

FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA
ROCHA" CENTRO UNIVERSITÁRIO EURÍPIDES DE
MARÍLIA - UNIVEM CURSO DE BACHARELADO EM
CIÊNCIA DA COMPUTAÇÃO

RENAN AVANSI MARQUES

**ANÁLISE COMPARATIVA SOBRE UMA SIMULAÇÃO DE TRÁFEGO REAL E
DADOS DE PREDIÇÃO DO GOOGLE MAPS**

MARÍLIA
2015

RENAN AVANSI MARQUES

ANÁLISE COMPARATIVA SOBRE UMA SIMULAÇÃO DE TRÁFEGO REAL E
DADOS DE PREDIÇÃO DO GOOGLE MAPS

Trabalho de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Fundação de Ensino "Eurípedes Soares da Rocha", mantenedora do Centro Universitário Eurípedes de Marília - UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador
Prof. Dr. Fábio Dacêncio Pereira

MARÍLIA
2015



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
MANTIDO PELA FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Renan Avansi Marques

Título: Rastreamento de Alvos em uma Rede Veicular Altamente Dinâmica.

Banca examinadora da monografia apresentada ao Curso de Bacharelado em
Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de
Bacharel em Ciência da Computação.

Nota: 7,0 (note)

Orientador: Fábio Dacêncio Pereira 

1º. Examinador: Rodolfo Barros Chiamonte 

2º. Examinador: Leonardo Castro Botega 

Marília, 01 de dezembro de 2015.

AGRADECIMENTOS

Faço aqui meus agradecimentos a todos os professores do Centro Universitário Eurípides Soares da Rocha, em especial ao Prof. Dr. Fábio Dacêncio Pereira, pelo apoio, motivação e orientação.

Agradeço também aos meus familiares por estarem comigo em todos os momentos.

Meus agradecimentos aos meus amigos que estiveram presentes em minha formação, em especial a Fabiana, pelo apoio.

RESUMO

Vehicular ad hoc network (VANET) consiste em uma rede veicular composta por carros, motos, caminhões, e outros veículos automotores, tratados como nós para criar uma rede móvel e assim transmitir pacotes entre seus nós. Uma VANET transforma todo veículo relacionado na rede em um roteador wireless ou um nó. Mas como estes nós permanecem em movimento, dados enviados para a infraestrutura, relacionados ao posicionamento não serão os reais, pois a infraestrutura toma tempo para processar a informação e reenviar para o veículo, além de que deve-se considerar também o tempo de transmissão da informação na rede. O intuito do presente projeto é simular tráfego real com a ferramenta SUMO, comparar com o dado real e com ferramentas disponíveis online como o Google Maps.

Palavras-chave: *Vehicular ad hoc network* (VANET), simulação real do tráfego, SUMO.

ABSTRACT

Vehicular ad hoc network (VANET) is a vehicular network containing cars, motorcycles, trucks, and other vehicles, treated as nodes to create a mobile network and send packets among their nodes. VANET transforms every vehicle related in a network into a wireless router or a node. But, as these nodes keep moving, data sent to an infrastructure, position related won't be the real ones, because the infrastructure takes time to process and send back all information to the vehicle, besides that, it also has to consider the network transmission time. The project's intention is to simulate a real traffic with SUMO framework, and compare to the real data and online tools as Google Maps.

Keywords: Vehicular ad hoc network (VANET), real traffic simulation, SUMO.

SUMÁRIO

INTRODUÇÃO.....	10
OBJETIVOS GERAIS	10
METODOLOGIA.....	11
ORGANIZAÇÃO DESTE DOCUMENTO	11
1 CONTEXTUALIZAÇÃO	12
2 TOPOLOGIA ALTAMENTE DINÂMICA	14
2.1 Protocolo de Roteamento Baseado em Topologia	14
2.2 Vantagens	15
2.3 Segurança	16
3 FERRAMENTAS	18
3.1 SUMO	18
3.2 TraCI	19
3.3 OMNeT++	19
3.4 VEINS	21
4 GUIA DE CONFIGURAÇÃO - CONFIGURANDO O AMBIENTE	24
4.1 SUMO	25
4.2 OMNeT++	26
4.3 VEINS	26
5 SIMULAÇÃO DE TRÁFEGO REAL	28
5.1 Simulação	31
5.2 Teste e validação	33
5.3 Resultados	33
CONCLUSÃO.....	37
REFERÊNCIAS BIBLIOGRÁFICAS	38

LISTA DE ILUSTRAÇÕES

Figura 1 - Protocolos de roteamento	13
Figura 2 - <i>Sybil Attack</i>	16
Figura 3 - Veins	22
Figura 4 – Estrutura	23
Figura 5 - Mapa de Marília no SUMO	28
Figura 6 - Trajeto	29
Figura 7 - Exemplo lógica dos semáforos	31
Figura 8 - Semáforos no sumo-gui	32
Figura 9 - Exemplo de vType	32

LISTA DE TABELAS

Tabela 1 - Pontos 1 e 8	29
Tabela 2 - Pontos 2 e 7	30
Tabela 3 - Pontos 3 e 6	30
Tabela 4 - Pontos 4 e 5	30
Tabela 5 - Tempo da simulação.....	33
Tabela 6 - Tempo médio do teste	33
Tabela 7 - Tempo médio do Google Maps	33
Tabela 8 - Comparativo de tempo entre simulação e teste	35
Tabela 9 - Diferença de tempo entre teste e simulação	36

LISTA DE ABREVIATURAS E SIGLAS

AODV	Ad hoc On-Demand Vector
DoS	Denial of Service
DSRC	Dedicated Short-Range Communication
GPS	Global Positioning System
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IP	Internet Protocol
IPv6	Internet Protocol Version 6
IRC	Inter Road Communication
ITS	Intelligent Transport Systems
IVC	Inter Vehicle Communication
MAC	Media Access Control
MANET	Mobile ad hoc Network
MiXiM	Mixed Simulator
MPLS	Multiprotocol Label Switching
NED	Network Description Language
OMNeT++	Objective Modular Network Testbed in C++
OSPF	Open Shortest Path First
PHY	Physical
PPP	Protocolo Ponto a Ponto
SCTP	Stream Control Transmission Protocol
RSU	Road Side Unit
RVC	Road Vehicle Communication
SUMO	Simulation of Urban Mobility
TCP	Transmission Control Protocol
TraCI	Traffic Control Interface
UDP	User Datagram Protocol
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
VANET	Vehicular ad hoc Network
VEINS	Vehicles in Network Simulation
WAVE	Wireless Access in Vehicular Environments

INTRODUÇÃO

VANET (*Vehicular ad hoc Network*) é uma área emergente no meio acadêmico, definida como uma rede *ad hoc* com menor infraestrutura, autoconfiguráveis e nós móveis, onde o veículo contém tecnologias para auxílio de posicionamento, e até mesmo condicionamento do veículo.

Com o avanço da tecnologia, é possível implementar diversas redes em vários ambientes, uma delas sendo *Vehicular Ad hoc Network* (VANET), um conceito dentro de *Mobile Ad hoc Network* (MANET), que são redes definidas como livres, com menor infraestrutura, autoconfiguráveis e nós móveis. VANETs distinguem-se de MANETs em alguns pontos, como na topologia altamente dinâmica, devido ao baixo tempo de conexão, pois os veículos trafegam em diferentes velocidades e direções, mobilidade restrita, pois geralmente seus nós movem em rodovias ou ruas, alta taxa de conexão e desconexão na rede, pelos veículos afastarem-se uns dos outros ou com poucos pontos [MENDONÇA, 2012].

Pesquisas relacionadas à VANET aumentaram consideravelmente nos últimos anos, pois essas redes têm como objetivo melhorar a segurança de rodovias, veículos e eficiência no tráfego [ZEADALLY et al, 2010]. Consequentemente ao aumento do número de carros nas rodovias, cresce também o número de acidentes no trânsito.

Controle de tráfego, analisar condições de estradas, evitar acidentes e analisar e evitar congestionamentos são vantagens de obter uma rede VANET instalada veículos e rodovias. Além de que VANETs são utilizados na localização veicular em caso de roubo, mapeamento do local e até na descoberta de vagas em estacionamentos.

OBJETIVOS GERAIS

O presente projeto tem como objetivo geral a simulação de tráfego real para verificação do posicionamento de um veículo, por meio de simulações, testes e validações de métodos de *vehicular ad hoc network*, também sobre topologia altamente dinâmica *ad hoc* com tecnologia wireless. Visando completar o objetivo geral, é necessário um estudo nas tecnologias utilizadas para o desenvolvimento. As tecnologias escolhidas para simulação e validação do projeto foram o *Simulation of Urban Mobility* (SUMO) para simular o tráfego, pois é capaz de computar a aceleração e posicionamento do veículo, e OMNeT++ como *framework* para desenvolver a simulação da rede.

