o valor 1 (representa a ausência de luz). Caso o valor obtido na coleta de iluminação do ambiente seja maior ou igual à 700, então o valor armazenado na variável *estado_atual* é 2 (representa a presença de luz). Ao iniciar a aplicação, a variável *estado_anterior* recebe o valor 0.

² Valor de saída do conversor ADC. Definido de acordo com testes expondo o mote à luz do sol ao final da tarde. Não foi definida uma unidade de medida universal para a luz porque não foi encontrada a fórmula devida para a conversão da saída apresentada pelo mote nos manuais de referência do sensor.

Após realizada a coleta e atribuição de valores para as variáveis, o algoritmo verifica se o transmissor está desocupado, caso afirmativo, o valor da variável *estado_atual* é comparado ao valor da variável *estado_anterior*. Se essa comparação ocorre quando a aplicação foi inicializada, então os valores comparados não serão iguais, o valor de *estado_atual* será atribuído à *estado_anterior* e a mensagem será carregada com o valor da variável *estado_atual* para que esta seja transmitida para o nó central.

Todo o processo se repete até a comparação entre as variáveis. Após a amostragem, a variável *estado_atual* receberá o valor que representa o estado atual do ambiente e é realizada novamente a comparação com a variável *estado_anterior*, que dessa vez possui o valor representativo do estado anterior do ambiente. Ao comparar as variáveis, se ambas possuírem o mesmo valor, nenhuma mensagem é transmitida, pois o estado do ambiente não foi alterado. Caso contrário, se os valores forem diferentes, isso implica que o estado do ambiente alterou e uma nova mensagem é transmitida para o nó central. Esta lógica evita que mensagens desnecessárias sejam transmitidas sem a alteração do estado do ambiente. O fluxograma a baixo ilustra os passos descritos até aqui.

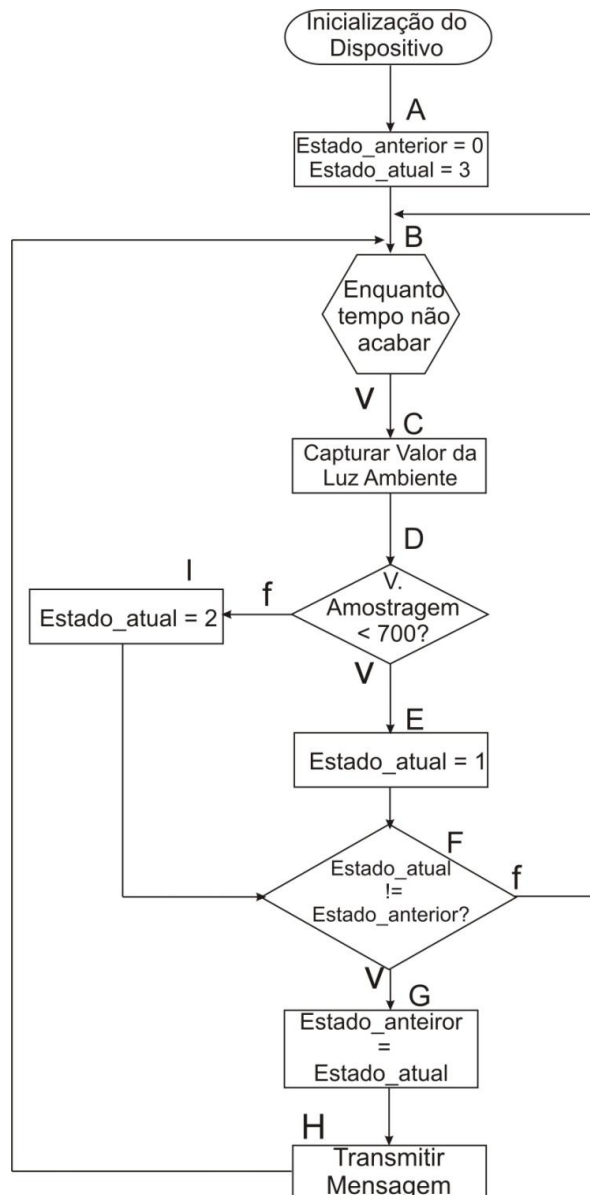


Figura 11 – fluxograma de aplicação instalada no nó sensor

A – Ao iniciar o nó sensor, as variáveis *estado_anterior* e *estado_atual* serão iniciadas com valores 0 e 3, respectivamente. Estes valores não representam nenhum dado, sendo apenas para inicialização das variáveis.

B – Depois da inicialização das variáveis, um temporizado de 250 milissegundos será iniciado. Quando este tempo for esgotado, será iniciada a captura da informação da intensidade de luz presente no ambiente.

C – Será coletado o valor da luz ambiente por meio do sensor e este valor será armazenado em uma variável local.

D – O valor contido na variável local será comparado com um valor fixo. Este valor fixo representa o limite entre o estado de ambiente claro ou escuro. Se o valor obtido for maior ou igual à 700, então o estado é considerado claro. Caso o valor obtido seja menor que 700, o estado do ambiente é considerado escuro.

E – O valor da amostragem foi menor que 700. Então a variável *estado_atual* receberá o valor 1, que representa a ausência de luz no ambiente.

I – O valor de amostragem foi maior ou igual à 700. Então a variável *estado_atual* receberá o valor 2, que representa a presença suficiente de luz no ambiente.

F – O valor da variável *estado_atual* será comparado com o valor da variável *estado_anterior*. Se os valores das variáveis forem iguais, significará que o estado do ambiente não foi alterado, então a coleta da luz ambiente será realizada novamente após o sinal do temporizador.

G – Se os valores das variáveis forem diferentes, significará que o estado do ambiente foi alterado e o valor da variável *estado_atual* será atribuído à variável *estado_anterior*.

H – Após a constatação da mudança de estado do ambiente, a mensagem contendo o id do nó transmissor e o valor do estado será enviada para o nó central.

A mensagem que é enviada do nó sensor para o nó central é definida no arquivo *luminosidade.h*. Dentro deste arquivo existe além de definições como tempo em que o nó sensor irá realizar a leitura da luminosidade, uma estrutura semelhante à utilizada na linguagem C que define quais e que tipo de informações serão enviadas. A estrutura *LuminosidadeMsg* possui dois elementos, o *nodeid* e *data*, ambos do tipo *uint16_t*. O primeiro elemento é necessário para identificar o identificador do nó que está transmitindo a mensagem e o segundo elemento armazenará o estado do ambiente (armazenará 1 para ausência de luz e 2 para presença de luz).

4.2.2. Nó Central

Neste dispositivo foi instalado a aplicação *BaseStation* encontrada nos diretórios de aplicação do TinyOS. Ele tem papel fundamental para a comunicação entre a

rede e o computador central. Ele envia e recebe dados dos dispositivos que compõem a RSSF, como também do computador por meio do barramento serial. Implementa filas em ambos os sentidos. Ao entrar na fila, a mensagem tem garantia de que será transmitida ou para a interface serial ou para a interface de rádio.

4.2.3. Computador Central

A aplicação que é executada no computador central é baseada no código Java pertencente à aplicação *RssiDemo* encontrada nos diretórios de aplicações do TinyOS 2.1.1. A partir de um conjunto de ferramentas fornecidas pelo TinyOS, todos os tipos e constantes do NesC podem ser mapeadas para classes Java.

Após definir a porta que a aplicação irá escutar, uma conexão é criada e a partir dela, a classe *MoteIF* é instanciada. Para que as mensagens sejam recebidas, é necessário chamar o método *RegisterListener*, passando ao mesmo um objeto do tipo *LuminosidadeMsg* que é a mensagem que será recebida. Esta classe é gerada a partir da definição de mensagem criada para os motes no arquivo *luminosidade.h*. Para a criação desta classe é utilizado o programa gerador de interface de mensagens (*message interface generator – mig*). Esta criação pode acontecer de maneira automática, desde que o comando necessário esteja especificado no arquivo de compilação automática (*Makefile*).

Ao receber a mensagem, a aplicação analisará a informação recebida contida no campo *data*. Se o valor for igual a 1, então o programa irá interpretar que os faróis deverão acender, caso o valor seja igual a 2, será interpretado que os faróis deverão apagar.

Além da aplicação descrita a cima, é utilizada a aplicação *Serial Forwarder* também disponibilizada pelo tinyOS. Esta aplicação permite que várias aplicações possam se comunicar com a RSSF, o que não é possível se a comunicação for realizada diretamente pela porta serial e também permite que essas aplicações possam ser executadas remotamente.

4.3. Testes

A aplicação de sensoriamento crepuscular foi testada em quatro ambientes distintos, ambos *indoor*. Foi verificada a interferência de sinais de outras redes, bem como a interferência eletromagnética causada por um motor, que pudessem prejudicar a transmissão de dados da rede Zigbee. Os requisitos analisados foram a potência do sinal emitido pelo nó sensor e recebido pelo nó central e a perda de pacotes.

Para análise da potência do sinal foi utilizado o Indicador de força do sinal recebido (*Received signal strength indication – RSSI*). Segundo (ROCA, 2008), como o próprio nome sugere, o RSSI é uma medida de força do sinal recebido que se refere apenas à energia do sinal e não a sua qualidade. Essa métrica é muito utilizada nos receptores de rádios comuns.

Segundo (ATMEL, 2009), o RSSI é um registro de 8 bits, mas seu valor é representado pelos cinco bits mais baixos e seu intervalo é de 0 a 28. Para um valor RSSI igual a zero, indica um sinal de potência menor que -91 dBm e para um valor 28, indica um sinal de potência maior ou igual a -10 dBm. Para um valor entre 1 e 28, a potência pode ser calculada de acordo com a seguinte fórmula:

$$P_{RF} = RSSI_BASE_VAL + 3 * (RSSI - 1)$$

Sendo *RSSI_BASE_VAL* definido como -91 e *RSSI* é o valor que varia de 0 a 28.

