

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Programa de Pós-Graduação em Informática

**CARTIM: UM PROTOCOLO DE IDENTIFICAÇÃO E MINIMIZAÇÃO
DE CONGESTIONAMENTOS VEICULARES**

Guilherme Braga Araújo

Belo Horizonte

2014

Guilherme Braga Araújo

**CARTIM: UM PROTOCOLO DE IDENTIFICAÇÃO E MINIMIZAÇÃO
DE CONGESTIONAMENTOS VEICULARES**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Informática.

Orientadora: Prof.^a Dr.^a Fátima de
Lima Procópio Duarte-
Figueiredo

Coorientador: Prof. Ph.D. Antonio Al-
fredo Ferreira Loureiro

Belo Horizonte

2014

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

A663c Araújo, Guilherme Braga
Cartim: um protocolo de identificação e minimização de congestionamentos
veiculares / Guilherme Braga Araújo. Belo Horizonte, 2014.
85f.: il.

Orientador: Fátima de Lima Procópio Duarte Figueiredo
Coorientador: Antonio Alfredo Ferreira Loureiro
Dissertação (Mestrado)- Pontifícia Universidade Católica de Minas Gerais.
Programa de Pós-Graduação em Informática.

1. Trânsito - Congestionamento. 2. Transporte urbano. 3. Trânsito urbano. 4.
Sistemas difusos. 5. Computação ubíqua. I. Figueiredo, Fátima de Lima
Procópio Duarte. II. Loureiro, Antonio Alfredo Ferreira. III. Pontifícia
Universidade Católica de Minas Gerais. Programa de Pós-Graduação em
Informática. IV. Título.

SIB PUC MINAS

CDU: 681.3.092

Guilherme Braga Araújo

**CARTIM: UM PROTOCOLO DE IDENTIFICAÇÃO E MINIMIZAÇÃO
DE CONGESTIONAMENTOS VEICULARES**

Dissertação apresentada ao Programa de
Pós-Graduação em Informática da Ponti-
fícia Universidade Católica de Minas Ge-
rais, como requisito parcial para obtenção
do título de Mestre em Informática.

Prof.^a Dr.^a Fátima de Lima Procópio
Duarte-Figueiredo (Orientadora) – PUC
Minas

Prof. Ph.D. Antonio Alfredo Ferreira
Loureiro (Coorientador) – UFMG

Prof. Ph.D. Markus Endler – PUC Rio

Prof. Dr. Henrique Cota de Freitas – PUC
Minas

Belo Horizonte, 30 de abril de 2014.

À minha família e àqueles que se interessem pelo tema.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus. Nenhum obstáculo é grande demais quando confiamos em Deus.

Agradeço à Mariana pelo carinho incondicional, companheirismo, dedicação e esforço em abrir mão de algumas realizações momentâneas, para realizarmos sonhos maiores no futuro. Essas ajudas foram imensuráveis.

Aos meus pais, José Célio e Alice, que sempre estiveram presentes nesse grande projeto, apoiando e ajudando de diversas formas. Agradeço também ao meu irmão pelas conversas e trocas de experiências neste período. A todos os meus familiares e amigos que me apoiaram, de alguma forma, também agradeço.

À minha orientadora Fátima que desde a graduação confiou no meu trabalho, meu muito obrigado. Sempre dedicada e com muitos comentários relevantes me ajudou e ensinou bastante. Considero um privilégio ter sido orientado por ela. Agradeço ao Loureiro pela orientação e à Anna Tostes (doutoranda UFMG), cujas ideias, opiniões e questionamentos contribuíram muito para este trabalho.

Agradeço ao grupo de pesquisa de Computação Móvel, pelas reuniões semanais, aprendizado e trocas de experiências que, direta ou indiretamente, contribuíram para a realização deste trabalho. Obrigado, em particular, aos mestrandos Matheus Queiroz, José Gláucio e Mauro Salomão.

À Pontifícia Universidade Católica de Minas Gerais e à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), agradeço pelo financiamento deste trabalho, que me possibilitou cumprir um regime de dedicação exclusiva.