METODOLOGIA

O projeto foi dividido em quatro fases principais: (i) pesquisa e análise sobre VANETs e redes altamente dinâmicas, (ii) pesquisa e análise sobre trabalhos correlatos, (iii) desenvolvimento da simulação e testes, e (iv) avaliação dos resultados obtidos.

i. Pesquisa e análise sobre VANETs e redes altamente dinâmicas

Pesquisa em questão para maior fonte de conhecimento sobre o tema.

ii. Pesquisa e análise sobre trabalhos correlatos

Trabalho necessário para verificar no meio acadêmico para o melhor desenvolvimento do projeto possível.

iii. Desenvolvimento da simulação e testes

Desenvolver, simular, testar e validar o projeto. O desenvolvimento e simulação foram feitos no *framework* SUMO, onde foi possível simular o tráfego como os dados coletados no trajeto. As tecnologias escolhidas para simulação e validação do projeto foram o *Simulation of Urban Mobility* (SUMO) para simular o tráfego, pois é capaz de computar a aceleração e posicionamento do veículo.

iv. Avaliação dos resultados obtidos

Avaliar o desempenho e eficiência da simulação em comparação com o que o Google Maps fornece, e dados reais coletados.

ORGANIZAÇÃO DESTE DOCUMENTO

O documento apresentado está dividido em cinco capítulos. No primeiro capítulo é tratado uma breve contextualização sobre redes veiculares. O segundo capítulo faz uma revisão do meio acadêmico sobre topologia altamente dinâmica. No terceiro capítulo é abordada a fundamentação teórica e revisão sobre as tecnologias escolhidas para simulação e validação do projeto. No quarto capítulo contém um guia de instalação e configuração das ferramentas, o *Simulation of Urban Mobility* (SUMO) e o *framework* OMNeT++, necessários para elaboração do projeto. O quinto capítulo está dividido em apresentação da estrutura de rede simulada, testes e exposição dos resultados obtidos.

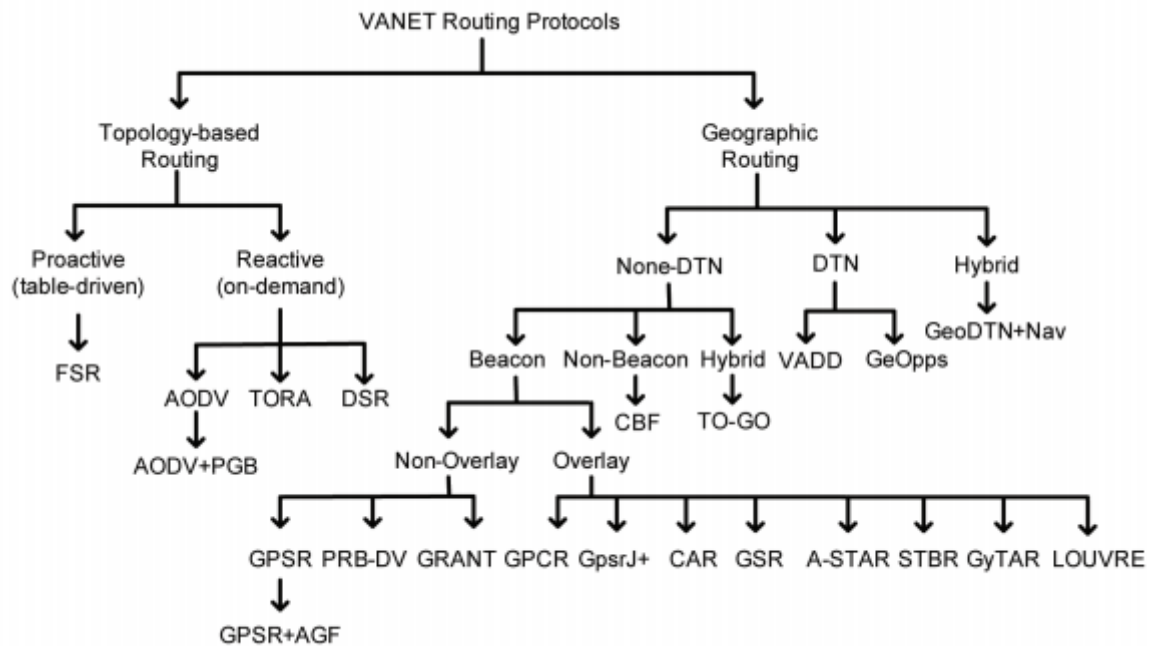
1 CONTEXTUALIZAÇÃO

Com a padronização recente do IEEE 802.11p WAVE (*Wireless Access in Vehicular Environments*) em 2010, este padrão de comunicação do IEEE é uma extensão da rede *Ethernet* local, porém *wireless* entre dois ou mais pontos. Segundo Barros, este padrão “define a camada física (PHY) e o controle de acesso ao meio (MAC) para VANETs, baseado em redes locais sem fio padrão IEEE 802.11a, que opera em uma faixa de frequência perto da alocada para comunicações de veículos, como também com uma taxa de transmissão compatível” [BARROS et al, 2012]. WAVE foi criado para ter um excelente desempenho em VANETs e também para ter suporte com *intelligent transport systems* (ITS). “O WAVE é construído sobre o DSRC americano, dividindo o espectro de frequência em sete canais de 10MHz cada. Um desses canais é usado apenas para gerenciamento da rede e serve para troca de mensagens de controle da rede” [MENDONÇA, 2012]. As redes VANETs garantem que carros comuniquem-se remotamente [KATUKA e SHAFIE, 2014].

A comunicação pode ocorrer entre veículos (*Inter Vehicle Communication* - IVC), entre veículo e infraestrutura (*Road Vehicle Communication* - RVC) e entre infraestrutura (*Inter Road Communication* - IRC), que são conhecidos como *Road Side Unit* (RSU). Os posicionamentos de infraestruturas são fixos e previamente conhecidos. O mapeamento geográfico de veículos em VANET é de baixo custo facilitado, pois as pessoas carregam celulares com GPS, ou mesmo nos próprios veículos e, também, geralmente os veículos trafegam em rodovias, e não em qualquer ponto da cidade.

Existem duas classes básicas de protocolos de roteamento de pacotes, o protocolo de roteamento baseado na topologia, e o protocolo de roteamento geográfico como mostrado na figura 1. No protocolo de roteamento geográfico, pode ser usado *beacon* para mapear os vizinhos e, então, eles decidem para onde redirecionar o pacote. E, no roteamento baseado na topologia, o protocolo AODV (*Ad hoc On-Demand Vector*) tem objetivo de atuar em ambientes de alta mobilidade, para diminuir o processamento da rede. Se um nó neste protocolo deseja enviar um pacote para um destino que não consta em sua tabela, este nó de origem envia uma requisição de rota aos seus vizinhos, e esses vizinhos distribuem a requisição até encontrar o nó destino, ou a rota do destino e, assim, o melhor caminho é mandado como resposta a origem [KAUR e VERMA, 2012].

Figura 1 - Protocolos de roteamento



Fonte: Lee et al, 2010.

Com a alta mobilidade em nós da rede por moverem-se em longas distâncias em um período curto de tempo causa o armazenamento de informações desatualizadas consequentemente tem influência na eficiência o processamento na infraestrutura [ZAKI et al, 2012]. Os veículos são representados por nós na rede e, como estes nós permanecem em movimento e por diversas vezes, em alta velocidade, dados mandados para a infraestrutura, relacionados ao posicionamento, provavelmente não serão os reais, pois a infraestrutura necessita de tempo para processar a informação e responder para o veículo.

Será apresentado um comparativo do tempo percorrido em um trajeto na cidade de Marília. Este comparativo foi desenvolvido entre uma simulação baseada em dados reais no SUMO, os dados de localização e previsão de tempo do trajeto disponibilizados pelo Google Maps, e o uso de um veículo para o teste do itinerário propriamente dito.

2 TOPOLOGIA ALTAMENTE DINÂMICA

Diferentemente de MANETs, onde os nós não se movem rapidamente, VANETs tem algumas características próprias, como a topologia altamente dinâmica, movendo seus nós em alta velocidade consequentemente com frequentes quebras de *links*. Em VANETs não é necessário limitar-se em energia como em redes MANETs, pois a rede é muito bem energizada sem a preocupação de economia energética. VANET é baseado em topologia, posicionamento e direção dos veículos e redes.

Constantes mudanças de velocidade e localização atrapalham a comunicação entre nós na rede. Veículos precisam trocar mensagens e informações para estabelecerem uma conexão. Por isso, em VANETs é necessário a utilização de uma topologia altamente dinâmica para que seus nós se auto-organizem na rede e troquem mensagens rapidamente antes de desconectar. Um veículo na rede não se limita apenas em adquirir informações, mas também é servido como um roteador, retransmitindo pacotes pela rede.

Protocolo de roteamento vem da necessidade da comunicação eficiente que supra a característica de alta quantidade de quebra de *links* em um curto período na rede, devendo transmitir os dados da origem para o destino. Há dois tipos básicos de protocolos de roteamento em VANETs, roteamento baseado em topologia e roteamento geográfico. O “Roteamento baseado em topologia utiliza a informações sobre *links* que existem na rede para repassar pacotes.” [Lee et al, 2010]. Roteamento geográfico utiliza o posicionamento dos seus vizinhos para repassar pacotes.

