



ANTONIO RAFAEL SANT'ANA

**SISTEMA DE DETECÇÃO DE COLISÃO
ENTRE VEÍCULOS UTILIZANDO GPS E
ZIGBEE**

LAVRAS - MG

2014

ANTONIO RAFAEL SANT'ANA

**SISTEMA DE DETECÇÃO DE COLISÃO ENTRE VEÍCULOS
UTILIZANDO GPS E ZIGBEE**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Ciência da Computação, para a obtenção do título de Mestre.

Orientador

Dr. Wilian Soares Lacerda

LAVRAS - MG

2014

**Ficha Catalográfica Elaborada pela Coordenadoria de Produtos e
Serviços da Biblioteca Universitária da UFLA**

Sant'Ana, Antonio Rafael.

Sistema de detecção de colisão entre veículos utilizando GPS e
ZigBee / Antonio Rafael Sant'Ana. – Lavras : UFLA, 2014.
75 p. : il.

Dissertação (mestrado) – Universidade Federal de Lavras, 2014.
Orientador: Wilian Soares Lacerda.
Bibliografia.

1. VANET. 2. ADAS. 3. Rede veicular. 4. Acidentes. 5.
Trânsito. I. Universidade Federal de Lavras. II. Título.

CDD – 004.6

ANTONIO RAFAEL SANT'ANA

**SISTEMA DE DETECÇÃO DE COLISÃO ENTRE VEÍCULOS
UTILIZANDO GPS E ZIGBEE**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Ciência da Computação, para a obtenção do título de Mestre.

APROVADA em 26 de fevereiro de 2014.

Dr. Luiz Henrique Andrade Correia UFPA

Dr. Ricardo Augusto Rabelo Oliveira UFOP

Dr. Wilian Soares Lacerda

Orientador

LAVRAS - MG

2014

*Dedico esta dissertação à minha família, que me apoiou plenamente
durante toda esta jornada.*

AGRADECIMENTOS

Ao meu orientador, professor Wilian Soares Lacerda, pelas dicas, conselhos e a plena disponibilidade.

Ao meu amigo, Erasmo Evangelista de Oliveira pelo incentivo constante me motivou a prosseguir com o projeto.

À minha amiga e colega de trabalho, Cássia Marques Batista Nobre, pela grande ajuda nas correções e dicas de melhoria em meu projeto.

Ao meu irmão, Alan Filipe Sant'Ana, pela grande ajuda na execução e construção dos experimentos deste trabalho.

Aos meus amigos, Douglas Garcia Botelho, Luiz Augusto, Mairo de Oliveira e Flávia Reis pelo auxílio na execução dos experimentos deste trabalho.

Aos meus pais que me acolheram em um período de dificuldade e me possibilitaram concluir o projeto.

A todos os meus colegas do PPGCC da UFLA pela ajuda durante as disciplinas do mestrado.

A todos os professores do PPGCC da UFLA pelas instruções e conhecimentos passados.

A secretaria do PPGCC da UFLA pela pronta disponibilidade.

RESUMO

Vehicle Ad Hoc Networks (VANETs) são redes *ad hoc* veiculares nas quais os nós são veículos automotores que apresentam alta mobilidade e podem se comunicar uns com os outros (*V2V - Vehicle-to-Vehicle*) ou com uma infra-estrutura de estrada (*V2I - Vehicle-to-Infrastructure*). Em decorrência do grande número de acidentes no trânsito, as principais aplicações desenvolvidas para VANETs estão relacionadas à segurança dos motoristas. Embora existam vários projetos de pesquisa, voltados à definição de padrões e desenvolvimento de hardwares específicos, para o funcionamento das aplicações em redes veiculares, ainda, existe uma indisponibilidade dessa tecnologia de comunicação de maneira funcional. Objetivou-se neste trabalho avaliar a utilização do padrão ZigBee, em redes veiculares, para o desenvolvimento de aplicações, voltadas à prevenção de acidentes no trânsito. Um protótipo foi desenvolvido, para que fosse possível avaliar a viabilidade de um sistema, para alerta de colisão entre veículos, utilizando GPS e ZigBee. Nos resultados mostraram-se que o ZigBee atende os requisitos mínimos necessários para a construção de aplicações, voltadas à segurança no trânsito, podendo ser utilizado como uma alternativa ao padrão IEEE 802.11p.

Palavras-chave: ZigBee; VANET; Vehicular Ad Hoc Networks

ABSTRACT

Vehicular Ad-Hoc Networks (VANETs) are networks that use vehicles with high mobility and can communicate to others vehicles V2V (Vehicle-to-vehicle) and/or to a road structure V2I (Vehicle-to-infrastructure) as its nodes. Due to a large number of traffic accidents, road safety has become a main issue and VANETS can be used as a tool to guarantee driver safety. The aim of this work is to evaluate the use of the standard ZeeBee in vehicular networks, developing safety relating applications in order to improve safety in road traffics. Latency tests and packet losses were performed using real testing between vehicles with different speed and distance. Furthermore, a prototype was developed so that we could assess the viability of the system to alert collisions among vehicles by using GPS and ZeeBee. The results indicate that ZigBee was efficient and has the minimum requirements needed for traffic safety related applications. ZigBee can be used as an alternative to the standard IEEE 802.11p.

Keywords: ZigBee; VANET; Vehicular Ad Hoc Networks

LISTA DE FIGURAS

Figura 1	Exemplo de uma VANET (ALVES et al., 2009).	11
Figura 2	Espectro para aplicações DSRC (ALVES et al., 2009).	19
Figura 3	Aplicações do ZigBee. Adaptado de Gislason (2008).	21
Figura 4	Arquitetura Padrão do ZigBee (BARONTI et al., 2007).	23
Figura 5	Topologias de rede ZigBee (GHAZY; OZKUL, 2009).	27
Figura 6	a) RPSMA, b) Whip, c) U.FL connector, d) Chip integrado.	30
Figura 7	Quadro do Modo API (DIGI, 2012).	31
Figura 8	Tipos de Quadros do Modo API (DIGI, 2012).	31
Figura 9	Funcionamento do A-GPS (DJUKNIC; RICHTON, 2001)	37
Figura 10	Nó para comunicação veicular.	49
Figura 11	Comunicação entre Arduino, Módulo Xbee e Dispositivo Móvel	49
Figura 12	Ambiente de teste do protótipo	57
Figura 13	Trajeto ambiente urbano denso	59
Figura 14	Trajeto ambiente urbano pouco denso	60
Figura 15	Trajeto ambiente rodoviário	61
Figura 16	Teste de Latência.	63
Figura 17	Perdas - Ambiente Urbano Denso	64
Figura 18	Perdas - Ambiente Urbano com Densidade Média	64
Figura 19	Perdas - Ambiente Rodoviário	65
Figura 20	Teste do protótipo para alerta de colisão	66
Figura 21	Teste do protótipo para alerta de colisão	66

LISTA DE TABELAS

Tabela 1	Principais características da camada física	23
Tabela 2	Tabela comparativa entre Xbee Pro S2B e WAVE.....	33

LISTA DE SIGLAS

ACK	Acknowledge
ADAS	Advanced Driver Assistance Systems
A-GPS	Assisted Global Positioning System
AOA	Angle of Arrival
AODV	Ad hoc On Demand Distance Vector
API	Application Programming Interface
APO	Application Objects
APS	Application Sub Layer
CSMA/CA	Carrie Sense Multiple Access With Collision Avoidance).
CTP	Collection Tree Protocol
DGPS	Differential GPS
DSRC	Dedicated Short-Range Communications
ED	Energy Detection
EDCA	Enhanced Distributed Channel Access
FFD	Full Function Devices
GPS	Global Positioning System
IEEE	Institute of Electric and Electronic Engineers
LQI	Link Quality Indication
MAC	Medium Access Contro
PAN	Personal Area Network
PPGCC	Pós-Graduação em Ciência da Computação
RFD	Reduced Function Devices
RSSF	Redes de Sensores sem Fio
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UART	Universal Assynchronous Receiver/Transmitter
UFLA	Universidade Federal de Lavras
UFOP	Universidade Federal de Ouro Preto
V2I	Vehicle-to-Vehicle
V2I	Vehicle-to-VehiclVehicle-to-Infrastrucuture
VANET	Vehicular Ad Hoc Networks
WAVE	Wireless Access in Vehicular Environments
ZDO	Zigbee Device Objetc

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos	13
1.2	Motivação	14
1.3	Organização	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	ADAS	16
2.2	Arquitetura WAVE	19
2.3	Padrão ZigBee	21
2.3.1	Pilha de protocolos	22
2.3.2	Módulos Xbee	28
2.4	Arduino	34
2.5	Localização em VANET	35
2.5.1	Sistema de Posicionamento Global Geográfico	35
2.5.2	Acompanhamento por Mapas (<i>Map Matching</i>)	38
2.5.3	Posição estimada (<i>Dead Reckoning</i>)	38
2.5.4	Localização por celular	39
2.5.5	Processamento de imagem e vídeo	40
2.5.6	Localização relativa ad hoc distribuída	41
2.6	Plataforma Android	42
3	Trabalhos Relacionados	45
4	METODOLOGIA	48
4.1	Arquitetura do Sistema	48
4.2	Desenvolvimento	51
4.3	Cenários	58
4.3.1	Ambiente urbano denso	59
4.3.2	Ambiente urbano pouco denso	59
4.3.3	Ambiente rodoviário	60
5	RESULTADOS E DISCUSSÕES	62
5.1	Teste de latência	62
5.2	Ambiente urbano denso	62
5.3	Ambiente urbano densidade média	63
5.4	Ambiente rodoviário	65
5.5	Protótipo	65
6	CONCLUSÃO	68
6.1	Contribuições	69
6.2	Trabalhos Futuros	69

1 INTRODUÇÃO

A Organização Mundial da Saúde (OMS) registrou em 2009 cerca de 1,2 milhões de mortes no trânsito, o que representa um total de três mil mortes por dia. O Brasil está entre os dez países que concentram mais de 60% dos óbitos por acidentes de trânsito (TOROYAN, 2009). Neste contexto, os chamados Sistemas de Transporte Inteligentes (*ITS - Intelligent Transportation Systems*) vêm sendo utilizados para diminuir o número de acidentes no trânsito. Como exemplo, temos os sistemas de monitoramento de tráfego, auxílio a cruzamento sem sinalização e os de prevenção de colisões (LI; WANG, 2007).

A maioria dos ITS são implementados, utilizando Redes *Ad Hoc* Veiculares (*VANET - Vehicular Ad Hoc Networks*), que são redes sem fio entre veículos. Nessas redes, os veículos comunicam entre si (*V2V - Vehicle-to-Vehicle*) e/ou com equipamentos de uma infra-estrutura fixa (*V2I - Vehicle-to-Infrastructure*). A Figura 1 mostra um exemplo de redes veiculares.

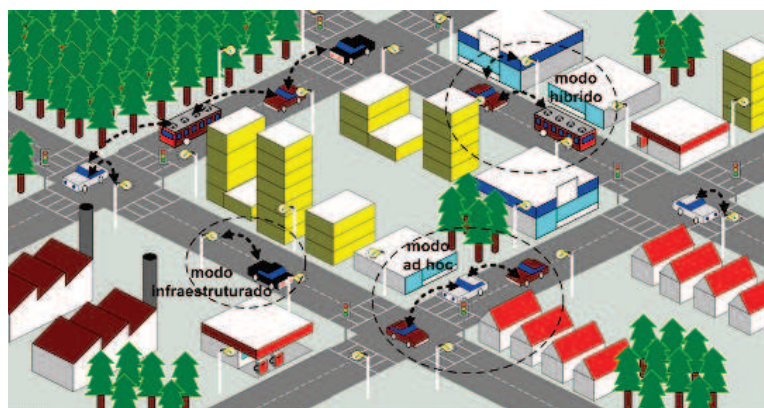


Figura 1 Exemplo de uma VANET (ALVES et al., 2009).

As comunicações V2V e V2I garantem o fornecimento de informações, considerando os arredores de um veículo, que são necessárias para o funcionamento dos Sistemas Avançados de Assistência ao Condutor (*ADAS - Advanced Driver Assistance Systems*). Esses sistemas utilizam as informações fornecidas pelos sensores locais de um veículo e de seus vizinhos para construir aplicações voltadas à segurança do condutor (ESAFETY, 2008).

A forma mais utilizada para fornecer a localização dos veículos é o uso do Sistema de Posicionamento Global (*GPS - Global Positioning System*). Receptores GPS podem ser, facilmente, encontrados nos veículos, em virtude de sua facilidade de instalação e da presença cada vez mais comum em dispositivos móveis. Embora esses receptores GPS, sozinhos, não proporcionem a precisão necessária para garantir o correto funcionamento de aplicações críticas de segurança no trânsito, as informações adquiridas podem ser combinadas com outras técnicas de localização como, por exemplo, *Dead Reckoning*, localização por celular, mapas digitais e localização por imagem e, assim, melhorar a confiabilidade dos sistemas (BOUKERCHE et al., 2008).