Para o cálculo da perda de pacotes foi implementado dois contadores, um no nó sensor que é incrementado quando um pacote é transmitido, e o outro na aplicação Java que está presente no computador central, que é incrementado quando um pacote é recebido. No final do teste, com os resultados dos dois contadores pode-se calcular a porcentagem de pacotes perdidos.

Tanto para a verificação da potência do sinal, quanto para o cálculo de pacotes perdidos, a aplicação do nó sensor e a aplicação do computador central sofreram modificações. Na mensagem foram inseridos mais dois campos, um para o valor do RSSI e o outro para o número de pacotes enviados.

Cada ambiente foi submetido à coleta de dados do valor do RSSI e de pacotes enviados e recebidos. Foram realizadas três amostras, cada uma com a transmissão de 1.000 pacotes. Com relação à distância, foram utilizados cinco comprimentos diferentes referentes a modelos de veículos como: popular, minivan, SUV, esportivo e ônibus. A medida de cada tipo de veículo pode ser analisada na tabela a seguir.

Tabela 11 – Medidas dos Tipos de Veículos

Tipo de veículo	Medida (m)
Popular	3,78
Minivan	4,04
SUV	4,58
Esportivo	4,84
Ônibus	11,10

Quatro ambientes foram criados para a aplicação dos testes para a análise de interferência: coexistência com *Wi-Fi*, coexistência com *Bluetooth*, interferência eletromagnética causada por um motor e, por último, o conjunto de todas as situações. Em todos os casos, o nó central e o nó sensor foram configurados com a potência de 3 dBm, potência máxima, e o canal 26 para evitar maiores interferências com tecnologias baseadas no padrão 802.11 que trabalham na banda de 2.4 GHz.

4.3.1. Ambiente 1: Coexistência com WI-FI

No primeiro ambiente, além da existência da rede Zigbee, foram inseridos dois equipamentos que utilizam a tecnologia *Wi-Fi*, um *Access Point (AP)*, que trabalha com os padrões 802.11 b/g/n e possui taxa de transmissão de até 150 Mbps e um Notebook, possui placa DW1501 Wireless-N WLAN que também trabalha com os padrões 802.11 b/g/n, ambos na banda de 2.4 GHz.

Para gerar o tráfego de dados entre os dois dispositivos *Wi-Fi*, o notebook foi submetido à transferência de um arquivo de tamanho relevante enquanto a rede de sensor estava gerando tráfego de dados. No momento do teste, o próprio notebook estava coletando as informações necessárias para a análise. O diagrama de blocos da figura 12 ilustra o ambiente *Wi-Fi*.

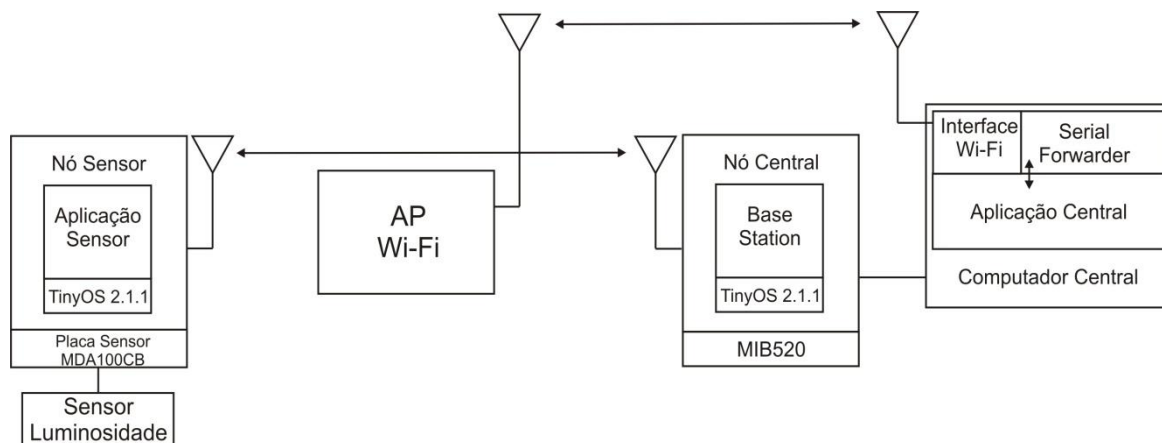


Figura 12 – Diagrama de blocos do ambiente de teste de coexistência Wi-Fi.

Neste ambiente pode-se notar a presença do nó a troca de mensagens entre os dispositivos que utilizam o Zigbee (Nó sensor e Nó central) e dois dispositivos que utilizam o Wi-Fi (AP e Computador Central). O AP ficou fixado a uma distância de aproximadamente 0,5 metros do nó sensor, enquanto que o computador estava à aproximadamente 0,05 metros do nó central. O nó sensor e o nó central estiveram separados por cinco distâncias diferentes. Estas distâncias podem ser verificadas na tabela 11. Também pode-se notar a presença do bloco Wi-Fi e de uma antena no computador central demonstrando a existência da comunicação Wi-Fi neste equipamento.

4.3.2. Ambiente 2: Coexistência com o *Bluetooth*

O segundo ambiente possui a presença de dispositivos que utilizam a tecnologia *Bluetooth*. Foram utilizados dois celulares realizando transferência de dados. Para a aplicação do teste foram realizadas diversas transferências de arquivos entre os dois celulares enquanto os dispositivos Zigbee estavam em operação. A figura 13 ilustra o ambiente com a presença de dispositivos *Bluetooth* realizando transferência de dados.

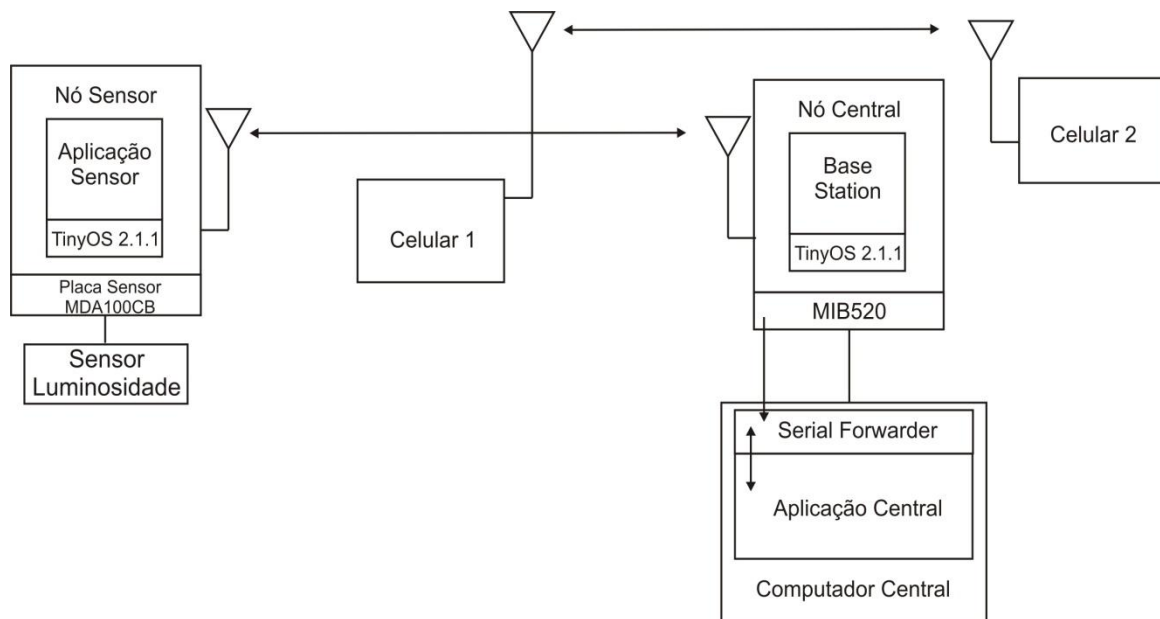


Figura 13 – Diagrama de blocos do ambiente de teste de coexistência *Bluetooth*.

Neste ambiente pode-se observar a presença de dois dispositivos que utilizam a tecnologia Zigbee, e dois celulares que utilizam a tecnologia *Bluetooth*. Os celulares estavam a uma distância de aproximadamente 0,05 metros do nó central. O bloco de interface Wi-Fi não está presente no bloco do computador central, neste diagrama, porque não há conexão Wi-Fi neste ambiente. As mesmas distâncias foram aplicadas neste teste.

4.3.3. Ambiente 3: Interferência Causada Por Um Motor

O terceiro ambiente possui a presença de um motor que, segundo (ROCA, 2008), o seu funcionamento pode causar ruído eletromagnético nos sinais dos dispositivos Zigbee. Para simular o motor de um veículo foi utilizado um liquidificador com potência de 350 W. O diagrama de blocos da figura 14 ilustra o ambiente com a presença de um motor.