2.1 Protocolo de Roteamento Baseado em Topologia

Protocolo de roteamento baseado em topologia é subdividido em proativo e reativo. Proativos são *table-driven* e reativos são dependente da demanda, diminuindo o congestionamento na rede. O modo que esse protocolo opera é por meio de *flooding*, enviando a mensagem recebida para todos os outros nós da rede, com exceção do nó em que originou a chegada da mensagem.

Em roteamento baseado em posição geográfica, temos o “... *beaconing*, uma técnica em que nós vizinhos são inseridos em uma tabela, com suas posições geográficas e um tempo que define o *timestamp* do último *beacon* recebido. Quando um sinal de *beacon* ultrapassa um dado valor, o nó vizinho é excluído da tabela.” [Mendonça].

Em topologia altamente dinâmica, como os veículos trafegam em altas velocidades, a topologia formada muda constantemente. Como, em rodovias, a velocidade é maior que na área urbana, o tempo de comunicação V2V ou V2I é muito curto. Além do mais, os links sempre quebram e a rede necessita de uma auto-organização.

“Um protocolo de roteamento robusto precisa reconhecer a frequente desconexão e dispor rapidamente um link alternativo para garantir uma comunicação ininterrupta” [Lee et al, 2010] “Devido às forças para padronização de protocolos de roteamento em vários sistemas, a segurança dos dados trafegados nas VANETs é deixada em segundo plano.” [Mendonça]

2.2 Vantagens

VANET auxilia em segurança do veículo e motorista, bem como outros usuários por meio de alertas nos veículos que estão trafegando em área que o locomotor não enxergue, por exemplo, em um dos vários pontos cegos existentes no carro ou, em outro exemplo, uma rodovia com três veículos trafegando na mesma faixa e direção, sem o auxílio desta rede, na suposição de o primeiro veículo frear abruptamente, o segundo veículo começa a frear, para então o terceiro veículo tomar uma reação, podendo causar algum acidente. Com a rede implantada existe a possibilidade de, quando o primeiro veículo frear, a rede envia a informação para os outros dois carros, para que sensores alertem o motorista, e tome uma atitude rapidamente, evitando a colisão.

Outro ponto importante na segurança do veículo é a manutenção, pois com o implante de sensores, é possível uma constante verificação do condicionamento do veículo.

Gerenciamento de tráfego também é um quesito peculiar em redes veiculares, devido ao fornecimento de informações para a rede como localização dos veículos, velocidade e direção. Como consequência por conhecer essas informações, congestionamentos são minimizados, número de acidentes é reduzido, pelo controle de tráfego e decisões em tempo real de rotas para trocas de percursos.

Além de segurança, VANETs garante acesso à internet, visualização de vídeos, redes sociais, e-mail, jogos, notícias locais, auxílio na navegação, uso de mapas, informações sobre a rodovia e ruas da cidade, tudo para um maior conforto do usuário.

Com a comunicação entre veículos podem-se ter avisos de colisões, informações de tráfego em tempo real e até posicionamento após roubo do veículo.

2.3 Segurança

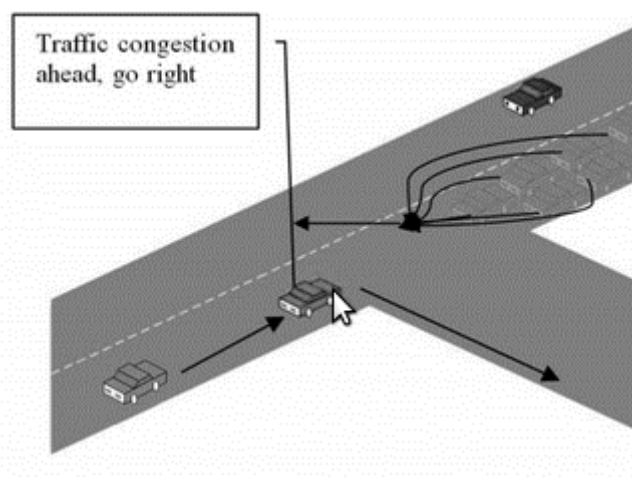
Há outros problemas nas redes VANETs que não são encontrados em outras redes, pelo fato da topologia altamente dinâmica e outros aspectos. Devido à alta mobilidade, a segurança em VANETs sofre alguns problemas e meios de segurança tradicionais podem não ser recomendados. Segurança em redes VANETs é pouco utilizada, porém de grande valia, pois há informações sensíveis nas redes que não devem ser roubadas, como dados pessoais e registros do veículo.

A transmissão da rede via *wireless*, a torna vulnerável, pois os quadros contendo os segmentos de dados estão no ar, assim, uma pessoa má intencionada pode coletar informações, como dados pessoais do motorista, rotas da viagem, velocidade, ou até mesmo injetar informações falsas na rede.

Um tipo de ataque na rede é o do tipo DoS (*Denial of Service*), onde um nó envia um grande número de mensagens falsas para congestionar a rede ou evitar que informações importantes cheguem ao destino [Verma et al, 2013].

Sybil Attack ocorre quando um nó na rede cria vários outros nós que não existem na rede para simular que há um congestionamento e forçar outros nós a utilizarem rotas alternativas. Com essa ação, qualquer outro tipo de ataque pode ser feito após mascarando o posicionamento ou IP desses nós na rede [Rahbari e Jamali].

Figura 2 – *Sybil Attack*



Fonte: Samara et al, 2010

Por que alguém tomaria vantagem sobre a rede VANET? Um motorista pode enviar informações falsas na rede para notificar outros motoristas que há um congestionamento em

sua rodovia, assim tomando outros caminhos, para que o tráfego em sua rota diminua [Samara et al, 2010]. Outra situação é quando um usuário mal-intencionado desvia um carro de sua rota original apenas por diversão e reconhecimento.

3 FERRAMENTAS

Neste capítulo é abordado a fundamentação teórica e revisão sobre as tecnologias escolhidas para simulação e validação do projeto, o *Simulation of Urban Mobility* (SUMO), e o framework OMNeT++, assim como, o *framework* VEINS, necessários para elaboração do projeto. Essas são as ferramentas utilizadas para a elaboração deste trabalho de conclusão de curso. Também será apresentado neste capítulo um guia de instalação dessas ferramentas na distribuição Linux Ubuntu 12.04.

3.1 SUMO

Simulation of Urban MOBility (SUMO) é um simulador microscópico de tráfego, rápido, portátil e *open source* desde 2001, desenvolvido pelo Instituto de Sistemas de Transporte no Centro Aeroespacial Alemão, na linguagem C++. Segundo o site oficial do SUMO, “Simulações de tráfego facilitam a avaliação das trocas de infraestrutura, bem como política de trocas antes da implementação deles nas estradas”. Este software pode ser usado em qualquer sistema operacional e realiza a modelagem da mobilidade de veículos microscopicamente. Isto significa que cada veículo na simulação é modelado explicitamente e tem sua própria rota. Ele permite a modelagem de sistema de tráfego, incluindo veículos, transportes públicos e pedestres.

Outra funcionalidade do software é que os veículos têm suas próprias características, como aceleração, desaceleração e tamanho, e varia entre os tipos dos veículos. São características do SUMO:

- Movimento veicular - onde a posição de cada veículo no cenário é atualizada a todo o momento;
- Simulação microscópica - veículos, pedestres e transportes públicos são explicitamente modelados;
- Diferentes tipos de veículos com suas determinadas características, como carros, caminhões e vans;
- Definição de rotas;
- Tempo dos semáforos podem ser importados ou gerados automaticamente pelo SUMO;
- Visualização de rotas;

- Sem limitações no tamanho da rede e número de veículos simulados.
- Importação de rede;
- Formatos de importação suportados: OpenStreetMap, VISUM, VISSIM, NavTeq.
- Cálculo de emissão;
- Sumo-gui: responsável pela interface gráfica do usuário na simulação, habilitando o usuário a observar a simulação em ação. Permite mostrar as propriedades relacionadas aos veículos e redes a serem modificadas. Também é possível habilitar para seguir um determinado veículo na simulação. [Phule, 2012]

O modelo do driver no SUMO é sofisticado e transforma em uma simulação de alta velocidade, porém em menores detalhes. Por causa da sua alta portabilidade, SUMO tem se tornado o simulador de tráfego para comunicação veicular mais utilizado. [Khandelwal, et al, 2012]

Seguem alguns dos componentes em que o pacote SUMO contém:

- SUMO: simulação por linha de comando;
- GUISSIM: simulação com interface gráfica do usuário;
- NETCONVERT: Importador de rede;
- NETGEN: Gerador de redes abstratas.

3.2 TraCI

Traffic Control Interface (TraCI), que utiliza a saída da simulação no SUMO, usa a arquitetura TCP cliente/servidor para ter acesso ao SUMO. Parte do projeto VEINS, TraCI é um módulo de gerenciamento de cenário no OMNeT++ que foi colocado como parte do *framework* MiXiM. Com este framework é possível agrupar o OMNeT++ e o SUMO para realizar um intermédio entre a simulação de rede no OMNeT++ e a simulação de tráfego no SUMO. [Phule, 2012].