Aos amigos e colegas do Mestrado em Informática da PUC Minas, obrigado pelas conversas, companhia e momentos de distração.

RESUMO

O congestionamento de trânsito é tratado pelos órgãos governamentais como problema de mobilidade urbana. Ele gera *stress* aos motoristas e perdas econômicas. Para auxiliar na identificação e na redução de congestionamentos no tráfego veicular, são propostos sistemas inteligentes de transporte. Este trabalho propõe o CARTIM, um protocolo de identificação colaborativa e minimização de congestionamentos veiculares. O protocolo se baseia na comunicação V2V (*Vehicle-to-Vehicle*) para mensurar cooperativamente o nível local de congestionamento de tráfego. Quando houver infraestrutura presente, pode ocorrer disseminação de informações consolidadas a veículos em outras regiões, através da comunicação V2I (*Vehicle-to-Infrastructure*). Para identificar localmente um congestionamento, de forma eficaz, o CARTIM possui um sistema baseado em lógica *fuzzy* (nebulosa), que faz o tratamento das informações qualitativas (densidade de veículos etc). O protocolo também utiliza eficientemente o canal colaborativo de comunicação, evitando sobrecarga. Os resultados obtidos pelo CARTIM mostram que o sistema *fuzzy* obteve boa eficácia para quantificar o nível local de congestionamento nos veículos em cenários heterogêneos (autoestrada e vias urbanas). A heurística proposta para alteração das rotas e minimização dos congestionamentos também obteve resultados satisfatórios nas simulações, reduzindo o tempo médio de viagem dos veículos. Assim, o CARTIM consegue detectar congestionamentos e minimizá-los.

Palavras-chave: VANET, Redes Veiculares, V2V, V2I, Congestionamento Veicular, Mobilidade Urbana, Identificação e Minimização de Congestionamentos, Lógica *Fuzzy*, Computação Ubíqua.

ABSTRACT

Traffic congestion is handled by government agencies as an urban mobility problem, which generates stress to drivers and economic loss. Intelligent transport systems can assist in the identification and reduction of vehicular traffic congestion. In this context, this paper proposes CARTIM, a protocol for collaborative identification and minimization of vehicular congestion. The protocol uses V2V (Vehicle-to-Vehicle) communication to cooperatively measure the local level of vehicular traffic congestion. Additionally, if any infrastructure is present, consolidated information dissemination may occur to vehicles in other regions through V2I (Vehicle-to-Infrastructure) communication. To effectively identify a traffic congestion locally, CARTIM employs a fuzzy logic-based system, which is used in the treatment of qualitative information (vehicle density etc). The protocol also efficiently uses the collaborative communication channel, preventing overload. The results obtained by CARTIM shows that the fuzzy system obtained good efficacy for quantifying the local level of congestion in the vehicles on heterogeneous scenarios (freeway and urban streets). The proposed congestion minimization heuristic also obtained satisfactory results in simulations, reducing the travel average time of vehicles. Thus, CARTIM can detect congestion and minimize it.

Keywords: VANET, Vehicular Networks, V2V, V2I, Vehicular Traffic Congestion, Urban Mobility, Identifying and Minimizing Congestion, Fuzzy Logic, Ubiquitous Computing.