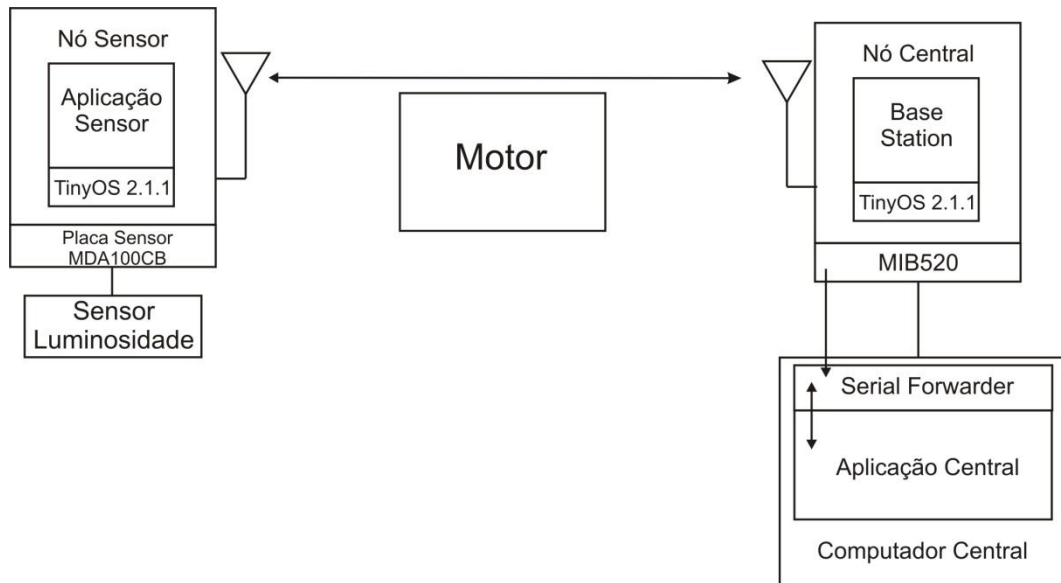


Figura 14 – Diagrama de blocos do ambiente de teste de interferência eletromagnética.

O motor foi colocado entre o nó sensor e o nó central a uma distância de aproximadamente 0,5 metros do nó sensor. Também foram aplicadas as mesmas distâncias dos ambientes anteriores.

4.3.4. Ambiente 4: Coexistência com Wi-Fi, Bluetooth e Motor

O quarto ambiente de testes considerou a presença de todos os dispositivos utilizados nos ambientes anteriores aumentando assim o número de possíveis interferências. O diagrama de bloco da figura 15 mostra o cenário com a presença de todos os dispositivos.

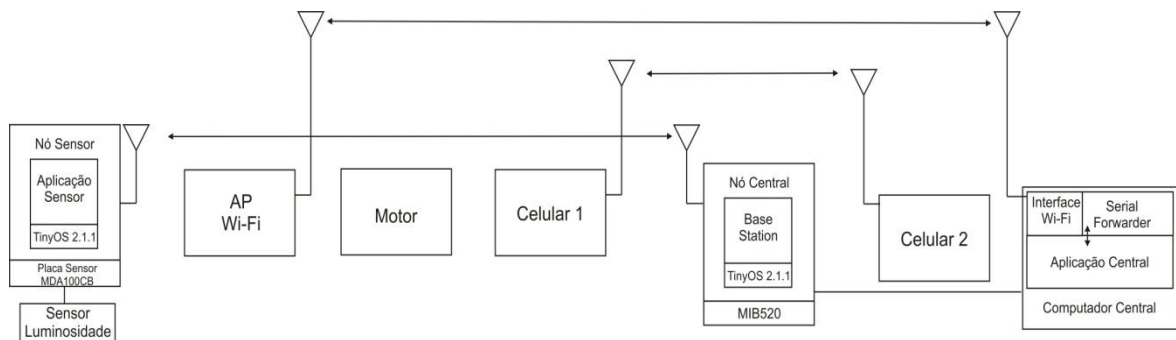


Figura 15 – Diagrama de blocos do ambiente de teste de coexistência Wi-Fi, Bluetooth e Motor.

Neste ambiente foram inseridos todos os dispositivos já mencionados nos ambientes anteriores. A distância entre os dispositivos Wi-Fi, Bluetooth e o motor

são as mesmas aplicadas nos outros cenários e a distância entre os nós são as descritas na tabela 11.

5. Resultados

Este capítulo apresenta os resultados obtidos nos testes realizados em quatro ambientes distintos. Em cada ambiente foram realizados cinco testes mudando a distância entre os nós de acordo com o comprimento de alguns tipos de veículos que podem ser vistos na tabela 11. Em cada teste foram analisadas a coleta de três amostras. Em cada amostra foram transmitidas 1.000 mensagens e foram analisadas a perda de pacotes e a força do sinal de transmissão por meio do RSSI. A força do sinal analisado é diferente de qualidade, que por sua vez é medido pelo LQI.

- **Interferência Wi-Fi**

O primeiro teste realizado analisou a possível interferência causada pela rede Wi-Fi. O nó sensor ficou a uma distância de 3,78 metros (comprimento médio de um carro popular) do nó central. Em seguida, foram aplicadas ao mesmo ambiente as distâncias entre os nós de 4,04 metros (referente a uma Minivan), 4,57 metros (referente ao comprimento de um carro do tipo SUV), 4,84 metros (comprimento de um veículo esportivo) e, por último, a distância de 11,10 metros (comprimento de um ônibus). A tabela 12 mostra a média obtida em cada uma dessas distâncias.

Tabela 12 – Coleta de dados coexistência Wi-Fi

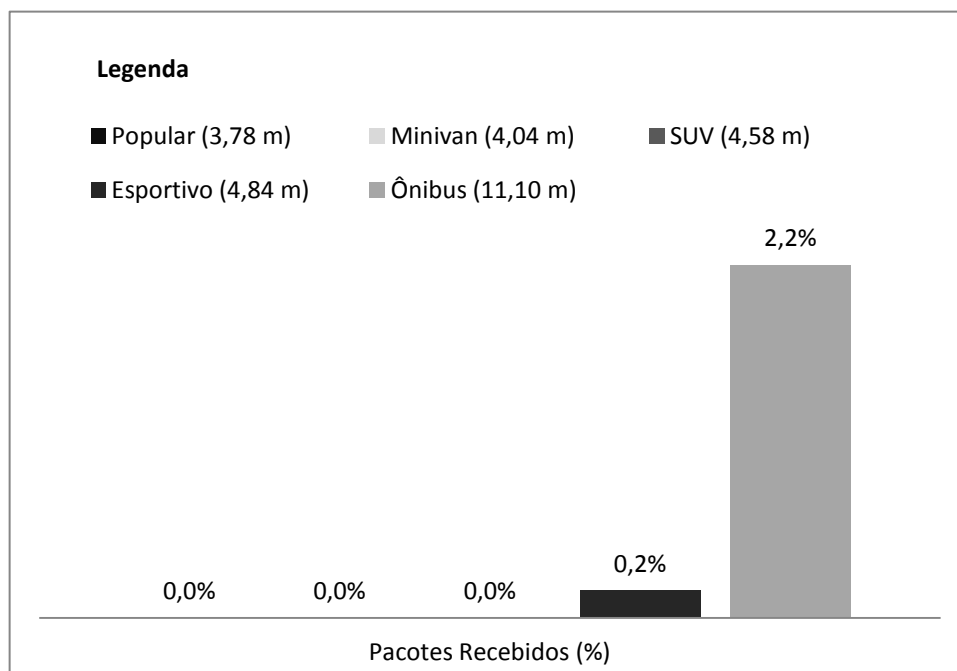
Distância (m)	Pacotes Perdidos (%)	Força do Sinal (%)
3,78	0,0	22,1
4,04	0,0	21,5
4,57	0,0	21,5
4,84	0,2	18,0
11,10	2,2	14,3

Com a distância de 3,78 metros não houve perda de pacotes. O bom desempenho aqui apresentado é consequência do canal 26 escolhido para a

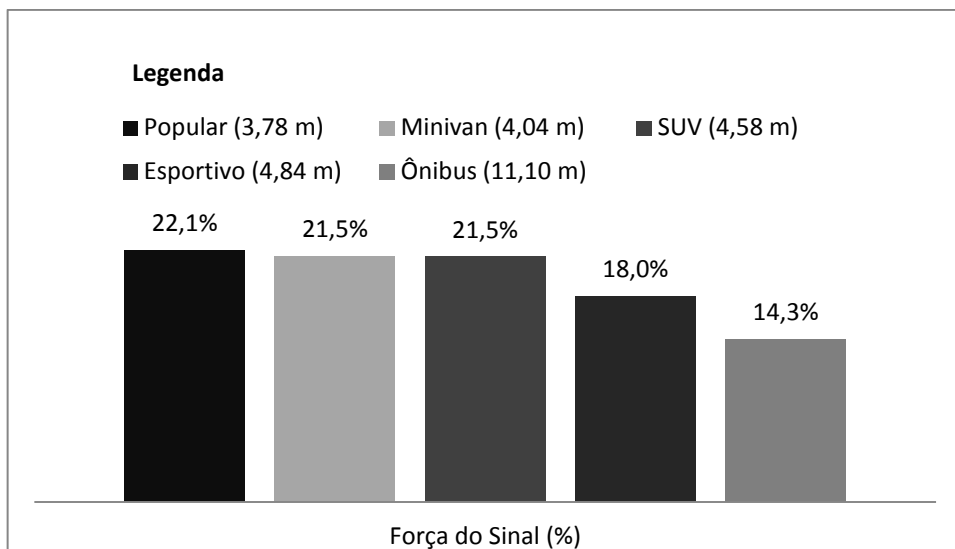
atividade de transmissão entre os nós sensores. Este canal está fora da faixa de frequência do padrão IEEE 802.11. A força do sinal recebido pelo nó sensor ficou, em média, em 22,1%. Este resultado foi obtido devido a distância entre os nós, pois quanto maior a distância entre eles, menor será a força do sinal RSSI recebido pelo nó central. A variação pode ocorrer devido a presença de obstáculos ou pessoas.