3.3 OMNeT++

OMNeT++ não é um simulador por si só, mas ele fornece a infraestrutura e ferramentas para descrever simulações. É um framework de simulação de redes orientado a objeto, baseado

em C++, usado para simular vários ambientes, dentre eles, tráfego de veículos. OMNeT++ foi desenvolvido na Universidade Técnica de Budapest, Departamento de Telecomunicações e está disponível para o uso acadêmico. O software fornece a infraestrutura e as ferramentas para desenvolver as simulações.

O modelo de simulação no OMNeT++ consiste em módulos reusáveis que, bem escritos, podem ser de grande valia para serem combinados e reutilizados, e estes módulos são conectados para transmitir mensagens entre eles. Módulos são conectados uns aos outros por “*gates*”, ou portas, e podem ser combinados para formarem módulos maiores. Módulos passam mensagens, em que podem conter dados, por caminhos predefinidos por *gates*, ou diretamente para o destino. Simulações com OMNeT++ são em interfaces distintas, como Tkenv, que é a interface gráfica do usuário, e o Cmdenv, que é a linha de comando.

Existem vários frameworks para simular redes em OMNeT++, porém, há dois principais, que são INET, usado para protocolos de camada, e MiXiM, usado para simulação PHY e MAC. No framework INET, contém modelos para protocolos de redes cabeadas e wireless, como UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, 802.11, MPLS, OSPF. Por outro lado, no MiXiM, há a propagação de ondas de rádio, estimação de interferência, consumo de energia de transceptor de rádio e protocolo MAC.

OMNeT++ fornece as ferramentas necessárias para definir a estrutura do sistema, tais como:

- Comunicação entre módulos;
- *Network Description language* (NED) usados para definir a topologia da rede;
- Meios de alterar os arquivos NED por texto ou interface gráfica.

Segundo Phule, um modelo de simulação consiste de:

- Arquivos .ned, que descrevem a posição e conexões entre os módulos, e também os valores dos parâmetros dos módulos;
- Arquivos .msg, que contém as mensagens de definição de campos dos dados que são transformados de OMNeT++ para classes C++;
- Arquivos C++;
- Simulação do kernel usado para gerenciar a simulação e suas classes;
- Arquivo .ini que é usado para especificar parâmetros modificáveis para os módulos envolvidos. [Phule, 2012]

Segundo seu site oficial, existem algumas características do OMNeT++, como por exemplo, todos os eventos em OMNeT são empacotados em mensagens, seus módulos são em C++, configurações são em arquivos NED.

Essas são algumas aplicabilidades do OMNeT++, segundo o seu site, usada para vários problemas, tais como:

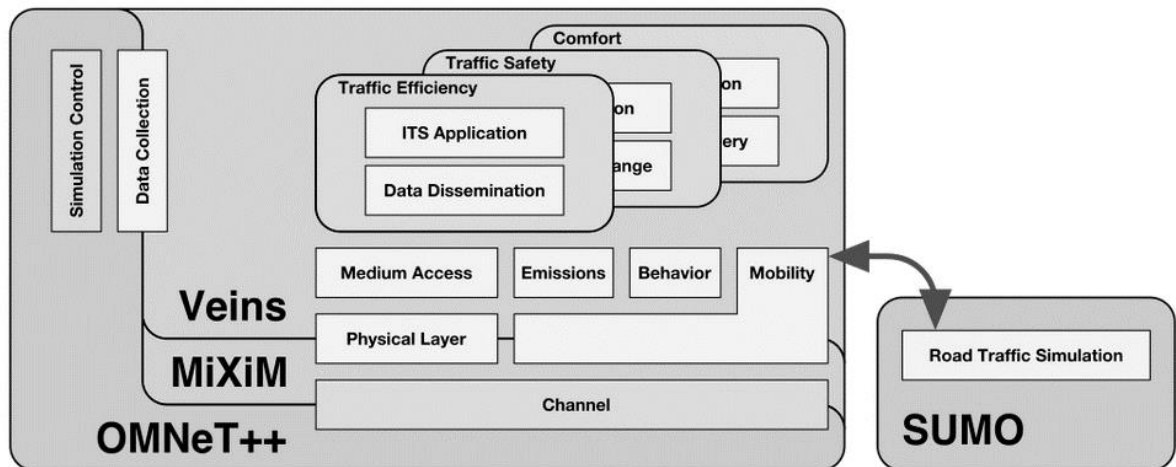
- Modelagem de comunicação de redes cabeadas e wireless;
- Modelagem de protocolos;
- Modelagem de redes de fila;
- Modelagem de multiprocessadores e outros sistemas distribuídos;
- Validação de arquiteturas de hardware;
- Avaliação de desempenho de complexos sistemas de softwares.

3.4 VEINS

Vehicles in Network Simulation (VEINS) é um *framework open source* de simulação de comunicação entre veículos, composto de simulação de redes baseado em eventos e um modelo de microsimulação de tráfego de rodovias. Modelos de domínios específicos para redes veiculares são construídas para fornecer um framework de fácil uso e aprendizado.

VEINS é o resultado do Veins Research Project, que foi desenvolvido no Departamento de Sistemas de Comunicações e Redes de Computadores, Universidade de Erlangen, Alemanha. O propósito do projeto era um simulador de comunicações entre veículos realísticos. Veins é feito para agrupar simuladores distintos, OMNeT++ para simulação de redes e SUMO para simulação de tráfego de rodovias. Houve a necessidade de agrupar o tráfego e a simulação de rede bidirecional para simulações realísticas das aplicações VANET no SUMO, então, esse framework foi criado para agrupar o simulador de redes OMNeT++ e o simulador de tráfego SUMO. *Traffic Control Interface* (TraCI) serviu para ligar os dois simuladores, realizando agrupamento bidirecional [Phule, 2012]. A seguir é ilustrado a arquitetura e interação do Veins com OMNeT++ e SUMO.

Figura 3 - Veins



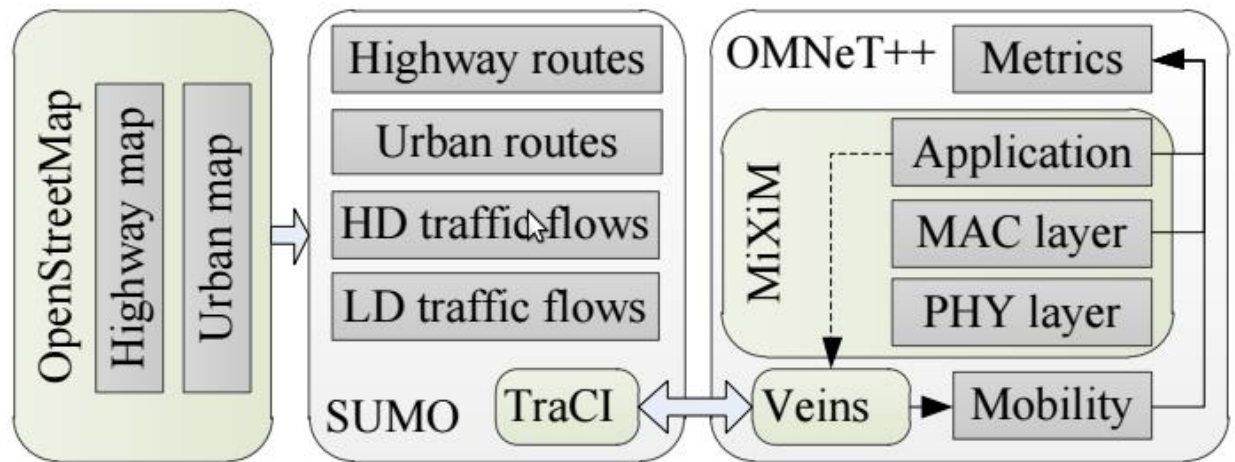
Fonte: Veins, 2015

Para realizar avaliações de comunicação entre veículos, ambos simuladores devem rodar em paralelo, conectados por um soquete TCP. O protocolo para essa comunicação foi padronizado como TraCI. Ele permite simulação bidirecional de tráfego de rodovias e tráfego de rede. Movimento de veículos nas simulações de tráfego de rodovias no SUMO é refletido no movimento de nós na simulação do OMNeT++. Nós podem, então, interagir com a simulação de tráfego que está em execução.

Simulação de tráfego de rodovias é realizada pelo SUMO, que é bem estabelecido no domínio de engenharia de tráfego. Simulação de rede é realizada pelo OMNeT++, ao lado de uma modelagem de camada física, como o toolkit MiXiM. “A maioria dos modelos de simulação de tráfego utilizam um modelo de movimentação conhecido como “*car-following-model*”, ou em tradução livre, modelo de perseguição veicular, em que o comportamento do motorista é dependente da distância e da velocidade do veículo à frente”. [MÜHLBAUER, 2014].

Na figura 4 pode ser visto a arquitetura e integração do SUMO, Veins, OMNeT++ e TraCI.