A troca de informações entre os veículos de uma VANET está sujeita a características peculiares como: curto tempo de contato entre os veículos, mudanças constantes de trajetória e presença de obstáculos entre os veículos. Para garantir que tais características não atrapalhem o funcionamento dos sistemas desenvolvidos para VANETs, a tecnologia de comunicação utilizada deve fornecer uma conexão confiável, minimizar atraso na entrega das informações, evitar perdas de pacotes e manter uma largura de banda suficiente para atender as necessidades das aplicações (CHENG; SHAN; ZHUANG, 2011).

No intuito de padronizar as VANETs, a FCC (*Federal Communication Commission*), órgão Norte Americano responsável pela regulamentação das comunicações no país, reservou uma faixa de 75 MHz do espectro de frequências, na faixa de 5,9 GHz, exclusivamente para aplicações em redes veiculares. Essa faixa é chamada DSRC (*Dedicated Short-Range Communications*). Além disso, o IEEE propôs o padrão IEEE 802.11p WAVE (*Wireless Access in Vehicular Environments*) para promover regras e padrões de operação destas redes. O padrão WAVE utiliza múltiplos canais de comunicação, podendo reservar uma faixa de frequência, para aplicações de segurança e outra para demais aplicações. O Padrão WAVE está em desenvolvimento pelo IEEE e é composto por documentos da família IEEE 1609 (JIANG; DELGROSSI, 2008).

Em consequência da falta de disponibilidade de equipamentos com padrão IEEE 802.11p no mercado, em muitos trabalhos é explorada a criação de VANETs, servindo-se de equipamentos pessoais, dentro dos veículos, como computadores portáteis, *tablets* e celulares, utilizando os padrões IEEE 802.11(a,b,g). Quando comparados com os equipamentos desenvolvidos para VANETs, esses dispositivos estão sujeitos a condições limitadas de comunicação. Contudo, podem ter desempenho suficiente para executar determinados tipos de aplicações (ALVES et al., 2008).

1.1 Objetivos

Este projeto foi conduzido com o objetivo principal de avaliar o padrão ZigBee quanto aos requisitos mínimos necessários para a criação de aplicações voltadas à segurança no trânsito. Foram avaliados os seguintes fatores na comunicação de dados: latência, alcance e taxa de perda dos

pacotes. O trabalho foi realizado por meio de experimentação em ambiente real.

O segundo objetivo é desenvolver um protótipo para alerta de colisão entre veículos que seja comercialmente atrativo, utilizando o padrão ZigBee para comunicação V2V/V2I e dispositivos móveis com GPS e plataforma Android.

1.2 Motivação

Muitos acidentes no trânsito ocorrem quando objetos distantes ou impedimentos rodoviários não são detectados rapidamente. Para evitar esses acidentes, são necessários sistemas de segurança de longo alcance com capacidade de detecção em tempo real. Para evitar colisões de trânsito, o conhecimento da posição de um veículo em relação aos veículos vizinhos é de suma importância para ajudar o motorista a decidir de forma segura a aceleração e desaceleração de seu veículo.

Embora exista um padrão de comunicação próprio para redes veiculares (IEEE 802.11p), existe uma falta de disponibilidade dessa tecnologia de comunicação de uma maneira funcional. Com isso, a utilização do padrão ZigBee, para transmissão de dados em redes V2V e V2I, é uma opção atrativa, dado o seu baixo custo e à facilidade de aquisição.

Mesmo se assumirmos que, do ponto de vista técnico, as soluções tecnológicas, baseadas em comunicação V2V, estejam saturadas, o principal desafio, para estas soluções, ainda, é que o nível de confiança, que lhes foi atribuído, seja proporcional à taxa de penetração destas tecnologias. Uma maior taxa de penetração pode ser obtida de duas maneiras: (i) Por meio de uma abordagem difícil: criar uma legislação que obriga ou recomenda

a integração desses sistemas de comunicação em carros novos, ou (ii) Por meio de uma abordagem fácil: o desenvolvimento de aplicações atraentes e comercialmente convincentes. Por isso, é muito importante demonstrar a viabilidade e utilidade deste tipo de aplicação, a fim de convencer as autoridades locais e/ou os usuários finais de sua efetiva contribuição no aumento da segurança na condução de veículos (TSUGAWA, 2002).

Outro ponto importante a ser ressaltado é que a maioria dos dispositivos móveis com GPS utilizam um sistema de localização híbrido denominado GPS Assistido (*A-GPS Assisted Global Positioning System*) que consiste em utilizar, quando disponível, uma conexão com internet para prover maior precisão à localização do GPS e maior velocidade na captura das coordenadas geográficas (DJUKNIC; RICHTON, 2001).

1.3 Organização

O trabalho começa com apresentação dos trabalhos relacionados no Capítulo 2, agrupados pelos seguintes assuntos específicos: ADAS, Arquitetura WAVE, ZigBee, Arduino, Localização em VANET e Plataforma Android. O capítulo 3 fala sobre os Trabalhos Relacionados. O Capítulo 4 descreve o ambiente de experimentação e a descrição dos testes realizados. A análise dos resultados obtidos é feita no Capítulo 5. As conclusões e trabalhos futuros são descritos no Capítulo 6.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentadas as principais características das tecnologias relacionados à proposta desta dissertação, são elas: ADAS, Arquitetura WAVE, Padrão ZigBee, Arduino, Localização em VANET e Plataforma Android.

2.1 ADAS

Os ADAS são sistemas utilizados para aumentar o conforto e segurança dos motoristas. Eles podem ser utilizados, para aliviar o motorista em muitas rotinas indesejáveis, na tarefa de condução, assim como reduzir o número de acidentes decorrentes de falha humana (HEIJDEN; MARCHAU, 2002).

Regiões de cruzamento podem ser perigosas em situações onde a via é mal sinalizada ou possui pouca visibilidade entre os veículos. Este trabalho foca-se no desenvolvimento de um ADAS à previsão de colisão entre veículos em cruzamentos. Nesse tipo de sistema, a necessidade do funcionamento, em tempo real, é muito importante, uma vez que o tempo da tomada de decisões, para avaliar um risco ou evitar um acidente, é muito curto quanto à velocidade do veículo e periculosidade da situação. Com isso, o conhecimento da posição de um veículo, em relação aos veículos vizinhos, é fundamental, para que o motorista possa decidir, de forma segura, a aceleração e desaceleração de seu veículo e, dessa forma, evitar acidentes (MARCHAU; HEIJDEN; MOLIN, 2005).

Existem vários trabalhos de ADAS destinados à previsão de colisão em cruzamento propostos na literatura. Dentre os principais, podemos citar o trabalho realizado por Li e Wang (2006) os quais utilizam o conceito

de condução colaborativa, para determinar rotas e velocidades de circulação, para que os veículos, transitando em direção a um cruzamento, não colidam. O sistema desenvolvido destina-se a cenários de trânsito lento e com veículos pesados, possuindo capacidades de aceleração e desaceleração relativamente baixas. Neste trabalho foram propostos quatro algoritmos de planejamento de rotas e foram analisadas algumas estratégias de comunicação das informações entre os veículos. Os experimentos foram realizados por simulação.

No trabalho realizado por Ammoun, Nashashibi e Laugeau (2007), é proposto um protótipo, para previsão de colisão em cruzamentos, utilizando padrão IEEE 802.11 e um receptor GPS de baixo custo. O sistema foi testado por meio de experimentação em ambiente real. Neste trabalho foi utilizado um algoritmo com o qual se utiliza uma função de risco de colisão, baseado no tempo de frenagem do veículos e com o qual, também, utiliza-se Filtro de Kalman, para correção da precisão das coordenadas do GPS e da latência de transmissão dos dados entre os veículos e da captura do sinal do GPS.

Outro trabalho realizado por meio da experimentação em ambiente real, foi desenvolvido por Hsu et al. (2009). O sistema proposto é dividido em duas partes: unidade de posicionamento e unidade veicular. A unidade de posicionamento é usada para fornecer a posição e informações de cabeçalho do módulo GPS. Já a unidade veicular é usada para receber informações de frenagem, aceleração e outros sinais vindos de uma interface CAN (*Controller Area Network*). Os hardwares móveis são construídos com um sistema embarcado usando o processador X86 em um sistema Li-

nux. Os veículos se comunicam via DSRC e protocolo WAVE. Os alertas são visualizados em tempo real na tela de um notebook.

O protótipo desenvolvido neste trabalho foi realizado com base nos requisitos para aplicações de previsão de colisão no trânsito, propostos por Xu et al. (2004):

- Tamanho do pacote: aproximadamente 100 bytes.
- Alcance da transmissão: 300 metros.
- Latência tolerada: aproximadamente 100 ms.

Os ADAS utilizam as informações fornecidas pelos sensores locais de um veículo e de seus vizinhos para construir aplicações voltadas à segurança do condutor. Neste trabalho utilizam-se informações dos receptores GPS de dispositivos móveis com plataforma presentes nos veículos para adquirir informações essenciais para funcionamento de um ADAS.

Para realização deste trabalho foi definido um quadro de dados com as seguintes informações:

- Latitude.
- Longitude.
- Velocidade de deslocamento do nó.
- Precisão da coordenada geográfica.
- Tempo decorrido entre a aquisição de duas coordenadas.

No contexto de redes veiculares, a localização dos veículos está limitada às ruas, estradas e rodovias. Com isso, a utilização da API do Google

Maps para Android juntamente com o recurso de A-GPS, permite que o sistema trabalhe com precisões de localização de até 5 metros.

Quanto à velocidade fornecida pelos receptores GPS, ela é muito precisa sob condições de velocidade e/ou aceleração constantes, no entanto, é menos precisa em situações com rápidas mudanças de velocidade (WITTE; WILSON, 2004). O protótipo desenvolvido contribuiu, também, para avaliar a eficácia dessas informações para aplicações de segurança no trânsito.

A próxima seção descreve as principais características da Arquitetura WAVE que foi desenvolvida, especificamente, para realizar a comunicação em ambiente veicular.

2.2 Arquitetura WAVE

Os primeiros esforços para padronização das redes veiculares foram realizados em 1999 pela FCC nos Estados Unidos, com a alocação de uma faixa de 75 MHz no espectro de frequência de 5.9 GHz (5,850 a 5,925 GHz), para funcionamento exclusivo das redes veiculares. Essa faixa é chamada DSRC (*Dedicated Short-Range Communications*). Como pode ser visto na Figura 2, essa faixa de frequência é dividida em 7 canais de 10 MHz cada, com uma reserva de 5 MHz em cada extremidade da faixa.

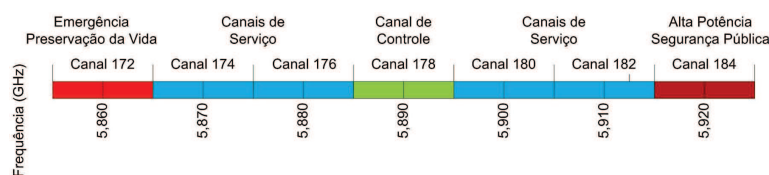


Figura 2 Espectro para aplicações DSRC (ALVES et al., 2009).

Em 2004, o IEEE iniciou a padronização das comunicações em redes veiculares dentro do grupo de trabalho IEEE 802.11. O padrão ficou co-

nhecido como IEEE 802.11p WAVE (*Wireless Access in Vehicular Environments*). Em abril de 2010, a proposta 11 do padrão IEEE 802.11p obteve 99% de aprovação (IEEE COMPUTER SOCIETY, 2010). O padrão é composto por documentos da família IEEE 1609, sendo eles:

- 1609.0: descreve a arquitetura e os serviços necessários ao WAVE.
- 1609.1: descreve o gerenciamento de recursos.
- 1609.2: descreve os serviços de segurança para aplicações e mensagens.
- 1609.3: descreve os serviços de rede.
- 1609.4: descreve a operação em múltiplos canais e as extensões para o 802.11p.

A tecnologia de rádio do WAVE é, essencialmente, um ajuste do IEEE 802.11a, para uma operação com baixo *overhead* e equivalente ao IEEE 802.11e EDCA (*Enhanced Distributed Channel Access*), com qualidade de serviço (JIANG; DELGROSSI, 2008). Por trabalhar na faixa DSRC, os nós dessa rede conseguem se comunicar com taxas de transmissão de dados que variam de 3 a 27 MB/s e com uma distância máxima de 1000 metros entre os nós. O padrão consegue garantir transmissão de dados entre os nós com velocidades relativas de até 200 km/h. Um canal de comunicação exclusivo é reservado para controle das comunicações e os outros canais são utilizados para diferentes categorias de serviços da rede (ALVES et al., 2008).

Na próxima seção iremos descrever as principais características do padrão ZigBee, que é a tecnologia de transmissão utilizada no desenvolvimento deste trabalho.

2.3 Padrão ZigBee

O ZigBee é um padrão de rádio frequência global que tem como foco a padronização e a interoperabilidade de produtos. O nome ZigBee surgiu como uma analogia à forma que as abelhas transitam entre flores e trocam informação com outras abelhas sobre onde encontrar recursos (SAFARIC; MALARIC, 2006). O padrão foi criado e é mantido pela ZigBee Alliance. Sua implementação é dada pelo padrão IEEE 802.15.4, nas camadas física e MAC (*Medium Access Control*) e pelo padrão ZigBee para as camadas de rede e aplicação. A ZigBee Alliance é um consórcio industrial que visa promover e desenvolver redes sem fio, para fins de controle e monitoramento industrial, residencial, aplicações em sensores médicos, jogos e outras áreas de aplicação onde são necessárias redes de baixo custo, baixa potência e interoperabilidade (GISLASON, 2008).