Com a medida de 4,04 metros e 4,57 metros, Como o aumento da distância entre os nós foi pequeno, os resultados obtidos são bastante semelhantes. A força do sinal variou pouco. Já com o comprimento de 4,84 metros, deu para notar pequena queda de desempenho, mas o valor de 0,2% é insignificante, podendo até mesmo ser considerado 0.

Já com a distância de 11,10 metros, a taxa de pacotes perdidos chegou a 2.2%. Este fato ocorre devido ao aumento da distância entre os nós que foi maior que as anteriores provocando também a queda da força do sinal. Mesmo com a utilização do canal 26 pelos dispositivos Zigbee, o sinal Wi-Fi causou interferência. O gráfico da figura 16 a ilustra o desempenho da rede Zigbee mediante a interferência do sinal Wi-Fi, de acordo com a perda de pacotes, em todas as distâncias analisadas e o gráfico da figura 16 b mostra a potência do sinal em cada teste.



a) Pacotes perdidos – Interferência Wi-Fi.



b) Sinal RSSI – Interferência Wi-Fi.

Figura 16 – Gráficos de desempenho da rede Zigbee com a presença de rede Wi-Fi.

Os resultados apresentados nos gráficos das figuras 16 a e 16 b mostram as variações de desempenho da rede Zigbee de acordo com as distâncias entre os nós. A perda de pacotes só foi notada a partir do comprimento de 4,84 metros. Com o aumento da distância, a força do sinal recebida pelo nó central diminuiu.

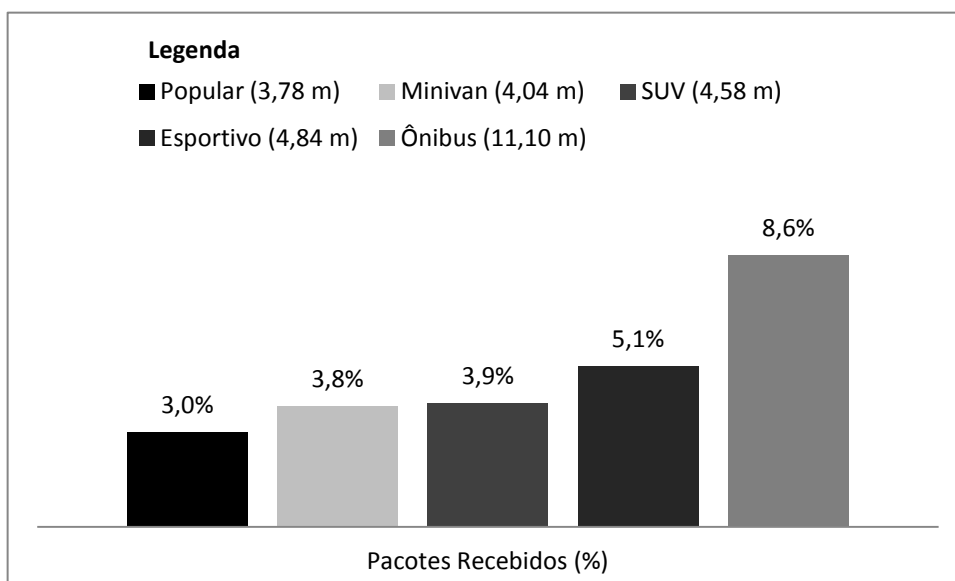
- **Interferência *Bluetooth***

A análise da interferência do *Bluetooth* nas redes Zigbee seguiu a mesma metodologia utilizada nos testes realizados com o Wi-Fi. A Tabela 13 mostra os dados obtidos na coleta de dados com os nós em todas as 5 distâncias.

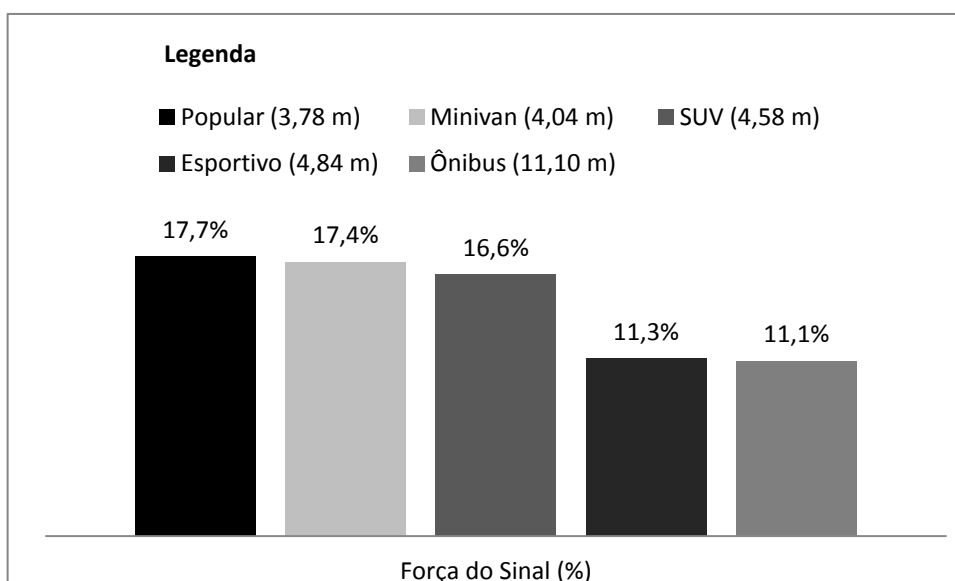
Tabela 13 – Coleta de dados coexistência *Bluetooth*

Distância (m)	Pacotes Perdidos (%)	Força do Sinal (%)
3,78	3,0	17,7
4,04	3,8	17,4
4,57	3,9	16,6
4,84	5,1	11,3
11,10	8,6	11,1

Notou-se significativa interferência do *Bluetooth* na comunicação entre os nós sensores. Isso ocorre devido às duas tecnologias trabalharem na mesma banda de frequência. Porém, a perda de pacotes poderia ser maior se não fosse a técnica de saltos de frequência implementada pelo *Bluetooth*. Os celulares utilizados no teste estavam a poucos centímetros do nó central (5 centímetros), o que pode ter ocasionado maior interferência. Os gráficos da figura 17 a e 17 b mostram os resultados obtidos neste com relação à perda de pacotes e potência do sinal recebido pelo nó, respectivamente.



a) Pacotes perdidos com interferência *Bluetooth*.



b) Sinal RSSI – Interferência *Bluetooth*.

Figura 17 – Gráficos de desempenho da rede Zigbee com a presença de rede *Bluetooth*.

Os gráficos a cima mostram como a quantidade de pacotes perdidos aumentou proporcionalmente ao aumento da distância (figura 17 a) enquanto que a potência do sinal diminuiu (figura 17 b). O Zigbee mostrou-se sensível a interferência do *Bluetooth*.

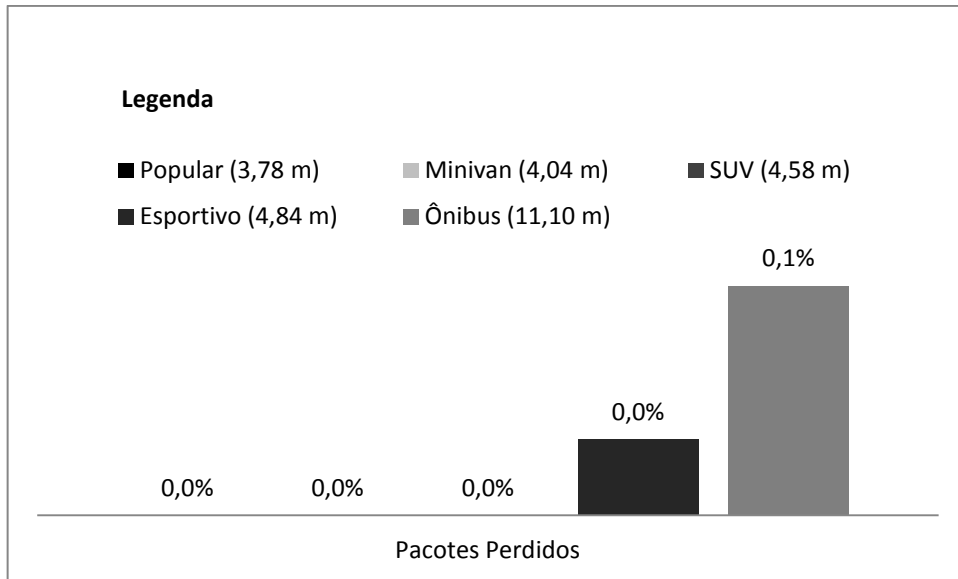
- **Interferência do Motor**

A mesma metodologia aplicada nos ambientes anteriores também foi utilizada neste teste. A tabela a seguir mostra os dados coletados.