Figura 4 - Estrutura



Fonte: Booyesen, 2012

4 GUIA DE CONFIGURAÇÃO – CONFIGURANDO O AMBIENTE

A seguir, será detalhado um breve guia de instalação das ferramentas acima citadas e descritas no sistema operacional utilizado para o desenvolvimento deste trabalho, o Ubuntu 12.04 LTS que, por questões de compatibilidade com as versões de OMNeT++, SUMO e VEINS, foi escolhido.

Neste passo, será configurado o sistema com os pacotes necessários previamente no sistema para a instalação das ferramentas. Antes de instalar, é necessário utilizar o comando *update* para atualizar o banco de dados dos pacotes disponíveis.

```
$ sudo apt-get update
```

Esses são os pacotes que, devem assim, ser instalados pelo terminal do linux:

```
$ sudo apt-get install build-essential gcc g++ bison flex perl tcl-dev tk-dev libxml2-dev  
zlib1g-dev default-jre doxygen graphviz libwebkitgtk-1.0-0 openmpi-bin libopenmpi-dev  
libpcap-dev
```

E também por intermédio do Ubuntu Software Center:

- build-essential
- gcc
- g++
- bison
- flex
- perl
- tcl-dev
- tk-dev
- libxml2-dev
- zlib1g-dev
- default-jre
- doxygen
- graphviz
- libwebkitgtk-1.0-0
- openmpi-bin
- libopenmpi-dev
- libpcap-dev

4.1 SUMO

A versão do SUMO que foi usada neste trabalho, é a 0.21, pois é compatível com a versão 3.0 do VEINS e a versão 4.4 do OMNeT++. O arquivo do SUMO pode ser encontrado em seu site oficial, http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/.

Para que o SUMO possa ser instalado com êxito, é necessária a instalação de algumas dependências do SUMO, e eles são: Xerces-C, Proj, GDAL e Fox. Antes, verifique se a sua distribuição do Linux vem esses pacotes instalados. Caso não estejam, instale-os com os seguintes comandos:

```
$ sudo apt-get install autoconf
```

```
$ sudo apt-get install proj libtool libgdal1-dev libxerces-c2-dev libfox-1.6-0 libfox-1.6-dev
```

Agora, extraia o .tar.gz, mova-o para a pasta /usr/local/src/ e navegue para esta pasta:

```
$ tar -xzf sumo-src-0.21.0.tar.gz
```

```
$ sudo mv -v sumo-0.21.0 /usr/local/src
```

Após esses comandos, navegue até a pasta lib:

```
$ cd /usr/lib
```

```
$ sudo ln -s libgdal1.7.0.so libgdal.so /usr/local/src/sumo-0.21.0
```

```
$ cd /usr/local/src/sumo-0.21.0
```

Use o comando “configure” com alguns parâmetros, e após, make:

```
$ sudo ./configure --with-fox-includes=/usr/include/fox-1.6 --with-gdal-includes=/usr/include/gdal --with-proj-libraries=/usr --with-gdal-libraries=/usr --with-proj-gdal
```

```
$ sudo make
```

Navegue até a pasta bin do sumo e adicione na variável de ambiente:

```
$ cd bin
```

```
$ sudo mv activitygen emissionsDrivingCycle netconvert polyconvert TraCITestClient dfrouter emissionsMap netgenerate sumo duarouter jtrrouter od2trips sumo-gui /usr/local/bin
```

4.2 OMNeT++

A versão do OMNeT++ utilizada nesse projeto será a omnetpp-4.4. Baixe esta versão do OMNeT++ no site <http://omnetpp.org>. Procure pelas versões antigas e baixe o arquivo OMNeTpp-4.4-src.tgz. Depois de concluído o *download*, aplique o seguinte comando para descompactar:

```
$ tar -xvf omnetpp-4.4-src.tgz
```

Com isso, será extraído para a pasta. Para configurar o OMNeT++ e adicionar o diretório bin para o PATH temporariamente, navegue para a pasta do OMNeT++:

```
$ cd omnetpp-4.4
```

```
$ . setenv
```

No diretório extraído, execute o comando configure:

```
$ ./configure
```

Esse comando verifica as dependências e as configurações do sistema e também cria um *make file*, necessário para o próximo passo. Após a finalização do comando anterior, digite no terminal:

```
$ make
```

E a instalação do OMNeT++ está concluída.

4.3 VEINS

A versão do VEINS utilizada será a 3.0, que está disponível em seu site oficial, onde é encontrado no endereço <http://veins.car2x.org/>. Depois de baixado, extraia-o. Importe o projeto extraído em seu *workspace* do OMNeT++ em *File > Import > General: Existing Projects into Workspace* e selecione a pasta em que foi extraído o framework. Use o build do projeto importado com *Project > Build All* no OMNeT++. Navegue até o diretório do VEINS.

```
$ cd ../veins-3.0/examples/veins
```

E após, execute o seguinte comando para abrir um arquivo do exemplo:

```
$ sumo-gui -c erlangen.sumo.cfg
```

Para não precisar executar o SUMO manualmente sempre que simular o OMNeT++, o framework Veins tem um script para isso, que pode ser executado com o seguinte comando:

```
$ ./sumo-launchd.py -vv -c sumo-gui
```

O script irá criar conexões TCP entre o OMNeT++ e o SUMO, iniciando uma cópia da simulação do SUMO para toda conexão de simulação do OMNeT++.

Agora, no OMNeT++, execute arquivo `omnetpp.ini`, que está na pasta do exemplo do Veins.

Após concluído, a simulação será executada, e você poderá observá-la. O que houve foi a execução do OMNeT++ e SUMO em paralelo para simular um fluxo de veículos sendo parados por um acidente.

Apesar de citados e abordados neste trabalho de conclusão de curso, as ferramentas OMNeT++ e TraCI não foram aplicadas, sendo aproveitado apenas o *framework* SUMO. Porém, tais ferramentas continuam descritas no texto para fonte de conhecimento.

5 SIMULAÇÃO DE TRÁFEGO REAL

Nesta seção será abordado a simulação do tráfego em um trajeto da cidade de Marília, desenvolvido através do SUMO, onde o trajeto será abordado a seguir.

Para o desenvolvimento desse projeto de conclusão de curso foi exportado o mapa de Marília através do openstreetmaps.org, que é um projeto *free* e colaborativo para o mapeamento de ruas, onde pode ser editado através de suas ferramentas, e que pode ser facilmente exportado através do browser, no formato *.osm*, selecionando a localidade e importando ao SUMO com o comando *netconvert*. A figura 5 ilustra o mapa importado no SUMO:

Figura 5 - Mapa de Marília no SUMO

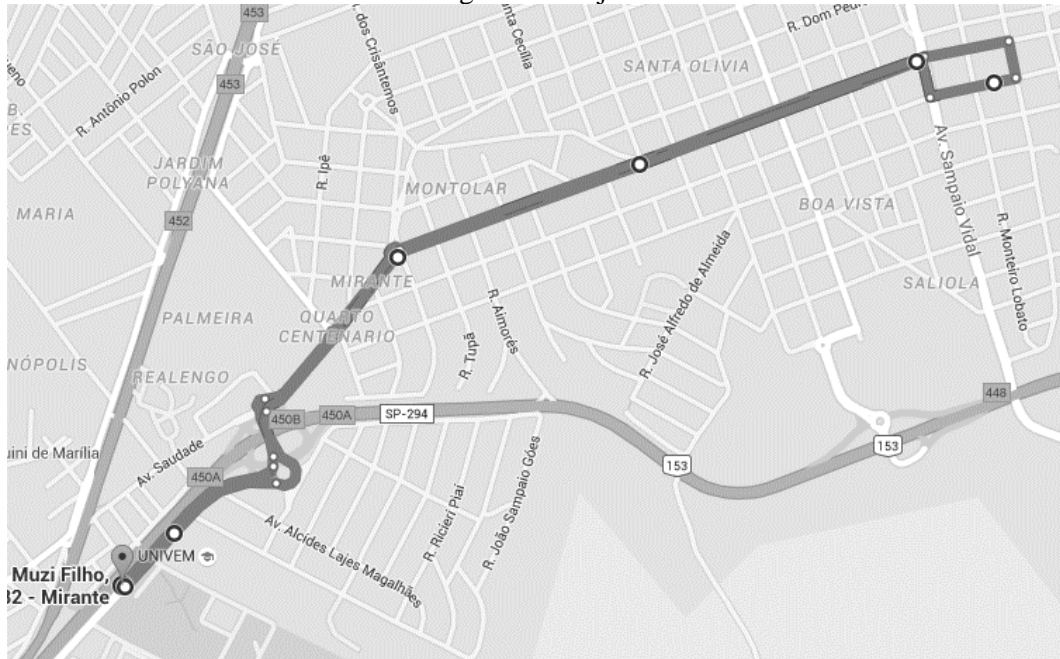


Fonte: Elaborado pelo autor, 2015

O trajeto para a simulação foi definido seguindo o fluxo principal saindo do Campus para o centro de Marília, evitando rodovias e passando pela Avenida Hygino Muzzi Filho, Avenida Saudade e Avenida Rio Branco, como visto na figura 6. Para a realização deste trabalho de conclusão de curso foi realizado uma tabulação do tráfego da cidade de Marília-SP. Como critério para a tabulação foi um dia sem chuva, para que os dados não fossem alterados pelas condições climáticas.