Existe uma variedade de aplicações em diversos setores da sociedade que utilizam essa tecnologia. A Figura 3 exemplifica algumas dessas áreas.

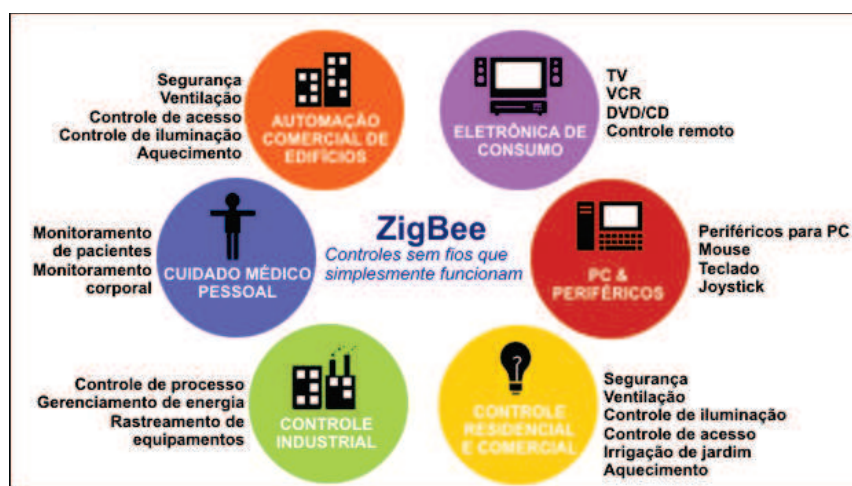


Figura 3 Aplicações do ZigBee. Adaptado de Gislason (2008).

As principais características do padrão ZigBee são (RAMYA; SHANMUGARAJ; PRABAKARAN, 2011):

- Auto reparação.
- Suporta um grande número de nós.
- Facilidade de implementação.
- Baixo consumo de energia.
- Grande número de fornecedores.
- Protocolos de padrão aberto.

2.3.1 Pilha de protocolos

A pilha de protocolos do padrão ZigBee possui elementos semelhantes ao modelo OSI, que são as camadas física (PHY), MAC (*Medium Access Control*) e rede (NWK). Já as demais camadas (transporte, sessão, apresentação e aplicação) estão englobadas na camada de aplicação do modelo ZigBee que é composto pela APS (*Application Sub Layer*) e ZDO (*ZigBee Device Object*). As camadas física e MAC são definidas pelo padrão IEEE 802.15.4, já o padrão ZigBee define as especificações da camada de redes para as topologias de rede e proporciona um framework para desenvolvimento de aplicações na camada de aplicação (GISLASON, 2008). A Figura 4 ilustra a arquitetura do padrão ZigBee.

As principais características da camada física estão resumidas na Tabela 1. As ondas de rádio utilizam faixas de frequências livres: 868 MHz para aplicações Europeias, 902- 928 MHz para América do Norte e 2.4 GHz para aplicações globais. Existe um único canal entre 868 e 868.6 MHz, 10

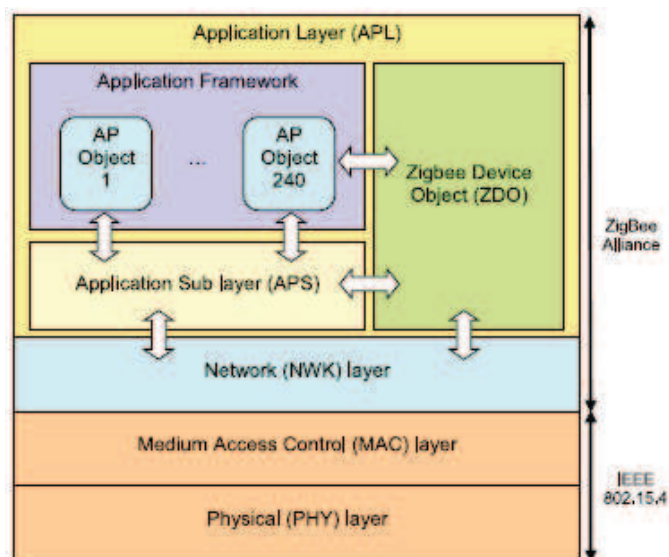


Figura 4 Arquitetura Padrão do ZigBee (BARONTI et al., 2007).

canais entre 962.0 e 928.0 MHz e 16 canais entre 2.4 e 2.4835 GHz. Todas as três bandas usam DSSS (*Direct Sequence Spread-Spectrum*) como técnica de modulação (SAFARIC; MALARIC, 2006).

Tabela 1 Principais características da camada física

	2450 MHz	915 MHz	868 MHz
Transmissão	250 kbps	40 kbps	20 kbps
Largura do canal	5 MHz kbps	2 MHz kbps	1 MHz
Canais	16	10	1
Licença	Global	América do Norte	Europa

A principal função da camada física é transformar pacotes de dados digitais em ondas eletromagnéticas que serão transmitidas pelo ar e vice-versa. Além disso, essa camada é também responsável pela:

- Ativação e desativação do transmissor de rádio.

- Detecção da potência dentro do canal (*ED - Energy Detection*).
- Indicação da qualidade do canal (*LQI - Link Quality Indication*).
- Seleção da frequência do canal para transmissão e recepção dos dados.

A camada MAC define os Dispositivos de Função Completa (*FFDs - Full Function Devices*), operando como roteadores ou coordenadores de rede e Dispositivos de Função Reduzida (*RFDs - Reduced Function Devices*) que operam apenas como dispositivo final. Os dois tipos principais de topologias, considerados no padrão IEEE 802.15.4, são denominadas estrela e ponto a ponto. Na topologia em estrela, um modelo mestre/escravo é adotado. Um FFD assume o papel de coordenador da PAN (*Personal Area Network*); os outros nós podem ser RFDs ou FFDs e somente comunicam-se com o coordenador da PAN. Na topologia ponto a ponto, um FFD pode se comunicar com outros FFDs, em raio de cobertura e podem retransmitir mensagens para outros FFDs, fora da sua cobertura de rádio, por meio de um FFD intermediário, formando uma rede *multihop*. Um coordenador da PAN é selecionado para administrar a operação da rede (BARONTI et al., 2007).

Os dispositivos da rede ZigBee operam em dois modos diferentes (ONDREJ et al., 2006):

- *Beacon*: Utilizado quando é necessário economizar energia. Para isso, a maioria dos dispositivos *End Devices* ficam em modo *Sleep*. Todos os nós, com funções de roteador, sinalizam sua presença aos outros roteadores da mesma rede. Já os *End Devices* não precisam sinalizar a sua presença, mas têm de ser configurados para perceberem o período em que ocorrerá a sinalização;

- *Non-Beacon*: Não foca na economia de energia pelos dispositivos da rede. Os nós da rede permanecem sempre com os seus receptores ativos, consumindo mais energia. A camada MAC é responsável pela programação (escalonamento) das transmissões de quadros no modo *non-beacon* usando o método CSMA/CA (*Carrie Sense Multiple Access With Collision Avoidance*). Neste método, cada nó verifica o canal antes de iniciar a sua transmissão (avaliação de canal livre) para diminuir a possibilidade de transmissões simultâneas. Se o canal estiver livre, o nó inicia a sua transmissão. Quando o nó suspeitar da possibilidade de colisão, aborta a transmissão e aguarda um período aleatório para tentar novamente.

Além da transferência de dados, a camada MAC oferece as funcionalidades de varredura de canal e associação/dissociação. O procedimento de varredura envolve a verificação de vários canais lógicos por meio do envio de *beacons* de requisição e escuta de mensagens (verificação ativa, para FFDs) ou simplesmente *beacons* para escuta (verificação passiva, para RFDs) para localizar PANs e coordenadores existentes. As camadas mais altas decidem em qual PAN entrar e depois pedem à camada MAC para iniciar um processo de associação para PAN selecionada. Esse procedimento envolve o envio de uma requisição ao coordenador e espera da mensagem de aceitação correspondente. Se aceito na PAN, o nó recebe um endereço curto de 16 bits que pode ser usado, mais tarde, no lugar do endereço IEEE estendido de 64 bits (BARONTI et al., 2007).

A camada de rede do Zigbee identifica três tipos de dispositivos: *End Device*, Roteador e Coordenador. O Zigbee *End Device* corresponde a um RFD ou a um FFD atuando como um dispositivo simples. O roteador é um

FFD com capacidade de roteamento. O coordenador é um FFD que gerencia toda a rede. Além da topologia em estrela (que, naturalmente, mapeia a topologia correspondente no IEEE 802.15.4), a camada de rede suporta, também, topologias mais complexas como árvore e malha. Entre as funcionalidades fornecidas pela camada de rede estão: roteamento com múltiplos saltos, descoberta e manutenção de rotas, segurança e entrada/saída da rede com atribuição de um endereço curto (16 bits) para dispositivos aderidos recentemente (BARONTI et al., 2007).

Principais características das topologias de rede do ZigBee:

- *Mesh* (Malha ou Ponto-a-Ponto): Na topologia Mesh, a rede pode se ajustar, automaticamente, tanto na sua inicialização como na entrada ou saídas de dispositivos na rede. A rede se auto-organiza para otimizar o tráfego de dados. Com vários caminhos possíveis, para a comunicação entre os nós, este tipo de rede pode abranger em extensão uma longa área geográfica.
- *Cluster Tree* (Árvore): Semelhante à topologia de Malha, esta é uma rede em árvore, que tem uma hierarquia muito maior, e o coordenador assume o papel de nó mestre para a troca de informação entre os nós *Router* e *End Device*.
- *Star* (Estrela): É uma das topologias de rede ZigBee mais simples de ser implantada. É composta de um nó Coordenador, e quantos nós *End Device* forem precisos. Este tipo de rede deve ser instalada em locais com poucos obstáculos à transmissão e recepção dos sinais, como por exemplo, em uma sala sem muitas paredes ou em locais abertos.

A Figura 5 mostra os dispositivos suportados em cada tipo de topologia de rede ZigBee.

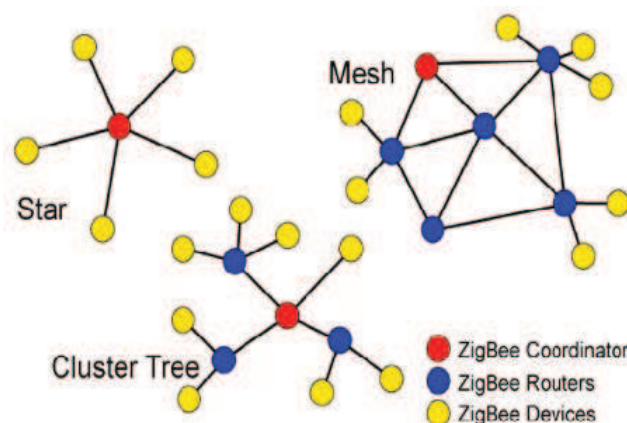


Figura 5 Topologias de rede ZigBee (GHAZY; OZKUL, 2009)

O algoritmo de roteamento depende da topologia utilizada na rede sensor. Na topologia em árvore, os roteadores guardam as informações de seu próprio endereço, dos filhos e de seu pai. Dada a forma como os endereços são atribuídos, um roteador que precisa transmitir uma mensagem pode facilmente determinar se o destino pertence a uma árvore com raiz em um dos roteadores filhos ou é um de seus *end-devices* filhos. Se assim for, ele encaminha o pacote para o filho apropriado, caso contrário, ele encaminha o pacote para o seu pai. A topologia de rede em malha é mais complexa de manipular e o modo *beacon* não é permitido, mas ela é mais robusta e tolerante a falhas. Os roteadores mantêm uma tabela de roteamento (RT) e empregam um algoritmo de descoberta de rota AODV (*Ad hoc On Demand Distance Vector*) para construir ou atualizar essa estrutura de dados em todos os nós do caminho (BARONTI et al., 2007).

A camada de aplicação do padrão Zigbee consiste de um conjunto de Objetos de Aplicação (*APOs - Application Objects*) distribuídos por vários nós na rede. Um APO é um módulo de software que controla uma unidade de hardware (transdutor, *switch*, lâmpada) disponível em um dispositivo. A cada APO é atribuído um número único, que pode ser usado como extensão do número do dispositivo de rede, para auxiliar a interação entre eles. O Objeto de Dispositivo Zigbee (*ZDO - Zigbee Device Object*) é um objeto especial que oferece serviços para os APOs: permite que eles descubram dispositivos na rede, quais recursos eles implementam e quais serviços de comunicação, rede e gerenciamento de segurança fornecem. A Sub Camada de Aplicação (*APS - Application Sublayer*) fornece serviço de transferência de dados para os APOs e ZDO (BARONTI et al., 2007).

A próxima seção descreve as principais características do módulo que implementa o padrão ZigBee utilizado neste trabalho.