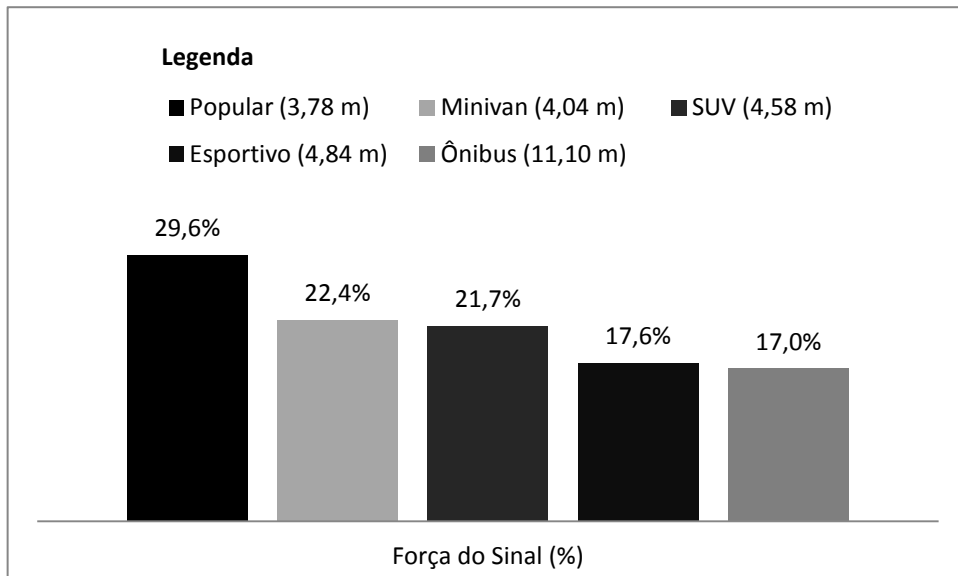
Tabela 14 – Coleta de dados coexistência com motor – interferência eletromagnética

Distância (m)	Pacotes Perdidos (%)	Força do Sinal (%)
3,78	0,0	29,6
4,04	0,0	22,4
4,57	0,0	21,7
4,84	0,0	17,6
11,10	0,1	17,0

Os resultados aqui obtidos mostram que a possível interferência eletromagnética causada pelo motor utilizado não é significativa. As perdas de pacotes em todas as amostras e em todas as medidas foram praticamente nulas. Os valores apresentados na tabela podem ser visualizados graficamente nas figuras 18 a (pacotes perdidos) e 18 b (força do sinal).



a) Pacotes perdidos – Interferência do Motor.



b) Sinal RSSI – Interferência do Motor.

Figura 18 – Gráficos de desempenho da rede Zigbee com a presença de um motor.

Analisando o gráfico da figura 18 a, percebe-se que praticamente em todos os casos o número de pacotes perdidos foi praticamente 0%. Também é percebido que nos testes com a distância entre os nós de 4,84 metros e 11,10 metros houve perda de pacotes, mas este valor pode ser desconsiderado. Na figura 18 b, nota-se que a força do sinal recebido também diminuiu de acordo com o aumento da distância entre os nós.

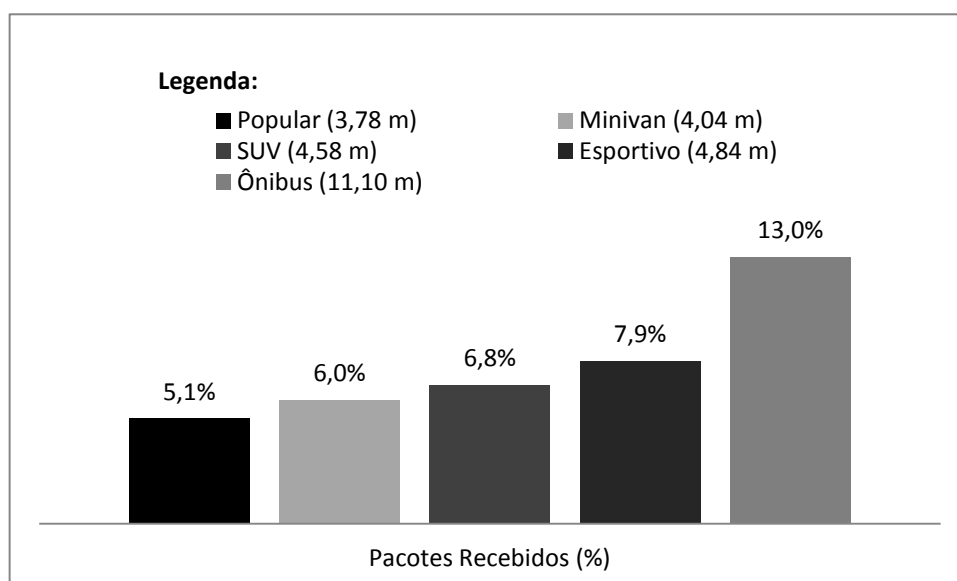
- **Interferência Wi-Fi, Bluetooth e Motor**

Para finalizar a bateria de testes, foi analisado como a rede Zigbee se comporta com a coexistência com redes *Bluetooth* e Wi-Fi e o motor. Novamente foi aplicada a mesma metodologia dos testes anteriores. A tabela a seguir mostra os resultados obtidos.

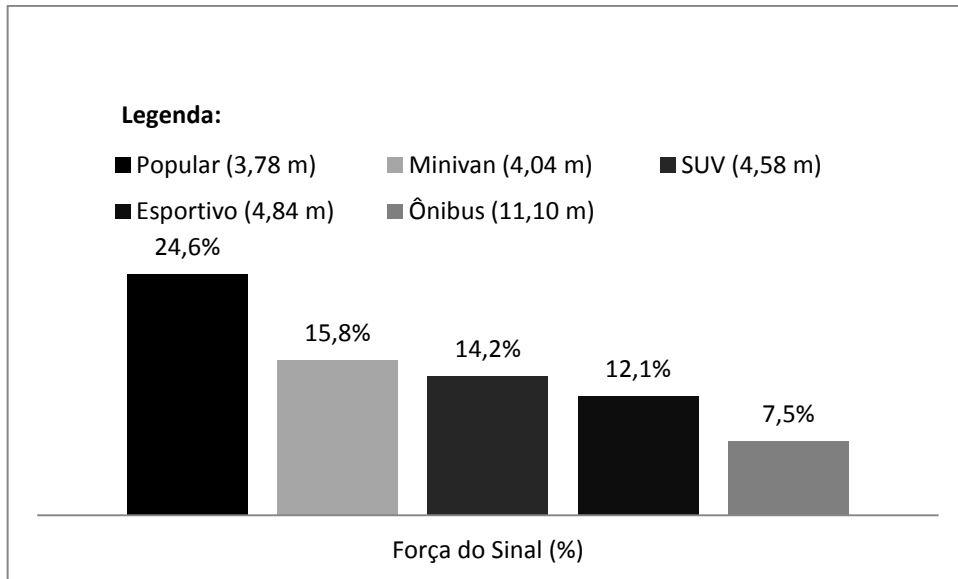
Tabela 15 – Coleta de dados coexistência Wi-Fi, *Bluetooth* e motor

Distância (m)	Pacotes Perdidos (%)	Força do Sinal (%)
3,78	5,1	24,6
4,04	6,0	15,8
4,57	6,8	14,2
4,84	7,9	12,1
11,10	13,0	7,5

Analisando a tabela, nota-se que em um ambiente onde existam redes Wi-Fi e *Bluetooth* e a presença de um motor, pode causar interferência significativa nas redes Zigbee. A perda de pacotes chegou a 13% com uma distância entre os nós de 11.10%. O nível do sinal também foi bastante afetado comparado aos resultados dos demais ambientes. Os gráficos da figura 19a e 19b ilustram o resultado obtido neste ambiente.



a) Pacotes perdidos – Interferência Wi-Fi, Bluetooth e Motor.



b) Sinal RSSI – Interferência Wi-Fi, Bluetooth e Motor.

Figura 19 – Gráficos de desempenho da rede Zigbee com coexistência *Wi-Fi*, *Bluetooth* e motor.

Ambiente em que houve maior perda de pacotes (figura 19a) e que obteve as piores potências de sinal (figura 19b). Esse resultado era esperado devido à exposição desse ambiente a todas as interferências já testadas anteriormente.

Os testes realizados mostraram que o Zigbee possui ótimo desempenho principalmente para veículos menores e em ambientes onde não haja a coexistência com o *Bluetooth*. Porém, o seu desempenho em ambientes com a presença dessa tecnologia ficou dentro do aceitável, já que a aplicação não possui nenhuma prevenção a perda de pacotes. Inicialmente pretendia-se incluir nos testes outras configurações de potência, mas os resultados não foram obtidos, pois a transmissão dos pacotes necessários para a coleta de informações não foi concluída devido ao não recebimento dos pacotes por parte do nó central.

6. Conclusão

Este trabalho consistiu no estudo do protocolo Zigbee utilizado em RSSF aplicado no ambiente intraveicular, visando a análise de características como a perda de pacotes e potência do sinal recebido. Por meio desse estudo, constatou-se que é possível substituir o meio de comunicação utilizado atualmente nas redes intraveiculares por uma rede sem fio adotando o protocolo Zigbee. Com a adoção dessa tecnologia, os custos de produção e a dificuldade de instalação de novos dispositivos em locais de difícil acesso podem ser reduzidos já que a mesma não utiliza mais fiação como meio de comunicação.

Foi realizado, também, um estudo sobre as tecnologias adotadas na indústria e pode ser constatado que o protocolo de comunicação Zigbee, baseado no padrão 802.15.4, foi o que mais se adequou às necessidades do ambiente dependendo da classe de protocolos veiculares em que este seja inserido (classes que não exijam altas taxas de transferência de dados). De acordo a análise das taxas de dados, este protocolo pode ser destinado às classes A, B, Diagnóstico e Safety Bus.

Uma das características que mais se destacam no Zigbee é a capacidade de suportar uma rede com mais de 65.000 dispositivos e a sua aplicação em RSSF. Já o *Bluetooth*, apesar de possuir taxa de transferência de dados superiores ao Zigbee e de poder ter o número de dispositivos estendidos por meio do Scatternet, esta técnica não é padrão do protocolo e a mesma pode causar interoperabilidade entre as entidades do *Bluetooth*, sendo assim mais viável o uso do Zigbee. Outra característica que influencia na escolha é o custo.