Foi selecionado 3 horários para retirar a amostragem dos dados: doze horas, quinze horas e dezoito horas. Doze e dezoito horas foram escolhidas, pois há um fluxo maior de veículos devido ao horário de almoço e saída do trabalho. Em contrapartida, também foi selecionado um horário de fluxo menor (quinze horas), para um comparativo.

Figura 6 - Trajeto



Fonte: Google Maps, 2015

Após decidido um trajeto, foi retirado amostragens sobre o tráfego real nas ruas, em determinados pontos do trajeto, onde a primeira tabulação foi extraída em frente à Fundação Eurípides Soares da Rocha, a segunda na rotatória que liga a Avenida Saudade com a Avenida Rio Branco, o terceiro e quarto pontos no meio e no fim da Avenida Rio Branco, respectivamente. Lembrando que foram observadas as tabulações nos dois sentidos das Avenidas. As tabulações foram arredondadas pelo método de aproximação e serão mostradas a seguir. As tabelas 1 a 4 mostram a média aritmética das tabulações dos três dias.

Tabela 1 - Pontos 1 e 8

	Ponto 1			Ponto 8		
	12:00	15:00	18:00	12:00	15:00	18:00
Carro	105	58	102	49	50	57
Moto	25	12	40	14	10	23
Ônibus	1	1	0	1	0	0
Circular	1	1	1	1	1	0
Caminhão	1	2	1	0	1	0
Van	2	1	1	2	2	0

Fonte - Elaborado pelo autor, 2015

Tabela 2 - Pontos 2 e 7

	Ponto 2			Ponto 7		
	12:00	15:00	18:00	12:00	15:00	18:00
Carro	117	114	128	81	83	120
Moto	47	33	45	22	21	37
Ônibus	1	0	0	0	0	2
Circular	2	1	1	1	1	1
Caminhão	1	2	1	1	2	1
Van	4	2	2	2	1	0

Fonte - Elaborado pelo autor, 2015

Tabela 3 - Pontos 3 e 6

	Ponto 3			Ponto 6		
	12:00	15:00	18:00	12:00	15:00	18:00
Carro	87	89	84	98	94	115
Moto	28	21	22	22	20	33
Ônibus	1	0	0	0	1	0
Circular	2	1	1	2	2	2
Caminhão	1	1	0	0	0	1
Van	1	2	0	1	3	1

Fonte - Elaborado pelo autor, 2015

Tabela 4 - Pontos 4 e 5

	Ponto 4			Ponto 5		
	12:00	15:00	18:00	12:00	15:00	18:00
Carro	84	77	80	67	73	96
Moto	27	14	18	21	8	24
Ônibus	0	0	0	0	1	0
Circular	2	3	3	1	2	2
Caminhão	1	0	0	0	0	0
Van	2	2	0	2	1	1

Fonte - Elaborado pelo autor, 2015

Como pode ser observado nas tabelas acima, o maior fluxo baseado no tipo de veículo é o de carro, que foi considerado como de porte médio, seguido de veículos de pequeno porte (moto) e por último, veículos de grande porte (caminhões e ônibus).

Com os dados em mãos, e com o auxílio da ferramenta SUMO, foi possível realizar a simulação do tráfego real da cidade de Marília. Foram feitas três simulações, uma para cada horário estipulado anteriormente. No SUMO há a possibilidade de implementar detectores para facilitar o controle de fluxo de veículos. Foi criada diversas rotas de acordo com a tabulação retirada do tráfego real, com vias mais movimentadas e *output* dos detectores.

5.1 Simulação

Para gerar o arquivo .net, que contém todas as conexões das ruas e rodovias, foi utilizado o netconvert passando como parâmetro o mapa gerado pelo Open Street Maps, que pode ser interpretado pelo SUMO. Diversas rotas foram traçadas a partir dos dados extraídos da tabulação. Também foi observado e implementado os comportamentos dos semáforos do trajeto principal, para uma simulação mais aproximada possível da realidade. A semaforização é obtida através da indicação na *tag* do arquivo .osm gerado pelo Open Street Maps antes da conversão pelo netconvert. Após a conversão, o semáforo é gerado e apenas é necessário configurar a lógica do comportamento compatível com a real, como visto na figura 7.

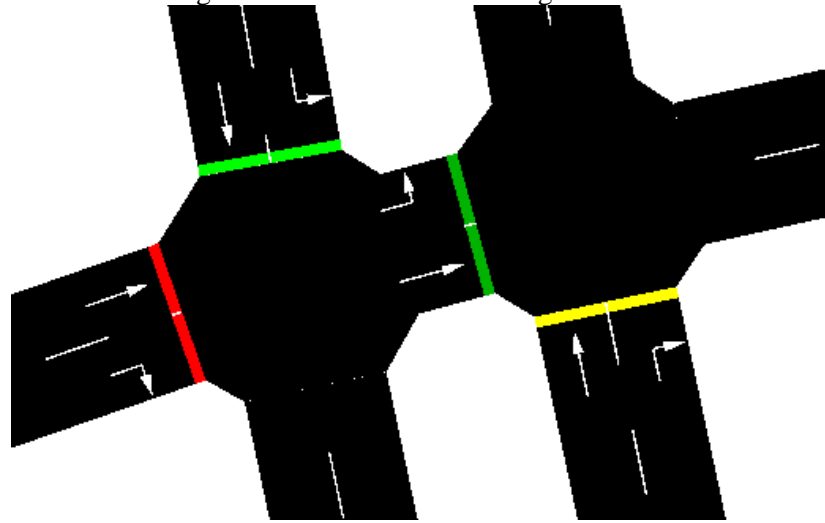
Figura 7 - Exemplo lógica dos semáforos

```
<tlLogic id="1420382467" type="static" programID="0" offset="0">
  <phase duration="35" state="GGGgggrrr"/>
  <phase duration="3" state="yyyyggrrr"/>
  <phase duration="1" state="rrrrggrrr"/>
  <phase duration="18" state="rrrrgggGGG"/>
  <phase duration="3" state="rrrrgggyyy"/>
  <phase duration="10" state="rrrrggrrr"/>
</tlLogic>
```

Fonte - Elaborado pelo autor, 2015

Onde “r” representa vermelho, “y” representa amarelo, “G” corresponde ao verde com preferência, e “g” é verde sem preferência. Verde sem preferência espera o veículo de preferência atravessar, demonstrados na figura 8 na interface gráfica do SUMO.

Figura 8 - Semáforos no sumo-gui



Fonte - Elaborado pelo autor, 2015

Com o auxílio dos detectores foi possível controlar o tráfego na simulação em que o resultado pode ser visto a seguir.

Para a implementação do tráfego no SUMO é necessário criar uma rota para o veículo, traçando os *ids* das rodovias por onde irá trafegar, definir o tipo veículo com a *tag* `<vType>`, definindo sua forma na visualização gráfica, aceleração, desaceleração, tamanho, espaço entre veículos, velocidade máxima e cor, como visto na figura 9.

Figura 9 - Exemplo de vType

```
<vType id="truck" accel="2.2" decel="3.0" sigma="0.5" length="12" minGap="2"
maxSpeed="16" guiShape="truck" color="1,0,0.75"/>

<vType id="passenger" length="4" minGap="1" guiShape="passenger"
osgFile="vw_golf.3ds" probability="20" color=".25,.5,1" accel="2.54" maxSpeed="18" /
>
```

Fonte - Elaborado pelo autor, 2015

Após criado os tipos de veículos que foram utilizados na simulação, que são carros, caminhões e motos, foram criados fluxos dos veículos para definir os parâmetros como tempo início e fim do fluxo, probabilidade e quantidade de veículos. Os resultados dos tempos de simulação serão apresentados na seção 5.3.

5.2 Teste e validação

Para testar e validar o projeto, foi realizado uma tabulação real com um carro trafegando pela rota estipulada na figura 6, contabilizando o tempo a partir do começo da rota até o seu fim, para um comparativo entre os resultados do teste real, os resultados deste projeto simulado na ferramenta SUMO, e os resultados dos dados disponíveis pelas tecnologias do Google Maps, acessíveis através do link www.googlemaps.com, resultados estes que foram extraídos nos mesmos horários.

5.3 Resultados

Nesta seção serão apresentados os resultados obtidos pela simulação do *Simulation of Urban MObility*, do teste fornecidos pelo Google Maps, e os dados do teste real com um veículo. Os resultados obtidos da simulação na ferramenta SUMO estão representados na tabela 5.

Tabela 5 - Tempo da simulação

	12:00	15:00	18:00
Tempo	10:03	11:44	12:00

Fonte - Elaborado pelo autor, 2015

A tabela da média da estipulação da localização real baseada no tempo foi retirada em dias sem chuva, nos horários determinados, e pode ser visto a seguir, na tabela 6.