2.3.2 Módulos Xbee

O Xbee é um modelo de hardware baseado no padrão ZigBee fabricado pela *Digi International*. Os módulos são compostos por um microcontrolador que contém o *firmware* com a implementação do protocolo ZigBee e um transceptor. A Digi disponibiliza dois grupos principais de componentes Xbee (DIGI, 2012):

- XBee/XBee-Pro Série 1 (XBs1): Utilizam a norma IEEE 802.504 como protocolo de rede para topologias de rede ponto a ponto e estrela.
- XBee/XBee-Pro Série 2 (XBs2): Utilizam como protocolo de rede o ZigBee permitindo, além das topologias ponto a ponto e estrela, a implementação de redes em malha.

Os dois grupos de módulos possuem a mesma forma física e muitos pinos são compatíveis, mas como as especificações de hardware não são muito diferentes, os dois grupos não são interoperáveis. Os modelos Série 2 podem variar de acordo com as seguintes características:

- Taxa de Transmissão: Modelos com frequência de operação maior possuem menor alcance e maior taxa de transmissão.
- Antena: A maior cobertura é alcançada com antenas externas (conector RPSMA e U.FL), seguido pelos modelos com antena Whip e por último antena Chip (integrada). A Figura 6 exemplifica os modelos de antenas utilizadas.
- Potência: Os módulos são fabricados em duas versões, a versão XBee PRO e a XBee. A principal diferença entre eles é que a potência de transmissão do modelo PRO é maior. Logo, proporcionam um alcance mais elevado. Por outro lado, o consumo de potência do XBee em relação ao XBee PRO é, significativamente, menor.
- Programação: O modelo S2B possui um microcontrolador adicional incorporado ao módulo, que permite ao projetista embarcar um *firmware* próprio que pode interagir com o *firmware* do fabricante.

Os módulos XBee comunicam-se de forma serial com qualquer dispositivo UART (*Universal Asynchronous Receiver/Transmitter*) e podem operar de duas maneiras (DIGI, 2012):

- Modo AT: Toda a informação recebida no pino de entrada do módulo é colocada em espera no *buffer* de transmissão para ser transmitida

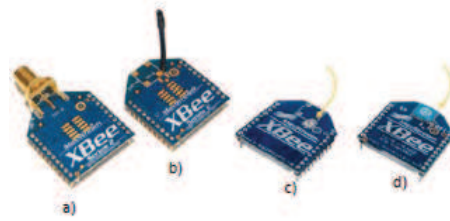


Figura 6 a) RPSMA, b) Whip, c) U.FL connector, d) Chip integrado.

pela antena. Da mesma forma, toda a informação recebida pela antena é colocada no *buffer* de recepção para ser enviada ao pino de saída. Esse modo de operação é fácil de usar, contudo, possui restrições como impossibilidade de configuração e utilização remota dos módulos, identificação do endereço de origem dos dados recebidos e recebimento de mensagens de status indicando o sucesso ou a natureza de uma falha na transmissão.

- Modo API (*Application Programming Interface*): Não possui as restrições do modo AT, porém, sua configuração é mais complexa. Nesse modo de operação, todos os dados enviados e recebidos pelo módulo XBee são organizados em quadros. A Figura 7 mostra a estrutura geral de um quadro utilizado no modo API. O byte 0x7E é o valor hexadecimal que representa o delimitador de início de qualquer quadro. O tamanho do quadro é informado pelos bytes 2 e 3. Os bytes de 4 a n definem a operação a ser realizada (*cmdID*) e os dados que compõem o quadro de dados (*cmdData*). O último byte contém a soma de verificação (*checksum*) que é definida como $0xFF - \sum_4^n bytes[i]$. A Figura 8 mostra os tipos de quadro suportados pelo modo API.

Todos quadros possuem os seguintes campos em comum:

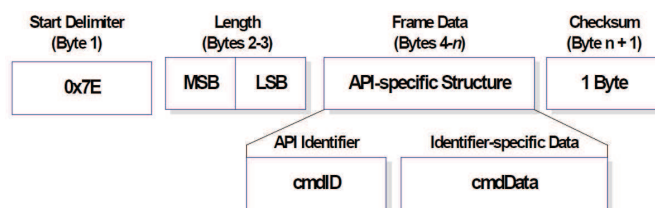


Figura 7 Quadro do Modo API (DIGI, 2012).

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
Remote Command Request	0x17
Create Source Route	0x21
AT Command Response	0x88
Modem Status	0x8A
ZigBee Transmit Status	0x8B
ZigBee Receive Packet (AO=0)	0x90
ZigBee Explicit Rx Indicator (AO=1)	0x91
ZigBee IO Data Sample Rx Indicator	0x92
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97
Over-the-Air Firmware Update Status	0xA0
Route Record Indicator	0xA1
Many-to-One Route Request Indicator	0xA3

Figura 8 Tipos de Quadros do Modo API (DIGI, 2012).

- Delimitador de início: Representado pelo valor 0x7E;
- Tamanho: Quantidade de bytes entre o campo tipo e o campo de dados;

- Tipo: Tipo do quadro a ser enviado;
- *Checksum*: Soma de verificação (*checksum*) definida como $0xFF - \sum_4^n bytes[i]$, onde n é o último byte antes do campo *Checksum*.

O tipo de quadro utilizado para transmitir dados no modo API foi o (*0x10 - ZigBee Transmit Request*), que possui os seguintes campos específicos:

- ID do quadro: Determinar se o pacote receberá ou não confirmação (*ACK - Acknowledge*) de recebimento.
- Endereço de 64 bits: Endereço de destino atribuído pela identificação única do módulo Xbee (Identificação de fábrica). O endereço 0x0000000000000000 é reservado para o coordenador da rede e o endereço 0x000000000000FFFF é destinado para *broadcast*.
- Endereço de 16 bits: Endereço de rede do módulo Xbee destino. Esse número é dado pelo coordenador a um determinado módulo, no momento de ingresso na rede. Quando o endereço não é conhecido, ele deve ser configurado com o valor 0xFFFE (*broadcast*).
- Saltos: Quantidade de saltos realizadas em uma transmissão *broadcast*.
- Opções: Opções adicionais, como por exemplo, habilitar criptografia do pacote.
- Dados: São as informações a serem transmitidas.

Quando um módulo Xbee recebe um pacote do tipo 0x10, ele é enviado para fora da UART usando um pacote do tipo 0x90 (*ZigBee Receive Packet*). Esse pacote é composto pelo campo origem(64 bits e 16 bits) que

identifica a fonte da mensagem, campo de opções que informa detalhes sobre o envio do pacote (Reconhecimento, Broadcast, Criptografia) e o campo de dados.

A configuração dos módulos pode ser realizada por meio de um terminal serial ou utilizando a ferramenta X-CTU, disponibilizada pela Digi International, que permite o carregamento do *firmware* e a configuração de todos os parâmetros do dispositivo, considerando uma interface intuitiva. Com essa ferramenta, o modo de operação AT ou API pode ser configurado, facilmente, assim como o papel do dispositivo na rede (Coordenador, Roteador ou Dispositivo Final).

Neste trabalho foi utilizado o Xbee modelo Pro S2B. Esses módulos possuem uma potência de transmissão de até +18 dBm e a sensibilidade do receptor é -102 dBm. Segundo o fabricante, o alcance deste dispositivo pode chegar a 3,2 km em transmissões com visada direta e 90 metros em com obstruções. No entanto, esse número pode variar de acordo com o modelo de antena utilizado e condições ambientais. A Tabela 2 faz uma comparação entre as principais características desse modelo e o padrão WAVE.

Tabela 2 Tabela comparativa entre Xbee Pro S2B e WAVE

	Xbee Pro S2B	WAVE
Especificação IEEE	802.15.4	802.11p
Banda de Frequência	ISM 2.4 GHz	5.9 GHz
Taxa Máxima de Dados	250 Kbps	27 MB/s
Alcance Máximo	3.2 Km	1 Km
Quantidade de Canais	15	7
Largura de banda do canal	5 MHz	10 MHz

A próxima seção descreverá as principais características do Arduino, que é o hardware utilizado junto ao Xbee para realização dos experimentos e desenvolvimento do protótipo.

2.4 Arduino

O Arduino é uma plataforma aberta de prototipação eletrônica composta por hardware (placa controladora) e software (ambiente de desenvolvimento) flexíveis, fáceis de usar e com baixo custo de aquisição. Geralmente são compostos por um microcontrolador Atmel AVR de 8 bits, utilizando chips da série megaAVR, especialmente os ATmega8, ATmega168, ATmega328 e a ATmega1280. O Arduino pode ser interligado a outros módulos pelos hardwares expansivos chamados *shields* (ARDUINO, 2014).

O modelo utilizado neste trabalho é o Arduino Uno, que possui um microcontrolador de 8 bits Atmel ATmega 328p. Sua CPU opera a 16MHz e oferece 14 portas de entrada/saída. Além disso, possui 32kB de memória de programa, 2kB de SRAM e duas interrupções externas que são mapeadas nos pinos 2 e 3 da placa. A comunicação com um computador externo é feita de forma serial por uma porta USB virtualizando uma porta COM (UNO, 2014).

O Arduino utiliza uma linguagem de programação padrão, baseada em *Wiring*, que pode ser estendida pelo uso de bibliotecas C/C++. Sua IDE é uma aplicação multiplataforma, escrita em Java, composta por um editor de código com recursos de realce de sintaxe, parênteses correspondentes e indentação automática. Além disso, a IDE é responsável por compilar e carregar programas para a placa Arduino (ARDUINO, 2014).

Este trabalho utiliza um *shield* para integrar o módulo Xbee com Arduino chamado (*Xbee Shield*). Dentre as informações transmitidas entre os módulos Xbee, a localização de cada veículo é a mais importante. A próxima seção faz um breve resumo sobre as principais técnicas de localização utilizada nas VANETs.

2.5 Localização em VANET

Várias técnicas de localização podem ser utilizadas nas redes veiculares. Nesta seção são descritas algumas características sobre as principais técnicas utilizadas nesse tipo de rede (BOUKERCHE et al., 2008):

- Sistema de Posicionamento Global Geográfico.
- Acompanhamento por mapas (*Map Matching*).
- Navegação estimada (*Dead Reckoning*).
- Localização por celular.
- Processamento de imagem e/ou vídeo.
- Localização relativa *ad hoc* distribuída.

2.5.1 Sistema de Posicionamento Global Geográfico

O Sistema de Posicionamento Global é composto por 24 satélites que operam em órbita ao redor da Terra. Cada satélite orbita na Terra a uma altura de 20.200 km e faz duas rotações completas a cada dia. As órbitas foram definidas de modo que cada região da Terra pode ver, pelo menos, quatro satélites no céu. O GPS utiliza o conceito de Tempo de Chegada (*TOA - Time of Arrival*) para determinar a posição do utilizador. São necessários, no mínimo, 4 satélites para que o sistema funcione corretamente. Os três primeiros satélites são responsáveis por fazer a triangulação do sinal. O quarto satélite envia um sinal para o receptor com a informação do tempo (hora), em que ocorrem as emissões, evitando, assim, que o receptor necessite de um relógio atômico. Além disso, os receptores GPS têm um erro de

localização médio que varia entre 10 e 30 metros (KAPLAN; HEGARTY, 2006).

Uma técnica comumente utilizada, para diminuir os erros do GPS, é o GPS diferencial (*DGPS, Differential GPS*). Um receptor GPS estacionário é colocado em um ponto de referência, com coordenadas conhecidas, geralmente chamado base. Assim, é possível calcular o valor verdadeiro da medida. Comparando-se o valor medido com o verdadeiro, obtém-se a correção diferencial, que deve ser aplicada às medidas do receptor usuário (KAPLAN; HEGARTY, 2006). Os receptores DGPS têm um erro de localização médio de 1 metro.

Outro método que vem ganhando espaço por ser muito utilizado em dispositivos móveis, como celulares por exemplo, é chamado GPS Assistido (*A-GPS Assisted Global Positioning System*). Este sistema consiste em utilizar, quando disponível, uma conexão com internet para prover maior precisão à localização do GPS e maior velocidade na captura das coordenadas geográficas (DJUKNIC; RICHTON, 2001).

O A-GPS consiste de um ou mais servidores assistentes (A-GPS Server), responsáveis por fazer o download completo da lista de coordenadas de todos os satélites presentes no espaço (também conhecida como almanaque). Quando isto acontece, a operadora de telefonia rastreia a torre na qual o celular com A-GPS está conectado. Como a operadora conhece as coordenadas de cada torre (*Base Station*), ela processa o almanaque (juntamente com outros dados) e separa as coordenadas dos satélites que, provavelmente, estão visíveis para o celular requisitante (*Handset With GPS*). Após a filtragem, essas informações são enviadas ao aparelho por meio da rede de telefonia. Esse sistema poupa o aparelho de diversas tarefas, diminuindo

o Tempo de Localização Inicial (*TTF* - *Time To First Fix*) de todo o sistema.

A Figura 9 mostra a estrutura básica para o funcionamento A-GPS.