Apesar de o ambiente veicular não ser considerado limitado energeticamente, pesquisas sobre veículos que utilizam fontes de energias renováveis, principalmente energia elétrica, estão sendo realizadas e o Zigbee pode se destacar pelo seu baixo consumo de energia.

Os testes mostraram que o Zigbee obteve bom desempenho em distâncias menores. Em distâncias maiores, a perda de pacotes chegou a 13%, sendo necessária a implantação de algum método ou arquitetura que reduza esse valor para a utilização do protocolo em um número maior de aplicações.

6.1. Trabalhos Futuros

Como proposta para trabalhos pode-se implementar a identificação e reenvio de pacotes perdidos diminuindo assim a probabilidade de erros da aplicação. Analisar a possível interferência de outras redes, Inserir mais dispositivos na rede e testar como a rede se comporta ao estar configurada em outras topologias. Incluir novas aplicações ao ambiente veicular como medição de temperatura do motor, temperatura interna do veículo, sensor de estacionamento, etc.

Após a análise de testes realizados em laboratório, realizar testes no ambiente veicular, instalando a rede em um veículo observando não só características como potência do sinal e perda de pacotes, mas também a qualidade e latência do sinal com o carro em movimento.

Referências Bibliográficas

ATMEL. **Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications - AT86RF230**. [S.l.], p. 98. 2009.

ATMEL. **4-megabit 2.5-volt Only Serial DataFlash - AT45DB041**. [S.l.], p. 26. 2001.

AHMED, M.; SARAYDAR, C. U.; YIN, J. E.; TALTY, T.; AMES, M. **Intra-vehicular Wireless Networks**. Conferência IEEE Global Telecommunications (Globecom '07), Washington, DC, E.U.A., Novembro de 2007.

AKYILDIZ, Ian F.; SU, Weilian; SANKARASHBRAMANIAM, Yogesh; CAYIRCI, Erdal. **A Survey on Sensor networks**. IEEE Communications Magazine. 2002. p. 102-114.

BARONTI, P.; PILLAI, P.; CHOOK, V. W.C.; CHESSA, S.; GOTTA, A. and HU Y. F. **Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards**, Computer Communications, Volume 30, Issue 7, 26 May, Pages 1655-1695, 2007.

BERNARDO, L. **Introdução ao desenvolvimento de aplicações em TinyOS 2.x** – Universidade Nova de Lisboa, 2011.

CAMPOS, C. F. P. S. **Estudo e especificação de um sistema de instrumentação para unidades de elevação de petróleo utilizando tecnologia sem fio**. 2006. 77 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal.

CHEN, Y.; CHEN, L. **Using Bluetooth Wireless Technology in Vehicles**. Vehicular Electronics and Safety, IEEE International Conference, 2005, p. 344-347.

CHEN, L.; SUN, T.; GERLA, M. **Modeling Channel Conflict Probabilities Between IEEE 802.15 Based Wireless Personal Area Networks"** IEEE International Conference On Communications 1, Junho, 2006, p. 343-348.

CROSSBOW. **MPR-MIB Users Manual**. [S.l.], p. 64. 2007. (7430-0021-08).

CROSSBOW. **MTS/MDA Sensor Board Users Manual**. [S.l.], p. 44. 2007. (7430-0020-05).

EICHLER, S. **Performance Evaluation of the IEEE 802.11p WAVE Communication Standard**. Vehicular Technology Conference, 2007.

ELBATT, T.; SARAYDAR, C.; AMES, M.; TALTY, T. **Potential for Intra-Vehicle Wireless Automotive Sensor Networks**. Sarnoff Symposium, 2006.

ERICSSON Technology. **Scatternet - Part 1 Baseband vs. Host Stack Implementation Licensing**. June, 2004.

FAEZIPOUR, M.; NOURANI, M.; SAEED, A.; ADDEPALLI, S. **Progress and Challenges in Intelligent Vehicle Area networks**. Communications of the ACM. 2012, p. 90-100.

GUIMARÃES, A. A.; SARAIVA, A. M. **O protocolo CAN: entendendo e implementando uma rede de comunicação serial de dados baseada no barramento Controller Area Network**. In: 11 Congresso SAE Brasil, 2002, São Paulo. artigo SAE 2002-01-3569. São Paulo: SAE Brasil, 2002.

IEEE 802.15.1 **Standard Report: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Personal Area Networks (WPANs)**. The Institute of Electrical and Electronics Engineers, Inc. New York, EUA. Junho, 2002.

IEEE 802.15.4 **Standard Report: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)**. The Institute of Electrical and Electronics Engineers, Inc. New York, EUA. Outubro, 2003.

JUNIOR, J. A. V. **Aplicação das Tecnologias Sem Fio nas Redes Intraveiculares**. 2010. 66 f. Monografia (Graduação em Ciência da Computação) – Faculdades Integrada Pedro Leopoldo, Pedro Leopoldo, Minas Gerais.

KARL, H.; WILLIG, A. **Protocols and Architectures for Wireless Sensor networks**. West Sussex: John Wiley & Sons Ltd, 2006.

LOUREIRO, Antonio Alfredo Ferreira; NOGUEIRA, José Marcos S; RUIZ, Linnyer Beatryz; MINI, Raquel Aparecida de Freitas. **Redes de Sensores Sem Fio**. In: 21º

Simpósio Brasileiro de Redes de Computadores. (Org.). Minicurso do SBRC. :, 2003, v. , p. -.

MEMSIC. **IRIS OEM Edition Hardware USER MANUAL**. MEMSIC, Inc. [S.l.], p. 36. (7430-0549-01).

OLIVEIRA, F. D. M. **Desenvolvimento de um Software de Comunicação sem Fio Aplicado à Instrumentação de Unidade de Elevação de Petróleo Tipo Plunger Lift**. 2009. 143 f. Dissertação (Mestrado em Engenharia Elétrica e de Computação) – Universidade Federal do Rio Grande do Norte, Natal.

PETERSEN, Stig; DOYLE, Paula; VATLAND, Sven; AASLAND, Christian Salbu; ANDERSEN, Trod Michael; SLONG, Dag. **Requirements, Drivers and Analysis of Wireless Sensor Network Solutions for the Oil & Gas Industry**. Institute of Electrical and Electronic Engineers, 2007.

RIBEIRO, M. P. N. **Introduzindo Tecnologia Sem-fio em Redes Intraveicular**. Artigo, Universidade Federal da Bahia, Campus Universitário de Ondina, Ondina, Salvador, BA, 2008.

ROCA, I. **Rede de Sensores Sem Fio em Ambientes Veiculares Baseada no Padrão Zigbee**. 2008. 136 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade de São Paulo, São Paulo.

SILVA, L. H. S. **Desenvolvimento de uma Rede de Sensores Sem Fio Utilizando ZigBee para Aplicações Diversas**. 2011. 80 f. Monografia (Graduação em Engenharia da Computação) – Universidade de Pernambuco, Recife.

SILVA, R. C. **Redes de sensores sem fio**. 2006. 54 f. Monografia (Graduação em Sistema de Informação) – Universidade Estadual de Monte Carlos, Monte Carlos.

SOUZA, R. V. **CAN (Controller Area Network): Uma Abordagem para Automação e Controle na Área Agrícola**. 2002. 94 f. Dissertação (Mestrado em Engenharia Mecânica) – Universidade de São Paulo, São Carlos, SP.

TSAI, Hsin-Mu; TONGUZ, O. K.; SARAYDAR, C.; TALTY, T.; AMES, M.; MACDONALD, A. **Zigbee-Based Intra-Car Wireless Sensor Networks: A Case Study**. IEEE Wireless Communications, 2007.

WOLF, D. F.; OSÓRIO, F. S.; FERNANDES, L. C.; SHINZATO, P. **Veículos Inteligentes: Conceitos, Tecnologias e Aplicações Práticas**. 1ª Conferência Brasileira de Sistemas Embarcados Críticos. Minicurso do CBSEC. 2011.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. **Wireless sensor network survey**, California, 2008.

ZHAO, F.; GUIBAS, L. **Wireless Sensor networks: An Information Processing Approach**. São Francisco: Elsevier, 2004.

ZIGBEE Specification. Documento técnico 053474r17. Zigbee Standards Organization, 2008.