Tabela 6 - Tempo médio do teste

	12:00	15:00	18:00
Tempo	13:06	15:51	18:23

Fonte - Elaborado pelo autor, 2015

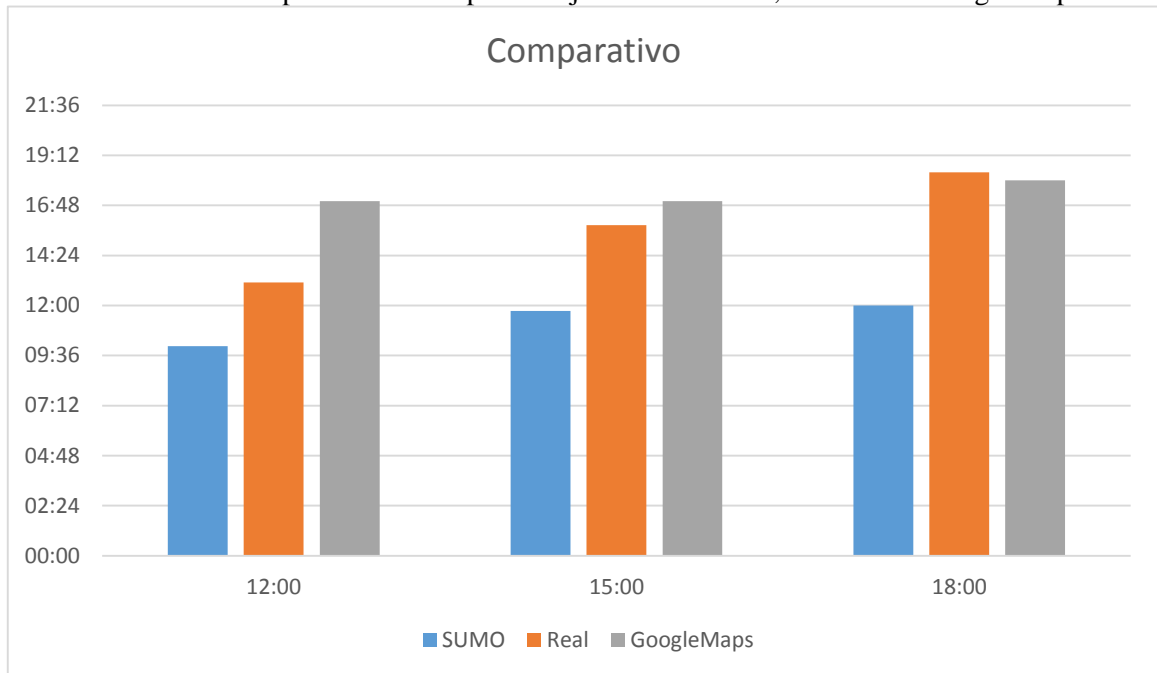
A próxima tabela expõe a média dos resultados obtidos através da ferramenta do Google Maps:

Tabela 7 - Tempo médio do Google Maps

	12:00	15:00	18:00
Tempo	17:00	17:00	18:00

Fonte - Elaborado pelo autor, 2015

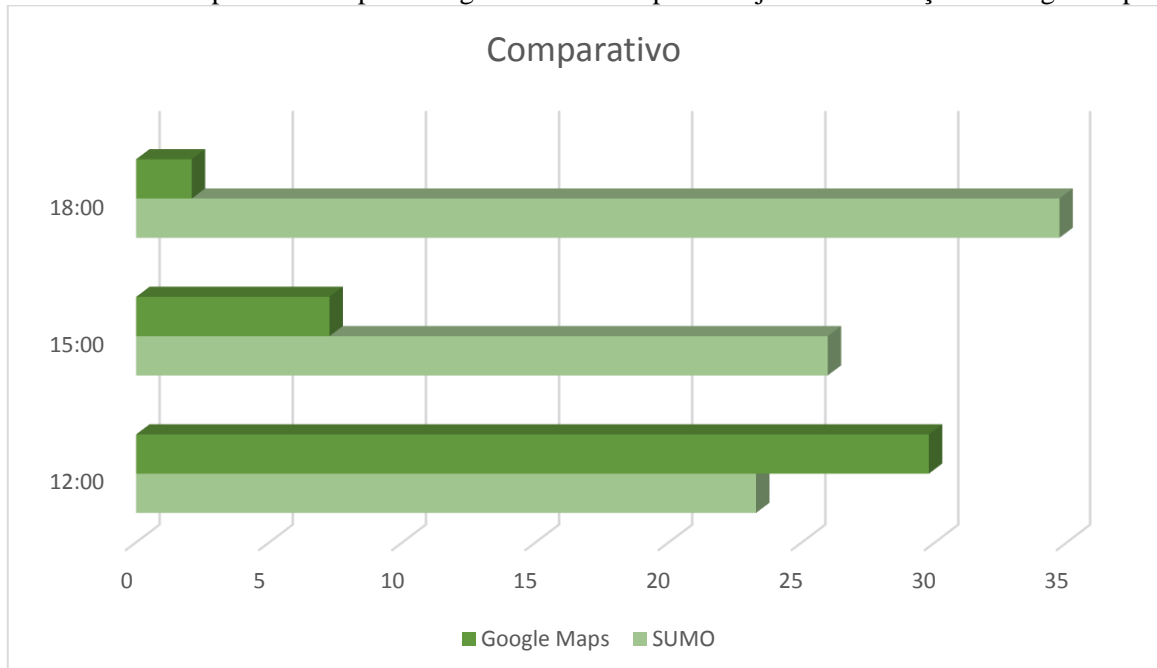
Gráfico 1 – Comparativo do tempo do trajeto entre SUMO, dado real e Google Maps



Fonte - Elaborado pelo autor, 2015

O gráfico 1 realça a superioridade do Maps, onde a simulação feita com o SUMO tem vantagem sobre o Google Maps apenas no horário das doze horas, pois seu tempo do trajeto foi mais próximo do dado real do que os resultados obtidos pela simulação, nos outros, o Google tem um valor muito próximo do real, enquanto a simulação está abaixo do obtido.

Gráfico 2 - Comparativo em porcentagens entre o tempo do trajeto da simulação e Google Maps



Fonte - Elaborado pelo autor, 2015

No gráfico 2 é mostrado o erro percentual da simulação e o Google Maps, comparados ao resultado real obtido, onde que, a simulação não pode ser descartada, pois obteve um valor próximo ao real, e levando em conta que o Google utiliza grande quantidade de dados e sensores. Basicamente, o Google Maps coleta dados por dois meios diferentes, históricos de dados por veículos que trafegaram pela rota, e dados enviados por sensores em tempo real por quem está trafegando na rota. Esse processo é feito anonimamente, e só é necessário ativar o Google Maps no smartphone, que os dados são coletados. Esses dados são disponibilizados para todos por meio de condições e tempo de tráfego no site do Google Maps [Barth].

A tabela 8 mostra a média da relação entre o tempo e o ponto do posicionamento do veículo.

Tabela 8 - Comparativo de tempo entre simulação e teste

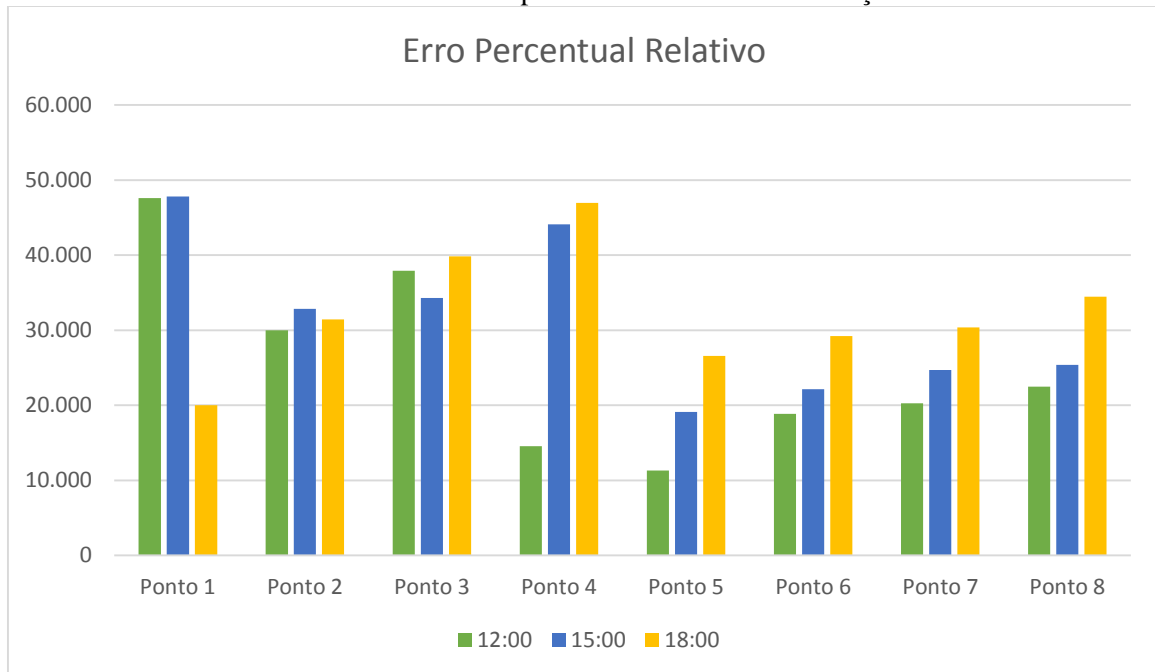
	Simulação			Teste		
	12:00	15:00	18:00	12:00	15:00	18:00
Ponto 1	0:11	0:12	0:16	0:21	0:23	0:20
Ponto 2	1:31	1:34	1:36	2:10	2:20	2:20
Ponto 3	2:11	2:14	2:16	3:31	3:24	3:46
Ponto 4	5:00	3:33	3:28	5:51	6:21	6:32
Ponto 5	6:55	8:19	8:28	7:48	10:17	11:32
Ponto 6	7:54	9:33	9:49	9:44	12:16	13:52
Ponto 7	8:44	10:22	10:37	10:57	13:46	15:15
Ponto 8	9:53	11:34	11:49	12:45	15:30	18:02