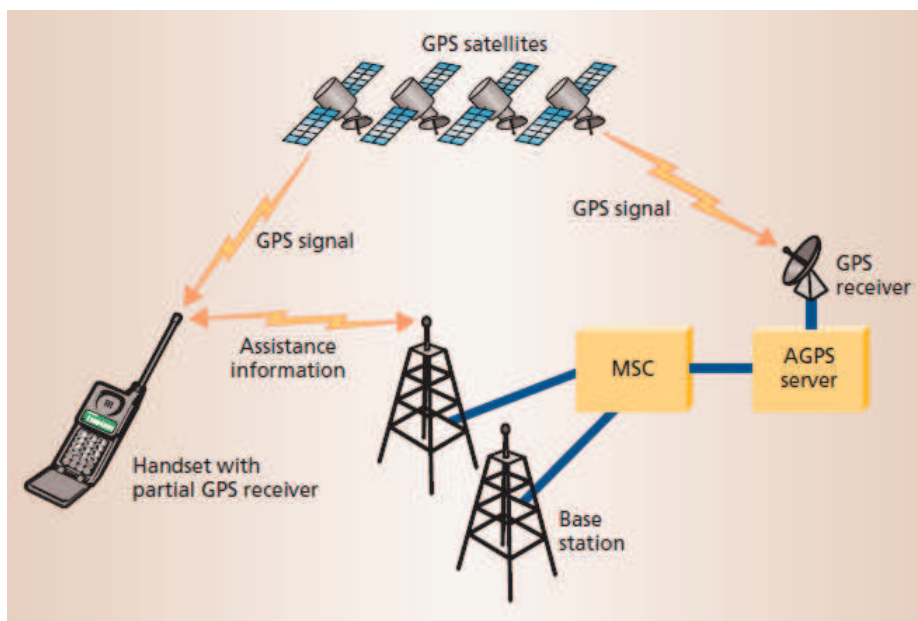


Figura 9 Funcionamento do A-GPS (DJUKNIC; RICHTON, 2001)

Abordagens híbridas, para sistemas localização, utilizam múltiplas tecnologias de geolocalização, para obter uma solução mais robusta para problemas de precisão e cobertura. Nas próximas seções desse capítulo, serão mostradas algumas técnicas, que podem ser utilizadas, para melhorar ou até mesmo substituir o GPS, quando ele não estiver disponível.

2.5.2 Acompanhamento por Mapas (*Map Matching*)

Informações de posição, baseadas puramente em coordenadas, são, muitas vezes, difíceis de interpretar para um motorista. No intuito de ajudar o motorista a relacionar as informações de posição para uma localização física, podem ser utilizados sistemas de navegação onde é exibida a posição do veículo em um mapa. Além disso, em condições normais, a localização e trajetória de um carro são restringidos pela rede rodoviária. Assim, um mapa digital da rede de estradas pode ser utilizado, para impor restrições sobre a solução de navegação, no sistema de navegação no interior do veículo.

O *Map Matching* é uma técnica utilizada, para alinhar as coordenadas, obtidas por receptores GPS, com sua respectiva posição em mapas digitais. Essa técnica é um passo de pré-processamento fundamental, para muitas aplicações, tais como o gerenciamento de veículos em movimento, análise de fluxo de tráfego e condução de veículos (LOU et al., 2009).

Uma série de algoritmos para *Map Matching* foram desenvolvidos por pesquisadores de todo mundo, utilizando diferentes técnicas, como por exemplo: Análise Espacial das Redes Viárias, Filtro de Kalman e Lógica Fuzzy. O desempenho destes algoritmos tem melhorado ao longo dos anos em virtude da aplicação de técnicas avançadas e melhoria nas técnicas de posicionamento geográfico (QUDDUS; OCHIENG; NOLAND, 2007).

2.5.3 Posição estimada (*Dead Reckoning*)

A técnica *Dead Reckoning* consiste em calcular a posição atual de um veículo por meio de sua última localização conhecida e do uso de informações sobre o movimento do veículo, como a direção, velocidade, aceleração, distância e tempo. A última posição conhecida pode ser obtida, por exemplo,

pela utilização de receptores GPS ou pela localização por referência conhecida (estrada cruzamento, parques de estacionamento, em casa, etc) em um mapa digital. As informações de deslocamento podem ser obtidas por meio de sensores como odômetros, bússolas digitais e giroscópios. Esta técnica pode acumular muitos erros caso o sistema de localização fique indisponível por muito tempo (KUKSHYA; KRISHNAN; KELLUM, 2005).

2.5.4 Localização por celular

Celulares necessitam de, pelo menos, uma estação de base, para estabelecer uma comunicação, contudo, várias estações de base podem ouvir e comunicar-se com um telefone celular, em qualquer momento. Este fato permite que um número grande de técnicas de localização possam ser utilizadas para estimar a posição do telefone móvel. Entre as principais técnicas, destacam-se os métodos, baseados em medidas da intensidade de sinal (*RSSI - Received Signal Strength Indicator*), ângulos de chegada (*AoA - Angle of Arrival*), tempo de chegada (*TOA - Time of Arrival*) e diferença de tempo de chegada (*TDOA - Time Difference of Arrival*) (BOUKERCHE et al., 2007).

O método RSSI baseia-se no fato de que a intensidade de um sinal eletromagnético é inversamente proporcional ao quadrado da distância até a fonte do sinal. Com isso, é possível medir esta intensidade e, então, calcular a distância até o emissor do sinal. Esta técnica não é muito precisa pois obstáculos como construções, veículos e árvores, também, reduzem a intensidade do sinal. Nos métodos TOA e TDOA os sinais eletromagnéticos se movem com uma velocidade, aproximadamente, constante. Logo, o tempo que um sinal leva para chegar a um dispositivo móvel pode ser usado

para calcular a distância entre este e a estação base. O mesmo princípio, também, aplica-se a sinais de ultra-som. Quando medimos a diferença entre dois sinais, usa-se o termo TDOA. O método AoA utiliza antenas direcionais para determinar a direção em que está vindo um determinado sinal. É possível determinar a posição de um dispositivo móvel, se houver duas ou mais antenas fixas, para detectar um sinal (ZEIMPEKIS; GIAGLIS; LEKAKOS, 2002).

Essas técnicas geralmente possuem uma baixa precisão quando comparadas com receptores GPS. Contudo, existem estudos relacionados a essas tecnologias que proporcionam localização de alta precisão, como descrito por Xu et al. (2013).

2.5.5 Processamento de imagem e vídeo

Imagens e vídeos podem ser usados, para determinar a posição de um dispositivo móvel por meio de algoritmos de visão computacional. Esses algoritmos podem ser utilizados, por exemplo, para detectar os lados de uma pista considerando imagens de vídeo. Eles utilizam parâmetros geométricos em um sistema de referência local, incluindo a largura da pista, a curvatura lateral da estrada, a distância do veículo baseada no lado esquerdo da pista, o ângulo de direção do veículo e o ângulo de inclinação da câmera. Estes dados locais podem ser transformados, a fim de ser expressos de forma precisa em um mapa digital ambiente. McCall e Trivedi (2006) fazem um estudo detalhado de vários algoritmos, utilizados para detecção da localização das pistas, baseado em imagens de vídeos, além de fornecer uma comparação crítica entre esses algoritmos. Estes algoritmos, geralmente, são combinados

com outras técnicas de localização, tais como GPS e *Map Matching*, para prover uma maior precisão na sua localização.

2.5.6 Localização relativa ad hoc distribuída

Mapas de posição relativa local podem ser construídos por um veículo estimando a distância entre ele e seus vizinhos por meio da troca de informações de distância com os veículos mais próximos de múltiplos saltos (*multihop*). Com esse mapa de posição dinâmica, um veículo pode se localizar, em relação aos veículos das proximidades, assim como localizar os veículos na sua vizinhança (BOUKERCHE et al., 2008). Esse tipo de rede vem sendo muito utilizada em Redes Ad Hoc e Redes Sensores, mas existem algumas soluções propostas para VANET.

Benslimane (2005) propõe um algoritmo de localização distribuído, para auxiliar veículos não equipados com GPS, na estimativa de suas posições, por meio de veículos vizinhos equipados com GPS. Para estimar uma posição, um veículo não equipado com GPS precisa se comunicar com, pelo menos, três veículos equipados com GPS, em sua vizinhança, a fim de estimar a distância e recolher as suas informações de posição. Quando o número de veículos equipados com GPS nas proximidades é menor do que três, ainda, é possível estimar a direção e a distância do veículo considerando um evento (acidente ou perigo), baseando-se em uma pequena quantidade de informação disponível.

No trabalho realizado por Parker e Valaee (2007), é proposto um sistema de localização, distribuída para VANET, que utiliza RSSI, para estimar as distâncias entre os veículos e um algoritmo de otimização, utilizado para melhorar a estimativa de posição inicial dos veículos, obtido, por

exemplo, por meio de GPS. Essa solução pode ser utilizada para melhorar a precisão de posições calculadas pela técnica *Dead Reckoning* na ausência de sinal do GPS.

Existem alguns trabalhos, como o proposto por Kukshya, Krishnan e Kellum (2005), que se baseiam em determinar a posição dos veículos sem a utilização de GPS. Contudo, a maioria dos aplicativos para VANETs funcionam melhor por um sistema híbrido onde as posições relativas podem, normalmente, ser convertidas em posições globais, quando o sinal de GPS estiver ausente.

Neste trabalho foi utilizado o sistema de localização por A-GPS presente na maioria dos dispositivos móveis com plataforma Android. No contexto de redes veiculares, a localização dos veículos está limitada às ruas, estradas e rodovias. Essa característica permite que a utilização de mapas digitais, combinada com a localização por GPS, melhore, significativamente, a precisão do sistema de localização. Na próxima seção serão descritas as principais características da plataforma Android e alguns detalhes sobre sua API para utilização do Google Maps.

2.6 Plataforma Android

Para criar uma aplicação comercialmente atrativa, este trabalho explora a utilização de dispositivos móveis com plataforma Android e GPS para criar o sistema de prevenção de colisões. Essa escolha é motivada por dois fatores:

- Popularidade: Os *smartphones* superaram, desde 2013, os celulares comuns no número de vendas e participação no mercado. O estudo contabilizou a venda de 435 milhões de aparelhos, no segundo tri-

mestre de 2013 e 51,8% deles foram *smartphones*. Nesse cenário, a plataforma Android possui cerca de 80% do mercado de SOs *mobile* (GARTNER, 2014).

- Recursos: A plataforma Android oferece novas possibilidades para aplicações móveis, oferecendo um ambiente de desenvolvimento de código aberto e kernel Linux. O hardware pode ser acessado por meio de uma série de bibliotecas de API padrão, permitindo ao usuário gerenciar diversos recursos como: Wi-Fi, *Bluetooth* e dispositivos de GPS (ANDROID, 2014).

A plataforma Android oferece um conjunto completo de *softwares* para dispositivos móveis: um sistema operacional, *middleware* e, principalmente, aplicações móveis. As plataformas Windows Mobile e iPhone da Apple oferecem um rico ambiente de desenvolvimento simplificado para aplicações móveis. No entanto, ao contrário do Android, eles são construídos sobre sistemas operacionais proprietários que, muitas vezes, priorizam aplicações nativas em relação às criadas por terceiros e restringem a comunicação entre aplicativos e dados nativos. A plataforma Android oferece novas possibilidades para aplicações móveis, proporcionando um ambiente de desenvolvimento de código aberto e kernel Linux. O hardware pode ser acessado por uma série de bibliotecas de API padrão, permitindo ao usuário gerenciar diversos recursos como: Wi-Fi, *Bluetooth* e dispositivos de GPS (ANDROID, 2014).

As aplicações para Android podem ser desenvolvidas com linguagem Java em ambientes de desenvolvimento como Eclipse, Netbeans ou IntelliJ IDEA. Neste trabalho foi utilizado o ambiente de desenvolvimento Eclipse, que, também, é o mais utilizado pela Google no desenvolvimento

de aplicações para Android. O Eclipse utiliza um *plug-in* chamado ADT (*Android Development Tools*) responsável por facilitar o desenvolvimento, teste e compilação do projeto.

O ADT permite a utilização de um emulador da plataforma Android diretamente do Eclipse facilitando o processo de *debug* e teste de recursos como envio de SMS e realização de ligações telefônicas. Pelo Eclipse é possível visualizar *logs* e arquivos do Android, executar o *garbage collector* e gerenciar vários recursos do sistema operacional (LECHETA, 2013).

Uma das funcionalidades que mais chamam a atenção na plataforma Android é a grande facilidade para construir aplicações integradas com o *Google Maps* e que fazem uso de localização por GPS. Para essa integração duas versões da API de Mapas podem ser utilizadas (LECHETA, 2013):

- Google Maps Android API V1: Compatível apenas com versões anteriores a Android 3.1.
- Google Maps Android API V2: Compatível apenas com versões Android 3.1 e superiores.

Dentre as principais melhorias, realizadas na API V2, podemos citar a utilização de vetores em todo *framework* de mapas, para suportar visualizações 2D e 3D, possibilitando ganho significativo de desempenho, tornando todas interações e animações muito mais fluidas. Além disso, caso a cidade em questão suporte a visualização 3D, esta é feita, automaticamente, se o nível do zoom estiver perto o bastante. Uma classe especial de compatibilidade permite que versões anteriores do Android possam usufruir dos recursos da nova API de mapas (LECHETA, 2013).

No próximo capítulo, são descritos os principais resultados dos estudos relacionadas a VANETs e o padrão IEEE 802.15.4/ZigBee na literatura.

3 Trabalhos Relacionados

Existem poucos trabalhos na literatura relacionados a redes veiculares utilizando o padrão ZigBee. Em alguns trabalhos são utilizados sensores sem fio com padrão IEEE 802.15.4, mas não utilizam toda a pilha de protocolo do padrão ZigBee. Nesta seção são descritos alguns desses trabalhos.