Apêndice A – Códigos das Aplicações

Nó Sensor

- **LuminosidadeC.nc**

```
#include "Luminosidade.h"
#include <Timer.h>

module LuminosidadeC{
  uses{
    interface Boot;
    interface Leds;
    interface SplitControl as AMControl;
    interface Read<uint16_t> as luz;
    interface Timer<TMilli> as Timer;
    interface Packet;
    interface AMPacket;
    interface AMSend;
    interface Receive;
  }
}

implementation {
  bool ocupado = FALSE;
  uint16_t estado_atual=1,estado_anterior=0;
  message_t pkt;
  event void Boot.booted(){
    call AMControl.start();
  }

  event void AMControl.startDone(error_t err){
    if(err == SUCCESS){
      call Timer.startPeriodic(TIMER_PERIOD_MILLI);
    }else{
      call AMControl.start();
    }
  }

  event void Timer.fired(){
    call luz.read();
  }
}
```

```

event void luz.readDone(error_t ok, uint16_t val){
    if(ok == SUCCESS){
        if(val < VALUE_DARK){
            call Leds.led0On();
            estado_atual = 1;
        }else{
            call Leds.led0Off();
            estado_atual = 2;
        }
        if(estado_atual != estado_anterior){
            if(!ocupado){
                LuminosidadeMsg* Lpkt = (LuminosidadeMsg*)(call
Packet.getPayload(&pkt, sizeof(LuminosidadeMsg)));
                if(Lpkt == NULL)return;
                Lpkt->nodeid = TOS_NODE_ID;
                Lpkt->data = estado_atual;
                if(call AMSend.send(AM_BROADCAST_ADDR,
&pkt, sizeof(LuminosidadeMsg))==SUCCESS){
                    ocupado = TRUE;
                    call Leds.led2On();
                }
            }
        }
        estado_anterior = estado_atual;
    }
}

event void AMSend.sendDone(message_t* msg, error_t err){
    if(&pkt == msg){
        ocupado = FALSE;
        call Leds.led2Off();
    }
}

event message_t* Receive.receive(message_t* msg, void* payload, uint8_t
len){
    return NULL;
}

event void AMControl.stopDone(error_t err){
}

```

- **LuminosidadeAppC.nc**

```

#include "Luminosidade.h"
#include <Timer.h>

configuration LuminosidadeAppC {

```

```

}

implementation {

    components MainC;
    components LedsC;
    components new TimerMilliC() as Timer;
    components LuminosidadeC as app;
    components new PhotoC() as luz;
    components new AMSenderC(AM_LUMINOSIDADEMSG);
    components new AMReceiverC(AM_LUMINOSIDADEMSG);
    components ActiveMessageC as AM;

    app.Boot -> MainC;
    app.Leds -> LedsC;
    app.Timer -> Timer;
    app.Packet -> AMSenderC;
    app.AMPacket -> AMSenderC;
    app.AMControl -> AM;
    app.AMSend -> AMSenderC;
    app.Receive -> AMReceiverC;
    app.luz -> luz;
}

```

Mensagem

- **Luminosidade.h**

```

#ifndef LUMINOSIDADE_H
#define LUMINOSIDADE_H

enum {TIMER_PERIOD_MILLI = 250,
VALUE_DARK = 700,
AM_LUMINOSIDADEMSG = 6};

typedef nx_struct LuminosidadeMsg{
    nx_uint16_t nodeid;
    nx_uint16_t data;
}LuminosidadeMsg;
#endif

```

Makefile

- **Makefile**

```
SENSORBOARD=mda100
```

```
BUILD_EXTRA_DEPS=LuminosidadeMsg.class
```

```
CFLAGS += -DRF230_DEF_CHANNEL=26
```

```
PFLAGS += -DRF230_DEF_RFPOWER=0
```

```
LuminosidadeMsg.class: LuminosidadeMsg.java  
    javac LuminosidadeMsg.java
```

```
LuminosidadeMsg.java:
```

```
    mig java -target=iris -java-classname=LuminosidadeMsg Luminosidade.h  
LuminosidadeMsg -o $@
```

```
COMPONENT=LuminosidadeAppC  
include $(MAKERULES)
```

Nó Central

A aplicação utilizada no nó central foi a *BaseStation*. Esta aplicação é encontrada no diretório `/tinys-2.1.1/apps/BaseStation`.

Aplicação Central

- **Coletor.java**

```
import net.tinys.message.Message;  
import net.tinys.message.MessageListener;  
import net.tinys.message.MotIF;  
import net.tinys.packet.BuildSource;  
import net.tinys.packet.PhoenixSource;  
  
public class Coletor implements MessageListener{  
  
    private MotIF mote;  
  
    @Override  
    public void messageReceived(int to, Message message) {  
        LuminosidadeMsg msg = (LuminosidadeMsg) message;  
        //int source = message.getSerialPacket().get_header_src();  
  
        if (msg.get_data() == 1) {  
            System.out.println("Ambiente Escuro. Acender Faróis!");  
        } else {  
            System.out.println("Ambiente Claro. Apagar Faróis!");  
        }  
    }  
}
```

```

public Coletor(MotelF mote){
    this.mote = mote;
    this.mote.registerListener(new LuminosidadeMsg(), this);
}
public static void main(String[] args) {
    PhoenixSource phoenix;
    phoenix = BuildSource.makePhoenix(null);
    MotelF m = new MotelF(phoenix);
    Coletor coletor = new Coletor(m);

}
}

```

- **LuminosidadeMsg.java**

```

/**
 * This class is automatically generated by mig. DO NOT EDIT THIS FILE.
 * This class implements a Java interface to the 'LuminosidadeMsg'
 * message type.
 */

public class LuminosidadeMsg extends net.tinyos.message.Message {

    /** The default size of this message type in bytes. */
    public static final int DEFAULT_MESSAGE_SIZE = 4;

    /** The Active Message type associated with this message. */
    public static final int AM_TYPE = 6;

    /** Create a new LuminosidadeMsg of size 4. */
    public LuminosidadeMsg() {
        super(DEFAULT_MESSAGE_SIZE);
        amTypeSet(AM_TYPE);
    }

    /** Create a new LuminosidadeMsg of the given data_length. */
    public LuminosidadeMsg(int data_length) {
        super(data_length);
        amTypeSet(AM_TYPE);
    }

    /**
     * Create a new LuminosidadeMsg with the given data_length
     * and base offset.
     */
    public LuminosidadeMsg(int data_length, int base_offset) {
        super(data_length, base_offset);
        amTypeSet(AM_TYPE);
    }
}

```

```

}

/**
 * Create a new LuminosidadeMsg using the given byte array
 * as backing store.
 */
public LuminosidadeMsg(byte[] data) {
    super(data);
    amTypeSet(AM_TYPE);
}

/**
 * Create a new LuminosidadeMsg using the given byte array
 * as backing store, with the given base offset.
 */
public LuminosidadeMsg(byte[] data, int base_offset) {
    super(data, base_offset);
    amTypeSet(AM_TYPE);
}

/**
 * Create a new LuminosidadeMsg using the given byte array
 * as backing store, with the given base offset and data length.
 */
public LuminosidadeMsg(byte[] data, int base_offset, int data_length) {
    super(data, base_offset, data_length);
    amTypeSet(AM_TYPE);
}

/**
 * Create a new LuminosidadeMsg embedded in the given message
 * at the given base offset.
 */
public LuminosidadeMsg(net.tinyos.message.Message msg, int base_offset) {
    super(msg, base_offset, DEFAULT_MESSAGE_SIZE);
    amTypeSet(AM_TYPE);
}

/**
 * Create a new LuminosidadeMsg embedded in the given message
 * at the given base offset and length.
 */
public LuminosidadeMsg(net.tinyos.message.Message msg, int base_offset,
int data_length) {
    super(msg, base_offset, data_length);
    amTypeSet(AM_TYPE);
}

/**
 * Return a String representation of this message. Includes the

```

```

* message type name and the non-indexed field values.
*/
public String toString() {
    String s = "Message <LuminosidadeMsg> \n";
    try {
        s += " [nodeid=0x"+Long.toHexString(get_nodeid())+"]\n";
    } catch (ArrayIndexOutOfBoundsException aioobe) { /* Skip field */ }
    try {
        s += " [data=0x"+Long.toHexString(get_data())+"]\n";
    } catch (ArrayIndexOutOfBoundsException aioobe) { /* Skip field */ }
    return s;
}

// Message-type-specific access methods appear below.

////////////////////////////////////
// Accessor methods for field: nodeid
// Field type: int, unsigned
// Offset (bits): 0
// Size (bits): 16
////////////////////////////////////

/**
 * Return whether the field 'nodeid' is signed (false).
 */
public static boolean isSigned_nodeid() {
    return false;
}

/**
 * Return whether the field 'nodeid' is an array (false).
 */
public static boolean isArray_nodeid() {
    return false;
}

/**
 * Return the offset (in bytes) of the field 'nodeid'
 */
public static int offset_nodeid() {
    return (0 / 8);
}

/**
 * Return the offset (in bits) of the field 'nodeid'
 */
public static int offsetBits_nodeid() {
    return 0;
}

```

```

/**
 * Return the value (as a int) of the field 'nodeid'
 */
public int get_nodeid() {
    return (int)getUIntBEElement(offsetBits_nodeid(), 16);
}

/**
 * Set the value of the field 'nodeid'
 */
public void set_nodeid(int value) {
    setUIntBEElement(offsetBits_nodeid(), 16, value);
}

/**
 * Return the size, in bytes, of the field 'nodeid'
 */
public static int size_nodeid() {
    return (16 / 8);
}

/**
 * Return the size, in bits, of the field 'nodeid'
 */
public static int sizeBits_nodeid() {
    return 16;
}

////////////////////////////////////
// Accessor methods for field: data
// Field type: int, unsigned
// Offset (bits): 16
// Size (bits): 16
////////////////////////////////////

/**
 * Return whether the field 'data' is signed (false).
 */
public static boolean isSigned_data() {
    return false;
}

/**
 * Return whether the field 'data' is an array (false).
 */
public static boolean isArray_data() {
    return false;
}

/**

```



```

    * Return the offset (in bytes) of the field 'data'
    */
    public static int offset_data() {
        return (16 / 8);
    }

    /**
     * Return the offset (in bits) of the field 'data'
     */
    public static int offsetBits_data() {
        return 16;
    }

    /**
     * Return the value (as a int) of the field 'data'
     */
    public int get_data() {
        return (int)getUIntBEElement(offsetBits_data(), 16);
    }

    /**
     * Set the value of the field 'data'
     */
    public void set_data(int value) {
        setUIntBEElement(offsetBits_data(), 16, value);
    }

    /**
     * Return the size, in bytes, of the field 'data'
     */
    public static int size_data() {
        return (16 / 8);
    }

    /**
     * Return the size, in bits, of the field 'data'
     */
    public static int sizeBits_data() {
        return 16;
    }
}