Fonte - Elaborado pelo autor, 2015

Pode-se notar nos resultados das simulações realizadas neste trabalho de conclusão de curso, que apesar de ocorrer um erro, ele não é demasiado, com exceção em alguns pontos às dezoito horas.

No gráfico 3 e tabela 9 a seguir, pode-se observar o erro percentual relativo dos dados coletados do teste real e simulação, e a Diferença de tempo entre teste e simulação, respectivamente.

Gráfico 3 - Erro percentual relativo da simulação



Fonte - Elaborado pelo autor, 2015

Tabela 9 - Diferença de tempo entre teste e simulação

	12:00	15:00	18:00
Ponto 1	0,16667	0,18333	0,06666
Ponto 2	0,65	0,76666	0,73333
Ponto 3	1,33334	1,16667	1,5
Ponto 4	0,85	2,8	3,06666
Ponto 5	0,88333	1,96666	3,06666
Ponto 6	1,83333	2,71667	4,05
Ponto 7	2,21667	3,4	4,63333
Ponto 8	2,86667	3,93333	6,21666

Fonte - Elaborado pelo autor, 2015

CONCLUSÃO

Com este projeto de conclusão de curso, e observando as tabelas de resultados e o gráfico disponibilizado, é notável que, apesar de que a simulação feita neste projeto faz uso dos dados do tráfego real no *framework* SUMO, como a simulação do tráfego, comportamento dos semáforos, velocidade máxima das rodovias, e condições dos veículos, o Google Maps aproxima-se mais dos dados reais obtidos do que a simulação no SUMO, mesmo que, no período de doze horas, seu valor seja menor que o esperado.

Para proposta de trabalhos futuros seria interessante analisar um trajeto maior ou com alta quantidade de pontos de tabulação para que a simulação seja ainda mais fiel em relação ao tráfego real. Também, poderia ser integrado os parâmetros do Google Maps, para alimentar a entrada de dados no SUMO, onde a simulação será feita com maior fidelidade, a precisão Google Maps será bem aproveitada, e unirá com a flexibilidade do SUMO em parametrizar praticamente todas as variáveis relacionados a simulação de trânsito.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBA, Carolina T.; MATEOS, Miguel A.; SOTO, Pablo R.; MEZHER, Ahmad M.; IGARTUA Mónica A. Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights. Disponível em: <http://sertel.upc.edu/~maguilar/papers_pdf/iv12.pdf>. Acesso em 5 de mar. 2015.

BARROS, M. T.; Gomes, R. C. M.; Costa, A. F. B. F. Routing Architecture for Vehicular Ad-Hoc Networks Disponível em: <https://www.academia.edu/2628331/_Routing_Architecture_for_Vehicular_Ad-Hoc_Networks>. Acesso em 28 de mar. 2015.

BARTH, Dave. The bright side of sitting in traffic: Crowdsourcing road congestion data. Disponível em: <<https://googleblog.blogspot.in/2009/08/bright-side-of-sitting-in-traffic.html>>. Acesso em 16 de nov. 2015.

BOOYSEN, Thinus. Tutorial: Simulating VANET and ITS (using OMNeT++ and SUMO). Disponível em: <<http://www.grc.upv.es/docencia/rwm/DOCS/VANETSsimulations>>. Acesso em 20 de Abril 2015.

FUENTES, Jose Maria de; GONZÁLES, Ana Isabel. Overview of security issues in Vehicular Ad-hoc Networks. Disponível em: <<http://e-archivo.uc3m.es/bitstream/handle/10016/9395/Overview-VANET-security-issues-earchivo.pdf?sequence=1>>. Acesso em 25 de ago. 2015.

HARTENSTEIN, Hannes. A Tutorial Survey on Vehicular Ad Hoc Networks. Disponível em: <<http://www.ijcic.org/ijcic-11-07016.pdf>>. Acesso em 28 de mar. 2015.

Institute of Transportation Systems. Disponível em: <http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/> Acesso em 20 de Abril 2015.

KATUKA, Jatau; SHAFIE, Muhammad. VANETs AND ITS RELATED ISSUES: AN EXTENSIVE SURVEY. Disponível em: <<http://www.jatit.org/volumes/Vol66No2/19Vol66No2.pdf>>. Acesso em 2 de mar. 2015

KAUR, Tajinder; Verma, A. K. Simulation and Analysis of AODV routing protocol in VANETs. Disponível em: <<http://www.ijscce.org/attachments/File/v2i3/C0750062312.pdf>>. Acesso em 29 de mar. 2015.

KHANDELWAL, Sumit A.; GULHANE, Vijay S. Public Survey the Simulation Tool of Vehicular Ad-hoc Networks. Disponível em: <<http://www.ijarcse.com/docs/papers/january2012/V2I1017.pdf>>. Acesso em 28 de Maio. 2015.

LEE, Kevin C.; Lee, Uichin; Gerla, Mario. Survey of Routing Protocols in Vehicular Ad Hoc Networks. Disponível em: <<http://www.irma-international.org/viewtitle/43169/>>. Acesso em 30 de mar. 2015.

MENDONÇA, Thiago F. VANETs: Vehicular Ad-Hoc Networks Disponível em: <http://grenoble.ime.usp.br/~gold/cursos/2012/movel/mono-1st/1105-1_ThiagoFurtado.pdf>. Acesso em 28 de mar. 2015.

NEHA; ISHA. Analysis of Hybrid GPSR and Location Based Routing Protocol in VANET. Disponível em: <<http://www.ijcsit.com/docs/Volume%206/vol6issue01/ijcsit20150601149.pdf>>. Acesso em 20 ago. 2015.

OMNeT++, User Manual. Disponível em: <<https://omnetpp.org/doc/omnetpp/manual/usman.html>> Acesso em 20 de Abril 2015.

PHULE, Harshad. Public Safety Application for Approaching Emergency Vehicle Alert and Accident Reporting in VANETs Using WAVE. Disponível em: <<http://scholarworks.rit.edu/cgi/viewcontent.cgi?article=4189&context=theses>>. Acesso em 6 de maio 2015.

QURESHI, Kashif Naseer; ADBULLAH, Abdul Hanan. Topology based routing protocols for VANET and their comparison with MANET. Disponível em: <<http://www.jatit.org/volumes/Vol58No3/27Vol58No3.pdf>>. Acesso em 19 de ago. 2015.

RAHBARI, Mina; JAMALI, Mohammad A. J. Efficient Detection of Sybil Attack Based on Cryptography in Vanet. Disponível em: <<http://arxiv.org/abs/1112.2257>>. Acesso em 26 de ago. 2015.

SAMARA, Ghassan; AL-SALIH, Waffa A. H.; SURES, R. Security Analysis of Vehicular Ad Hoc Networks (VANET). Disponível em: <<http://arxiv.org/ftp/arxiv/papers/1206/1206.1337.pdf>>. Acesso em 25 ago. 2015.

VERMA, Karan; HASBULLAH, Halabi; KUMAR, Ashok. Prevention of DoS Attacks in VANET. Disponível em: <http://www.researchgate.net/publication/257675979_Prevention_of_DoS_Attacks_in_VANET>. Acesso em 26 ago. 2015.

ZAKI, Salim M.; NGADI, M. A.; KAMAT, Maznah; RAZAK, Shukor, A.; SHARIFF, Johan M. A Location Based Prediction Service Protocol For VANET City Environment. Disponível em: <<http://www.ijicic.org/ijicic-11-07016.pdf>>. Acesso em 28 de mar. 2015.

ZEADALLY, Sherali; HUNT, Ray; CHEN, Yuh-Shyan; IRWIN, Angela; HASSAN, Aamir. Vehicular Ad Hoc Networks (VANETS): Status, Results, and Challenges. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.188.7416&rep=rep1&type=pdf>>. Acesso em 4 de mar. 2015.