Uma das características fundamentais das VANETs é a sua capacidade de auto reconfiguração. No trabalho desenvolvido por Naranjo et al. (2011), utilizam-se dispositivos com plataforma *TinyOS* que implementam a norma IEEE 802.15.4 até a camada de enlace e o protocolo CTP (*Collection Tree Protocol*) na camada de rede para o roteamento *mesh*. Objetivou-se neste trabalho avaliar a característica de auto reconfiguração das redes utilizando esses dispositivos. Para simular o tráfego de dados para aplicações de segurança no trânsito, os experimentos foram realizados para transmissão de pacotes com 30 bytes no campo de dados, contendo as informações da coordenada geográfica e velocidade do veículo. Esses parâmetros são utilizados como base para qualquer ADAS.

Os testes realizados por Naranjo et al. (2011) foram executados em ambiente real. A frequência de envio dos dados foi de 2Hz. Os testes consistem, de forma geral, em três veículos circulando com velocidades variadas enviando mensagens para um ponto infraestruturado. Pelos resultados deduz-se que a latência da rede *mesh* é de menos de 1 milissegundo em conexão direta a 1 e a 2 saltos. O número de mensagens perdidas foi de 6,25% a velocidades de até 50 km/h e a estrutura da rede *mesh* é reconfigurada

somente quando um dos nós perde a conexão com a rota estabelecida. Para reconfiguração das rotas são utilizados parâmetros como a intensidade do sinal de transmissão e recepção.

Uma extensão do trabalho de Naranjo et al. (2011) visa avaliar a característica de auto reorganização da redes *mesh* em autopistas (NARANJO et al., 2012). O ambiente testado possui menor quantidade de obstruções entre os veículos, no entanto, a velocidade relativa entre eles é maior. Os testes deste trabalho foram realizados com três veículos, circulando em vias públicas de tráfego livre, com variações de velocidade de 80 km/h, 100 km/h e 110 km/h. Os dados são transmitidos em *broadcast* e os nós fazem a retransmissão dos dados por mais um salto. A frequência de envio das mensagens foi de 2Hz. O tamanho do pacote enviado foi de 30 bytes. Os experimentos resultaram em perdas de dados de até 8% e a latência máxima foi de 0,5 segundos em transmissões com múltiplos saltos. Pelos resultados infere-se que esta tecnologia de comunicação é capaz de suportar a transmissão de dados em malha contínua, promovendo eficiência e confiabilidade, para ser usado em um grande conjunto de sistemas de assistência ao condutor.

Já Anurag, Ghosh e Bandyopadhyay (2008) desenvolveram um sistema, para alerta de colisão, utilizando o padrão ZigBee. Seu projeto foi feito, especificamente, para evitar a colisão de caminhões em uma mineradora. Os experimentos foram realizados utilizando simulador. Os dados foram transmitidos em *broadcast* e *unicast*. Os intervalos de envio foram de 0,25 segundos, 0,5 segundos e 1 segundo. Foram transferidos pacotes de dados de 80 bytes. Neste trabalho foram simulados, também, a quantidade de nós da rede, variando de 2 a 10 nós. Não houve variação de velocidade entre os veículos. Os dados transmitidos por *broadcast* tiveram taxa de perda

abaixo de 2%. Já os dados transmitidos em *unicast* tiveram taxas de perdas de até 80%. Os resultados conseguidos pelas transmissões por *broadcast* nas simulações, sugerem que o padrão ZigBee, também, pode ser utilizado com sucesso em aplicações para a prevenção de colisão entre veículos em ambientes urbanos e rodoviários.

Neste trabalho, o padrão ZigBee será avaliado com relação à viabilidade de sua utilização na construção de aplicações voltadas à segurança no trânsito. No próximo capítulo é descrito o ambiente de experimentação, métricas e os testes realizados para alcançar esse objetivo.

4 METODOLOGIA

Visando mostrar que o padrão ZigBee suporta os requisitos de aplicações voltadas à segurança no trânsito, foram realizados experimentos para medir a latência e taxa de perda de pacotes por distância em situações reais de condução. Os experimentos deste trabalho foram realizados, em três cenários diferentes, visando analisar o comportamento da comunicação veicular ambientes de tráfego com diferentes velocidades médias e densidades de obstáculos. Na seção 4.3 são descritos os experimentos executados em cada cenário.

Para mostrar que aplicações comercialmente atrativas podem ser criadas utilizando o padrão ZigBee, foi desenvolvido neste trabalho um protótipo para previsão de colisão entre veículos em cruzamentos sem sinalização. Nesse protótipo é utilizado o padrão ZigBee para a comunicação V2V/V2I e dispositivos móveis munidos de GPS com plataforma Android para adquirir as informações necessárias para o funcionamento do sistema.

A visão geral do sistema e o *hardware* utilizado nos experimentos de latência, taxa de perda e no protótipo são descritos na seção 4.1. O algoritmo utilizado para comunicação dos dados e para construção do sistema de previsão de colisão em cruzamentos são descritos na seção 4.2.

4.1 Arquitetura do Sistema

O protótipo desenvolvido neste trabalho foi projetado para prever colisões em cruzamentos não sinalizados. Para alcançar esse objetivo o sistema desenvolvido utiliza as informações de posição e velocidade do próprio veículo e dos veículos mais próximos ao cruzamento. Essas informações são extraídas com base no receptor GPS de um dispositivo móvel com plata-

forma Android presente no interior de cada veículo. Para que essas informações sejam disseminadas entre os veículos, por meio do padrão ZigBee, foi utilizado um Arduino comportando um dispositivo *xbee* e um *bluetooth*. A Figura 10 mostra o nó construído.



Figura 10 Nó para comunicação veicular

O sistema de transmissão utiliza o Arduino para receber os dados vindos do dispositivo móvel via *bluetooth* e envia os dados para os demais veículos pelo módulo Xbee. O Arduino, também, faz o caminho inverso, recebendo os dados vindos de outros veículos pelo módulo Xbee e envia esses dados para o dispositivo móvel via *bluetooth*. A Figura 11 mostra um diagrama representativo da comunicação dos dados entre Arduino, Xbee e Bluetooth.



Figura 11 Comunicação entre Arduino, Módulo Xbee e Dispositivo Móvel

Os módulos Xbee e *Bluetooth* fazem a comunicação com o Arduino via interface serial. Contudo, os modelos de Arduino, utilizados neste trabalho, possuem apenas uma interface serial para realizar essa comunicação. Para viabilizar o funcionamento simultâneo dos dois módulos, foi utilizada a biblioteca *SoftwareSerial* do Arduino que tem como função virtualizar uma porta serial por meio de seus pinos digitais (ARDUINO, 2014). Uma placa Xbee Shield foi utilizada para conectar o módulo Xbee ao Arduino. O módulo *Bluetooth* foi conectado ao Arduino por fios na interface serial virtualizada utilizando portas lógicas do Arduino.

Para realização dos experimentos e para construção do protótipo desse trabalho foram utilizados os seguintes equipamentos para montagem dos nós transmissores:

- 3 Módulos Xbee Pro S2B.
- 3 módulos bluetooth modelo JY-MCU.
- 3 Arduinos Uno.
- 3 Xbee Shields.
- 3 Mini protoboard 170 pontos.
- 3 Antenas Omnidirecionais de 5dBi.
- 1 Tablet Motorola Xoom com GPS e Android 3.1.
- 1 Celular Samsung Galaxy 5 com GPS e Android 2.2.
- 1 Celular Samsung Galaxy Young com GPS e Android 4.1.

O sistema para previsão de colisão foi desenvolvido no dispositivo móvel utilizando o *framework* de aplicação do Android. O mecanismo montagem e tradução dos quadros de dados do Xbee foi desenvolvido no Arduino. Na próxima seção são mostrados os detalhes de implementação do sistema.

4.2 Desenvolvimento

Além das informações descritas na Seção 2.1, foram adicionados ao quadro de dados, um número para identificação do pacote, um caractere ponto-e-virgula (;) para separar as informações e um caractere cifrão (\$) para determinar o final do campo de dados, dando um total de 47 caracteres ou 47 bytes de dados a serem transmitidos. As informações foram organizadas da seguinte forma:

- Latitude;Longitude;Precisão;Velocidade;Tempo;Id\$.

O módulo Xbee foi configurado para operar no modo API para possibilitar a mudança dinâmica dos destinos das transmissões, recebimento do estado de uma transmissão (sucesso ou falha) e identificação da origem de cada informação recebida.

Visando simplificar os experimentos deste trabalho, não foi realizado nenhum procedimento para proteção das informações transmitidas entre os módulos Xbee. No entanto, o padrão ZigBee possui técnicas de proteção com a utilização de chaves e criptografia que são suportadas pelos módulos Xbee. Um ponto importante, que foi levado em consideração, ao habilitar tais recursos, é a fragmentação. Quando habilitada, a criptografia diminui em 4 bytes o tamanho do campo de dados do quadro do tipo 0x10 para envios sem fragmentação do quadro. A quantidade máxima de informações

armazenadas, no campo de dados, para envio de um quadro, sem a sua fragmentação, é de 84 bytes.

A fragmentação das informações impacta diretamente no desempenho das aplicações destinadas à segurança no trânsito, uma vez que aumentam o tempo de entrega dos dados, a probabilidade de falhas na transmissão e a quantidade de informações trafegando na rede. Neste trabalho, o campo de dados foi definido em 47 bytes, sendo assim, mesmo utilizando criptografia na transmissão dos pacotes, os quadros não seriam fragmentados.

Os módulos Xbee Série 2 implementam toda pilha de protocolo do padrão ZigBee. Uma característica importante desses módulos é que o envio de mensagens por *broadcast* possui um mecanismo de contingência que torna o processo mais seguro, porém, mais demorado. Esse mecanismo, chamado de consciência passiva, funciona da seguinte forma: depois de enviar uma mensagem em *broadcast*, o módulo espera e ouve os seus vizinhos. Se um vizinho não retransmitir o pacote, o módulo faz um novo *broadcast* do pacote (DIGI, 2012).

Para proporcionar um ambiente favorável às aplicações de tempo real, os experimentos foram configurados para transmitir os pacotes de dados por *unicast* e sem confirmação de recebimento.

Como a comunicação entre Arduino e Xbee é realizada com o módulo Xbee, operando em modo API, é necessário que os quadros que chegam no Arduino pela interface serial sejam montados para que as informações possam ser extraídas e, posteriormente, utilizadas pela aplicação. O Arduino foi responsável, também, pela montagem dos quadros que serão enviados via interface serial para o módulo Xbee. Esse processo de criação e interpretação dos quadros pode ser simplificado utilizando a biblioteca xbee-

arduino disponível no endereço <https://code.google.com/p/xbee-arduino/>. Contudo, o módulo utilizado neste trabalho (Xbee Pro S2B) não foi homologado para o funcionamento dessa biblioteca. Com isso, foi necessário o desenvolvimento de funções para realizar tais operações. Os códigos foram criados, de acordo com a necessidade dos experimentos e do protótipo desenvolvidos neste trabalho e podem ser baixados pelo endereço: <http://www.dgti.ufla.br/veicular.zip>. Os tipos de quadros tratados neste trabalho foram:

- 0x10 *ZigBee Transmit Request*.
- 0x90 *ZigBee Receive Packet*.
- 0x88 *AT Command Response*.
- 0x8A *Modem Status*.
- 0x8B *ZigBee Transmit Status*.

O sistema, para previsão de colisão em cruzamentos, desenvolvido para plataforma Android, consiste em calcular o tempo de chegada dos veículos ao cruzamento e emitir um alerta de colisão, quando detectado diferenças no tempo de chegada de até 5 segundos entre dois ou mais veículos ao cruzamento. Esse tempo foi determinado, com base no trabalho realizado por Ammoun, Nashashibi e Laugeau (2007), onde foi determinada uma função do risco de colisão, baseado no tempo de frenagem dos veículos e na estimativa de tempo de reação do motorista levantado por Taoka (1989).

O cálculo do tempo de chegada ao cruzamento é realizado utilizando a fórmula da distância percorrida em função do tempo $S = S_0 + V_0 * t + \frac{a * t^2}{2}$, onde:

- S = Posição final.
- S_0 = Posição inicial.
- V_0 = Velocidade inicial.
- a = Aceleração.
- t = Tempo.

A API do Android, responsável por integrar o Google Maps e a posição geográfica recebida pelo sistema de GPS/A-GPS, fornece, também, a precisão dessa informação. O valor dessa precisão é subtraído da distância entre o veículo e o cruzamento para evitar que a distância seja subestimada. Para o cálculo da distância entre os veículos foi utilizado a fórmula de Haversine. Esse método foi escolhido por ser preciso em distâncias pequenas (CHAMBERLAIN; DUQUETTE, 2007). A equação desse método é expressa por $D = R * c$ (SHUMAKER; SINNOTT, 1984), onde:

- R = Raio da Terra (6.378.137 metros).
- $c = 2 * \arctan2(\sqrt{a}, \sqrt{1-a})$ (Distância angular em raiano entre dois pontos).
- $a = \sin^2(\frac{\Delta latitude}{2}) + \cos(latitude1) * \cos(latitude2) * \sin^2(\frac{\Delta longitude}{2})$.
- $\Delta latitude = latitude1 - latitude2$.
- $\Delta longitude = longitude1 - longitude2$.