```

Apêndice B – Imagens do Ambiente de Testes

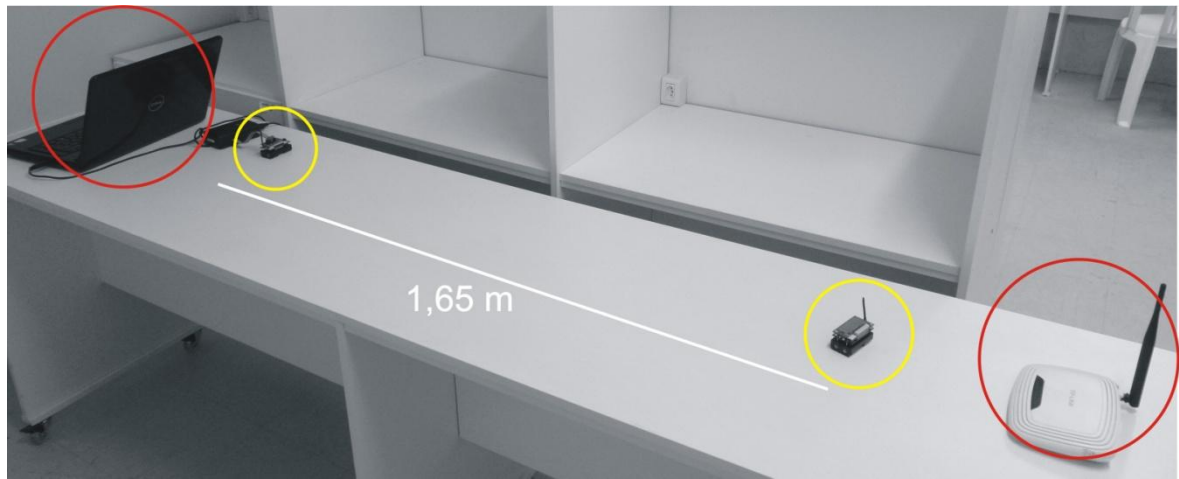


Figura 20 – Ambiente de teste de coexistência Wi-Fi. Dispositivos Zigbee com círculos amarelos e dispositivos Wi-Fi com círculos vermelhos.

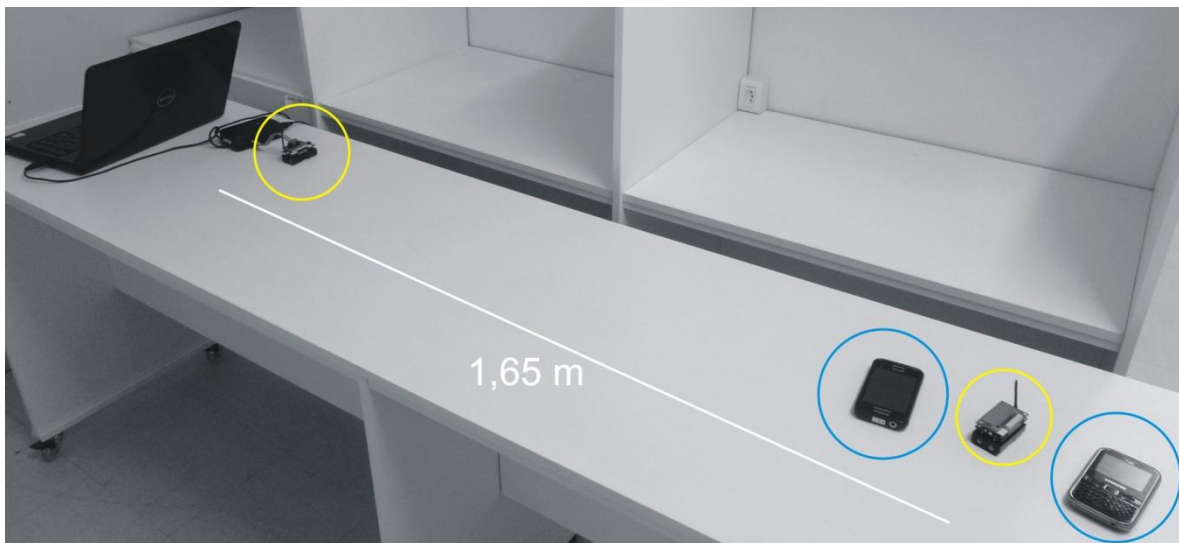


Figura 211 – Ambiente de teste de coexistência Bluetooth. Dispositivos Zigbee com círculos amarelos e dispositivos Bluetooth com círculos Azuis.

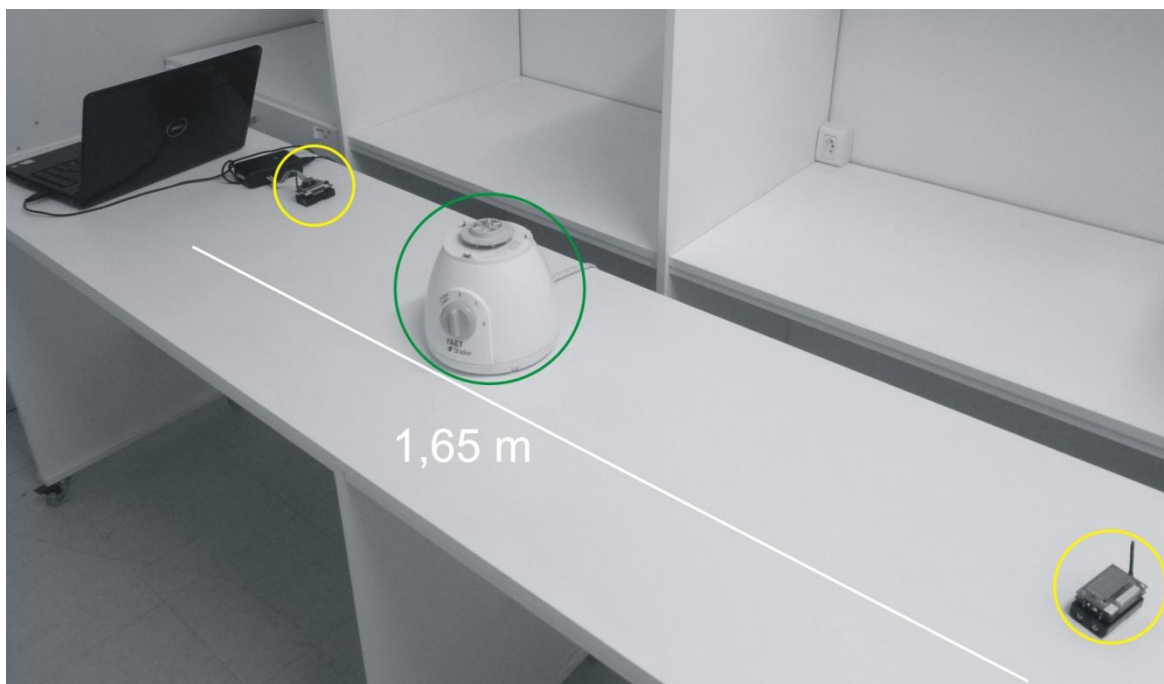


Figura 222 – Ambiente de teste de interferência eletromagnética.
Dispositivos Zigbee com círculos amarelos e motor com círculo verde.

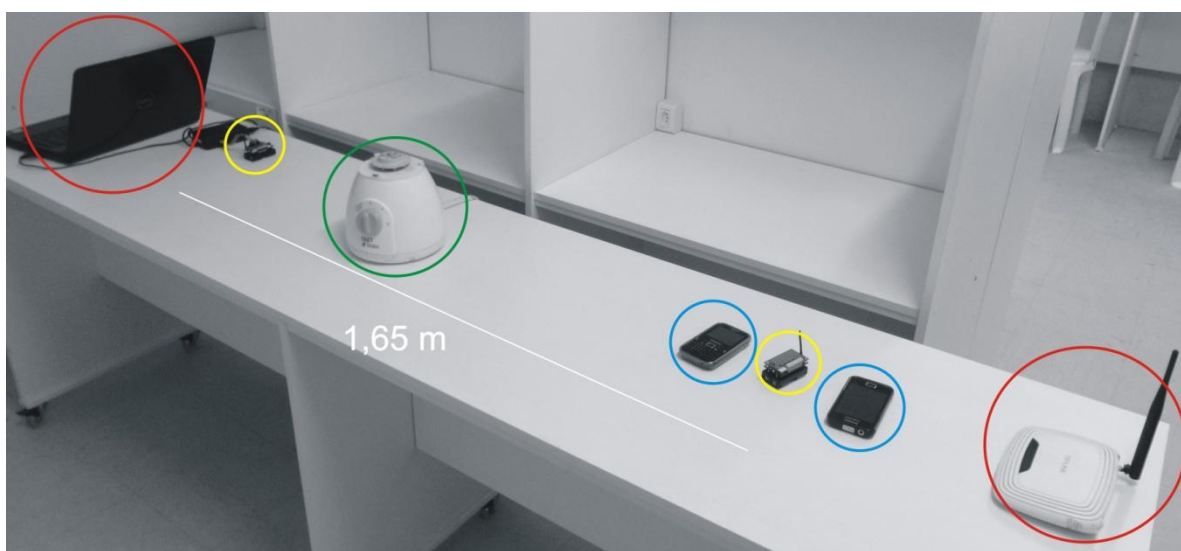


Figura 233 – Ambiente de teste de coexistência Wi-Fi, Bluetooth e eletromagnética causada por motor. Dispositivos Zigbee com círculos amarelos, dispositivos Wi-Fi com círculos vermelhos, dispositivos Bluetooth com círculos azuis e motor com círculo verde.