O parâmetro tempo foi calculado com base no tempo decorrido entre o recebimento de dois sinais do GPS consecutivos. Para as pacotes recebidos de outros veículos, a latência da transmissão entre os módulos Xbee é

subtraída do tempo decorrido. A aceleração é calculada pela diferença entre as duas últimas velocidades recebidas dividida pelo tempo.

O nó coordenador possui, além das funcionalidades citadas anteriormente, o papel de distribuição dos endereços de destino para cada nó da rede. Essa funcionalidade é necessária em virtude da utilização de transmissão por *unicast* em vez de *broadcast*. O algoritmo consiste em atribuir para cada nó, os endereços de destino referentes aos nós mais próximo do veículo, em cada uma das diferentes vias, que chegam no cruzamento.

O código desenvolvido neste trabalho utiliza uma *thread* responsável por receber as informações dos demais veículos por meio do bluetooth dispositivo móvel. Os dados recebidos são utilizados para plotar no mapa a localização dos veículos vizinhos. No Android, toda vez que é necessário atualizar a interface gráfica, considerando uma *thread* diferente da principal que controla a tela, é necessário a utilização da classe *Handler*. O método *post* da classe *Handler* é responsável por enviar uma mensagem para fila de mensagens do Sistema Operacional que irá processá-la logo que possível.

O trecho de código a seguir demonstra a utilização da classe *Handler* e seu método *post*.

```
1 workerThread = new Thread(new Runnable() {
2     public void run() {
3         while (!stop) {
4             try {
5                 int bytesAvailable=mmInputStream.available();
6                 if (bytesAvailable > 0) {
7                     byte [] packetBytes=new byte[bytesAvailable];
8                     mmInputStream.read(packetBytes);
9                     for (int i = 0; i < bytesAvailable; i++) {
10                        byte b = packetBytes[i];
11                        data += (char) b;
12                        if (data.length() > 10) {
13                            handler.post(new Runnable() {
14                                public void run(){imprimir(data);}
15                            });
16                        }
17                    }
18                }} catch (Exception ex) { stop = true; ex.
19                    printStackTrace(); }
20            }
21        });
```

O código desenvolvido para plataforma Android está disponível em <http://www.dgti.ufla.br/veicular.zip>.

Para facilitar a avaliação do protótipo os experimentos foram realizadas com os veículos circulando com velocidades médias abaixo de 40 km/h nas vias do campus da Universidade Federal de Lavras. O ambiente de experimentação proposto é composto por um ponto infraestruturado, localizado na esquina de um cruzamento e dois veículos circulando em direção ao cruzamento por diferentes vias. Os veículos comunicam seus dados entre si e com o ponto infraestruturado. Um alerta sonoro é emitido sempre que o algoritmo detectar um perigo de colisão. O cenário proposto pode ser visualizado pela Figura 12.

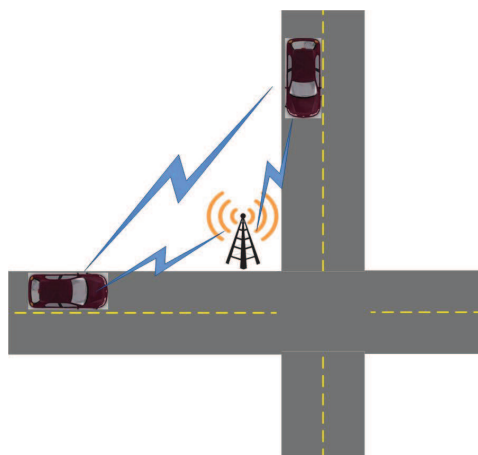


Figura 12 Ambiente de teste do protótipo

No capítulo 4 apresenta-se os resultados obtidos na utilização do protótipo. Na próxima seção são descritos os cenários dos experimentos de avaliação do padrão ZigBee em diferentes situações de velocidade e densidade de obstáculos.

4.3 Cenários

Antes de realizar os testes do protótipo desenvolvido, o padrão ZigBee foi avaliado quanto à sua capacidade de transmissão de dados em um ambiente veicular. Para isso foram realizados testes de latência e taxa de perda de quadros em três cenários distintos:

- Ambiente urbano denso (muitos obstáculos) com velocidades de circulação de até 40 km/h.
- Ambiente urbano pouco denso com velocidades de circulação de até 60 km/h.
- Ambiente rodoviário com poucos obstáculos e velocidades de circulação de até 80 km/h.

Para os testes de latência e taxa de perda de quadros foram utilizados dois nós:

- Um ponto infraestruturado com Xbee configurado como Coordenador para receber os quadros.
- Um veículo móvel com Xbee configurado como Roteador para enviar os quadros.

Os teste de latência, em cada cenário, foram definidos como o tempo decorrido entre o recebimento do primeiro e do ultimo byte do quadro de dados. Todos os experimentos foram realizados utilizando quadros de 65 bytes sem retransmissão de pacotes, com os dados transmitidos entre um veículo em movimento e um ponto infraestruturado.

4.3.1 Ambiente urbano denso

Este cenário é caracterizado por uma grande quantidade de obstáculos entre as vias onde o veículo circula e o ponto infraestruturado. A Figura 13 exhibe o trajeto realizado neste cenário. A seta de cor azul indica o sentido de deslocamento do veículo. Os dados dos testes de latência e taxa de perda neste cenário foram realizados com 5 repetições do trajeto e 5 mil quadros transmitidos. Neste cenário os veículos circulam com velocidades máximas de 40 km/h.



Figura 13 Trajeto ambiente urbano denso

4.3.2 Ambiente urbano pouco denso

Este cenário é caracterizado por poucos obstáculos entre as vias onde o veículo circula e o ponto infraestruturado. A Figura 14 exhibe o trajeto realizado neste cenário. A seta de cor azul indica o sentido de deslocamento

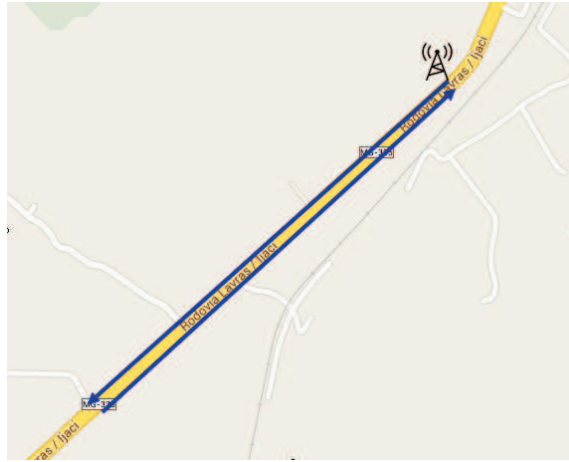


Figura 15 Trajeto ambiente rodoviário

Os resultados dos testes realizados em cada cenário são apresentados no próximo Capítulo.

5 RESULTADOS E DISCUSSÕES

Neste capítulo são mostrados os resultados dos testes de latência, taxa de perda de quadros realizados, em cada cenário, além do resultado do protótipo de previsão de colisão.

5.1 Teste de latência

Os resultados dos testes mostram que a latência média foi muito próxima em vias de trânsito lento (urbano) e em vias de trânsito rápido (rodovia). As variações de velocidade, distância e densidade de obstruções não tiveram grande impacto no valor da latência média que ficou em torno de 63 ms. A latência máxima medida foi de 64 ms nos três cenários e a latência mínima foi de 61 ms. Como o requisito de latência, para aplicações de segurança no trânsito é de 100 ms (XU et al., 2004), pode-se dizer que o padrão ZigBee, no quesito latência, é suficiente para criação de tais aplicações. Os resultados do teste de latência podem ser observados pelo gráfico da Figura 16.

5.2 Ambiente urbano denso

A taxa de perda de quadros no cenário de ambiente urbano denso pode ser visualizada na Figura 17. O gráfico representa a densidade de pacotes perdidos entre o veículos e o ponto infraestruturado ao longo do percurso realizado. A cor vermelha representa áreas onde ocorreram maior concentração de perdas e a cor verde representa as áreas com menor concentração de perdas.

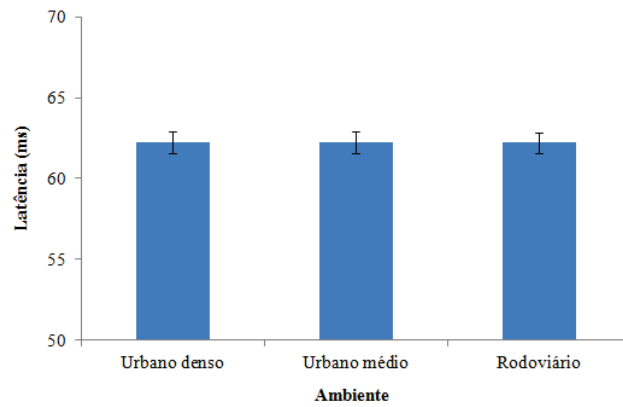


Figura 16 Teste de Latência

A distância máxima alcançada entre o veículo e o ponto infraestruturado foi de 185 metros e nenhum dado foi entregue a uma distância maior que 150 metros. Verificou-se que a maior quantidade de perdas ocorrem nas quadras situadas à direita do ponto infraestruturado. Uma vez que ambos os lados estão repletos de construções que obstruem o sinal, a maior quantidade de perdas do lado direito ocorre em razão do nível da rua estar mais baixo que o nível onde está situado o ponto infraestruturado.

5.3 Ambiente urbano densidade média

Os testes de taxa de perda no cenário de ambiente urbano de densidade média realizados no campus da Universidade Federal de Lavras, obtiveram uma maior distância de transmissão do que no cenário de alta densidade. O alcance máximo entre o ponto infraestruturado e o veículo foi de 475 metros. Com base nos dados levantados, verificou-se que a perda de dados sofreu mais influência dos obstáculos do que da distância entre os pontos de transmissão. Na Figura 18 observa-se que a maior taxa de perda

5.4 Ambiente rodoviário

Os testes de taxa de perda no cenário de ambiente rodoviário obtiveram um alcance máximo de 800 metros nas transmissões com pouquíssimas perdas de dados, como demonstrado na Figura 19. Por esse experimento mostrou-se que o alcance do padrão ZigBee em visada direta utilizando o Xbee modelo S2B Pro com uma antena de 5 dBi foi próximo ao alcance de 1000 metros da especificação do padrão IEEE 802.11p.



Figura 19 Perdas - Ambiente Rodoviário

5.5 Protótipo

Os testes do protótipo, para alerta de colisão em cruzamentos, foram realizados no campus da Universidade Federal de Lavras por meio de experimentos reais. Dois destes testes são apresentados nas Figuras 20 e 21.

No primeiro cenário, representado pela Figura 20, os veículos transitaram em direção ao cruzamento com velocidades médias de 35 e 40 km/h para os carros azul e preto, respectivamente. Com isso, o alerta de colisão foi disparado para um tempo de chegada coincidente de 5 segundos ou



Figura 20 Teste do protótipo para alerta de colisão

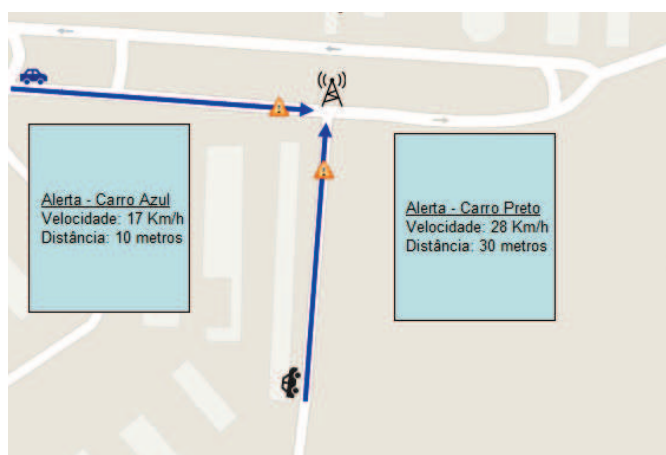


Figura 21 Teste do protótipo para alerta de colisão

menos no cruzamento. Nesse instante, o carro de cor azul registrou uma velocidade de 40 km/h a uma distância de 50 metros do cruzamento. O carro preto registrou no mesmo instante uma velocidade de 37 km/h a 34 metros do cruzamento.

No segundo cenário, mostrado na Figura 21, os veículos transitaram em direção ao cruzamento com velocidades médias de 23 e 26 km/h para os carros azul e preto, respectivamente. No momento em que foi disparado o alerta, o veículo azul registrou uma velocidade de 17 km/h a uma distância de 10 metros do cruzamento. O carro preto registrou uma velocidade de 28 km/h a uma distância de 30 metros do cruzamento.

Pelos testes foram mostrados que o protótipo funciona adequadamente com os veículos transitando com pouca variação de velocidade. Contudo, foi observado que desacelerações bruscas geraram falsos positivos e acelerações bruscas não alertaram situações de colisão em tempo hábil.

6 CONCLUSÃO

Este trabalho tem foco na comunicação *ad hoc* entre veículos utilizando a tecnologia ZigBee. Essa comunicação foi avaliada quanto à sua capacidade de atender os requisitos mínimos para as aplicações de segurança no trânsito. Para isso, foram realizados experimentos para medir a latência e taxa de perda de pacotes por distância em situações reais de condução.

Antes da realização desses experimentos, foram definidas as informações a serem transmitidas e, conseqüentemente, o tamanho de cada quadro de dados. As informações selecionadas são as mínimas necessárias para o funcionamento da maioria dos Sistemas Avançados de Assistência ao Condutor, como latitude, longitude e velocidade do veículo. Outras informações foram utilizadas para auxiliar no desenvolvimento do protótipo para alerta de colisão em cruzamento, como aceleração, precisão da coordenada geográfica e tempo decorrido em relação à última coordenada adquirida.

Os testes de latência registram atrasos máximos de 64 ms em ambos cenários testados. A latência média foi de 63 ms. Esses resultados mostram que o padrão ZigBee atende aos requisitos de latência para redes veiculares. Os testes da taxa de perda sofreram maior impacto em ambientes densos com muitos obstáculos. Como a velocidade de deslocamento em ambientes urbanos densos é menor do que em ambientes menos densos, o padrão, também, mostra-se suficiente neste aspecto.

O protótipo desenvolvido mostrou-se eficaz com veículos se deslocando com pouca variação de velocidade. Isso se deve ao fato de que a precisão da informação de velocidade adquirida pelo GPS é imprecisa em situações de aceleração e desaceleração bruscas.

6.1 Contribuições

A primeira contribuição deste trabalho foi o estudo da utilização do padrão ZigBee em um ambiente veicular, pelos seus experimentos práticos, uma vez que não foram encontrados na literatura muitos trabalhos, utilizando o ZigBee para transmissão de informações entre veículos.

Mostrou-se, neste trabalho, por meio de seus experimentos que, para aplicações de segurança no trânsito, possuindo como característica a transmissão de pacotes de dados pequenos, o padrão ZigBee é uma solução eficaz e de baixo custo, podendo ser utilizado como uma alternativa ao padrão IEEE 802.11p para esse tipo de aplicação.

Por fim, mostramos que os recursos da plataforma Android de dispositivos móveis com GPS podem ser utilizados para criar aplicações de segurança no trânsito comercialmente atrativas, uma vez que esses dispositivos possuem uma grande taxa de aceitação no mercado. Contudo, para aumentar a confiabilidade desses sistemas é necessário que as informações do GPS sejam utilizadas em conjunto com dados extraídos do próprio veículo, como por exemplo, a velocidade capturada diretamente do velocímetro ou utilizando acelerômetro acoplado ao sistema. Além disso, para que sejam criados sistemas de segurança confiáveis, é necessário que outras técnicas de localização sejam utilizadas juntamente com o GPS e possam fornecer a localização do veículos na ausência de seu sinal.

6.2 Trabalhos Futuros

Como trabalhos futuros, pretende-se criar um sistema de alerta de colisão mais robusto combinando os dados do GPS com as informações for-

necidas pelos próprios veículos e outros sensores, como acelerômetros, por meio de uma técnica de fusão de dados.

Outro trabalho a ser realizado é a avaliação do padrão ZigBee, por meio de experimentos práticos, com a finalidade de criar aplicações onde é necessária a realização de transmissões por múltiplos saltos. Nesses trabalhos, os pontos-chave a serem pesquisados serão o algoritmo de roteamento e a característica de auto-reconfiguração da rede.

REFERÊNCIAS

ALVES, R. et al. Redes veiculares: princípios, aplicações e desafios. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 27., 2009, Recife. **Anais...** Recife: SBRC, 2009. p. 1-56.

ALVES, R. D. S. et al. Uma análise experimental da capacidade de redes ad hoc veiculares. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 27., 2008, Rio de Janeiro. **Anais...** Rio de Janeiro: SBRC, 2008. p. 1-6.

AMMOUN, S.; NASHASHIBI, F.; LAURGEAU, C. Crossroads risk assessment using gps and inter-vehicle communications. **IET Intelligent Transport Systems**, Stevenage, v. 1, n. 2, p. 95-101, 2007.

ANDROID. Disponível em: <<http://www.android.com/>>. Acesso em: 20 jan. 2014.

ANURAG, D.; GHOSH, S.; BANDYOPADHYAY, S. GPS based vehicular collision warning system using ieee 802.15: 4 mac/phy standard. In: INTERNATIONAL CONFERENCE ON IEEE, 8., 2008, Hilton Phuket. **Proceedings...** Hilton Phuket: ITST, 2008. p. 154-159.

ARDUINO. **Getting started with Arduino**. Disponível em: <<http://arduino.cc/en/Guide/HomePage>>. Acesso em: 20 jan. 2014.

BARONTI, P. et al. Wireless sensor networks: a survey on the state of the art and the 802.15.4 and zigbee standards. **Computer Communications**, New York, v. 30, n. 7, p. 1655-1695, 2007.

BENSLIMANE, A. Localization in vehicular ad hoc networks. In: _____. **Systems communications**. Montreal: IEEE, 2005. p. 19-25.

BOUKERCHE, A. et al. Localization systems for wireless sensor networks: wireless communications. **IEEE**, Ottawa, v. 14, n. 6, p. 6-12, 2007.

BOUKERCHE, A. et al. Vehicular ad hoc networks: a new challenge for localization-based systems. **Computer Communications**, New York, v. 31, n. 12, p. 2838-2849, 2008.

CHAMBERLAIN, R. G.; DUQUETTE, W. H. **Some algorithms for polygons on a sphere**. Pasadena: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2007. 26 p.

CHENG, H. T.; SHAN, H.; ZHUANG, W. Infotainment and road safety service support in vehicular networking: from a communication perspective. **Mechanical Systems and Signal Processing**, New York, v. 25, n. 6, p. 2020-2038, 2011.

DIGI. **XBee ZB RF Modules**. Disponível em: <<http://www.digi.com>>. Acesso em: 10 dez. 2012.

DJUKNIC, G.; RICHTON, R. Geolocation and assisted gps. **Computer IEEE**, Ottawa, v. 34, n. 2, p. 123-125, 2001.

ESAFETY, W. G. (Ed.). **Strategic research agenda, ICT for mobility**. Ljubljana: RTD, 2008. 33 p.

GARTNER. **Gartner says smartphone sales grew 46.5 percent in second quarter of 2013 and exceeded feature phone sales for first time**. Disponível em: <<http://www.gartner.com/newsroom/id/2573415>>. Acesso em: 20 jan. 2014.

GHAZY, A.; OZKUL, T. Design and simulation of an artificially intelligent vanet for solving traffic congestion. In: INTERNATIONAL SYMPOSIUM ON MECHATRONICS AND ITS APPLICATIONS, 6., 2009, Sharjah. **Proceedings...** Sharjah: IEEE, 2009. p. 1-6.

GISLASON, D. **Zigbee wireless networking**. Burlington: Newnes, 2008. 448 p.

HEIJDEN, R. van der; MARCHAU, V. Innovating road traffic management by its: a future perspective. **International Journal of Technology, Policy and Management**, Bucks, v. 2, n. 1, p. 20-39, 2002.

HSU, C. et al. Verification of on-line vehicle collision avoidance warning system using dsrc. **World Academy of Science, Engineering and Technology**, Las Cruces, v. 31, n. 55, p. 362-368, 2009.

IEEE COMPUTER SOCIETY. **Wireless LAN medium access control: MAC and physical layer (PHY) specifications: amendment 6, wireless access in vehicular environments**. New York: IEEE, 2010. 51 p.

JIANG, D.; DELGROSSI, L. IEEE 802.11 p: towards an international standard for wireless access in vehicular environments. In: VEHICULAR TECHNOLOGY CONFERENCE, 65., 2008, Singapore. **Proceedings...** Singapore: VTC Spring, 2008. p. 2036-2040.

KAPLAN, E.; HEGARTY, C. **Understanding GPS: principles and applications**. Norwood: Artech House, 2006. 726 p.

KUKSHYA, V.; KRISHNAN, H.; KELLUM, C. Design of a system solution for relative positioning of vehicles using vehicle-to-vehicle radio communications during gps outages. In: VEHICULAR TECHNOLOGY CONFERENCE, 62., 2005, Dallas. **Proceedings...** Dallas: VTC, 2005. v. 2, p. 1313-1317.

LECHETA, R. R. **Google Android**. 3rd ed. São Paulo: Novatec, 2013. 821 p.

LI, F.; WANG, Y. Routing in vehicular ad hoc networks: a survey. **Vehicular Technology Magazine**, New York, v. 2, n. 2, p. 12-22, 2007.

LI, L.; WANG, F. Y. Cooperative driving at blind crossings using intervehicle communication. **IEEE Transactions on Vehicular Technology**, New York, v. 55, n. 6, p. 1712-1724, 2006.

LOU, Y. et al. Map-matching for low-sampling-rate GPS trajectories. In: ACM SIGSPATIAL INTERNATIONAL CONFERENCE ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS, 17., 2009, Seattle. **Proceedings...** Seattle: ACM, 2009. p. 352-361.

MARCHAU, V.; HEIJDEN, R. van der; MOLIN, E. Desirability of advanced driver assistance from road safety perspective: the case of ISA. **Safety Science**, Amsterdam, v. 43, n. 1, p. 11-27, Jan. 2005.

MCCALL, J.; TRIVEDI, M. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. **IEEE Transactions on Intelligent Transportation Systems**, New York, v. 7, n. 1, p. 20-37, 2006.

NARANJO, J. E. et al. Evaluation of v2v and v2i mesh prototypes based on a wireless sensor network. In: INTERNATIONAL IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, 14., 2011, Washington. **Proceedings...** Washington: IEEE, 2011. p. 2080-2085.

NARANJO, J. E. et al. Highway test of v2v mesh communications over wsn. In: INTERNATIONAL IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, 15., 2012, Washington. **Proceedings...** Washington: IEEE, 2012. p. 25-30.

ONDREJ, S. et al. Zigbee technology and device design. In: NETWORKING, INTERNATIONAL CONFERENCE ON SYSTEMS AND INTERNATIONAL CONFERENCE ON MOBILE COMMUNICATIONS AND LEARNING TECHNOLOGIES, 1., 2006, Moorea. **Proceedings...** Moorea: ICN/ICONS/MCL, 2006. p. 129-129.

PARKER, R.; VALAEE, S. Vehicular node localization using received-signal-strength indicator. **IEEE Transactions on Vehicular Technology**, New York, v. 56, n. 6, p. 3371-3380, 2007.

QUDDUS, M.; OCHIENG, W.; NOLAND, R. Current map-matching algorithms for transport applications: state-of-the art and future research directions. **Transportation Research Part C: Emerging Technologies**, New York, v. 15, n. 5, p. 312-328, 2007.

RAMYA, C.; SHANMUGARAJ, M.; PRABAKARAN, R. Study on zigbee technology. In: INTERNATIONAL CONFERENCE ON ELECTRONICS COMPUTER TECHNOLOGY, 3., 2011, Kanyakumari. **Proceedings...** Kanyakumari: ICECT, 2011. v. 6, p. 297-301.

SAFARIC, S.; MALARIC, K. Zigbee wireless standard. In: INTERNATIONAL SYMPOSIUM ELMAR-2006 FOCUSED ON MULTIMEDIA SIGNAL PROCESSING AND COMMUNICATIONS, 48., 2006, Zadar. **Proceedings...** Zadar: IEEE, 2006. p. 259-262.

SHUMAKER, B.; SINNOTT, R. Astronomical computing: 1., computing under the open sky: 2., virtues of the haversine. **Sky and Telescope**, Cambridge, v. 68, p. 158-159, 1984.

TAOKA, G. Break reaction times of unalerted drivers. **ITE Journal Institute of Transportation Engineers**, Washington, v. 59, n. 3, p. 19-21, 1989.

TOROYAN, T. **Global status report on road safety: time for action**. Geneva: World Health Organization, 2009. 288 p.

TSUGAWA, S. Inter-vehicle communications and their applications to intelligent vehicles: an overview. In: INTELLIGENT VEHICLE SYMPOSIUM, 4., 2002, Versailles. **Proceedings...** Versailles: IEEE, 2002. v. 2, p. 564-569.

UNO, A. **Arduino Uno**. Disponível em: <<http://arduino.cc/en/Main/ArduinoBoardUno>>. Acesso em: 20 jan. 2014.

WITTE, T.; WILSON, A. Accuracy of non-differential gps for the determination of speed over ground. **Journal of Biomechanics**, New York, v. 37, n. 12, p. 1891-1898, 2004.

XU, B. et al. High-accuracy tdoa-based localization without time synchronization. **IEEE Transactions on Parallel and Distributed Systems**, Montreal, v. 24, n. 8, p. 1567-1576, 2012.

XU, Q. et al. Vehicle-to-vehicle safety messaging in dsrc. In: ACM INTERNATIONAL WORKSHOP ON VEHICULAR AD HOC NETWORKS, 1., 2004, Philadelphia. **Proceedings...** Philadelphia: ACM, 2004. p. 19-28.

ZEIMPEKIS, V.; GIAGLIS, G.; LEKAKOS, G. A taxonomy of indoor and outdoor positioning techniques for mobile location services. **ACM SIGecom Exchanges**, Cambridge, v. 3, n. 4, p. 19-27, 2002.