LISTA DE FIGURAS

FIGURA 1 – Arquiteturas das redes veiculares	22
FIGURA 2 – A pilha de protocolos WAVE.....	23
FIGURA 3 – Taxonomia de tecnologias de comunicação sem fio	23
FIGURA 4 – Modelos de mobilidade aplicados à movimentação veicular	26
FIGURA 5 – Classificação para simuladores de mobilidade	27
FIGURA 6 – Arquitetura do VEINS	29
FIGURA 7 – Arquitetura do iTETRIS	29
FIGURA 8 – Arquitetura do TraNS.....	30
FIGURA 9 – SUMO - Simulação de tráfego rodoviário	31
FIGURA 10 – Interface WEB do OpenStreetMap.....	32
FIGURA 11 – Mecanismo de lógica <i>fuzzy</i>	34
FIGURA 12 – Fluxograma das etapas da metodologia proposta.....	45
FIGURA 13 – Ambiente usado para simulações de VANETs.....	46
FIGURA 14 – Estrutura do CARTIM	48
FIGURA 15 – Funções de pertinência dos conjuntos <i>fuzzy</i> de entrada.....	51
FIGURA 16 – Funções de pertinência para congestionamento veicular	51
FIGURA 17 – Nível de congestionamento obtido pelo sistema	52
FIGURA 18 – Tela do qtfuzzylite com sistema <i>fuzzy</i> configurado para CARTIM ...	53
FIGURA 19 – Disseminação de mensagens CTE em um cenário de exemplo	54
FIGURA 20 – CTDiss, processo de retransmissão de mensagens <i>unicast</i>	56
FIGURA 21 – Disseminação de mensagens CTA em um cenário de exemplo	56
FIGURA 22 – Diagrama de atividades - Controle de mensagens na rede	57
FIGURA 23 – Redes rodoviárias simuladas	61

FIGURA 24 – Nível de congestionamento no Cenário <i>A</i> versus tempo de simulação .	63
FIGURA 25 – Nível de congestionamento no Cenário <i>B</i> versus tempo de simulação .	63
FIGURA 26 – Congestionamento geral por cenário versus tempo de simulação	66
FIGURA 27 – Mapa de calor com a densidade de veículos por região do mapa	67
FIGURA 28 – Tempo médio de viagem versus cenário	67
FIGURA 29 – Distância média de viagem versus cenário	68
FIGURA 30 – Emissão de CO ₂ versus cenário	68
FIGURA 31 – Mapas da região central de Belo Horizonte, MG	70

LISTA DE QUADROS

QUADRO 1 – Sumário dos <i>frameworks</i> para simulações de VANETs	28
QUADRO 2 – Propostas de monitoramento de tráfego veicular avaliadas	37
QUADRO 3 – Principais parâmetros de simulação	46
QUADRO 4 – Regras <i>fuzzy</i> para o CARTIM	50
QUADRO 5 – CARTIM, parâmetros de simulação	61
QUADRO 6 – Cenários de simulação	64

LISTA DE TABELAS

TABELA 1 – Frota de Veículos no Brasil	18
TABELA 2 – Nível de congestionamento proposto pela Skycomp.....	52
TABELA 3 – Total de mensagens CTE no Cenário <i>B</i>	63
TABELA 4 – Mensagens de congestionamento por cenário	65
TABELA 5 – Contagem volumétrica de veículos (CVV) em Belo Horizonte, MG ...	69
TABELA 6 – Número médio de veículos por faixa por via.....	69
TABELA 7 – Desempenho do protocolo CARTIM no cenário real	71

LISTA DE ABREVIATURAS E SIGLAS

3G Terceira Geração

4G Quarta Geração

ACK *Acknowledgement*

BH Belo Horizonte

BHTRANS Empresa de Transportes e Trânsito de Belo Horizonte

DENATRAN Departamento Nacional de Trânsito

DSRC *Dedicated Short Range Communications*

CARTIM *Cooperative Vehicular Traffic Congestion Identification and Minimization*

C_{th} *Congestion threshold*

CO₂ Gás Carbônico

COC *Content Oriented Communications*

COTEC *COoperative Traffic congestion detECTION*

CT_{cl} *Cooperative Traffic congestion level*

CTA *Cooperative Traffic Alert*

CTA_{th} *Cooperative Traffic Alert threshold*

CTDiss *Congestion Traffic Dissemination*

CTE *Cooperative Traffic Congestion Detection*

CVV Contagem Volumétrica de Veículos

DTMon *Dynamic Traffic Monitoring System*

FCC *Federal Communications Commission*

FCL *Fuzzy Control Language*

GUI *Graphical User Interface*

GIS *Geographic Information System*

GPS *Global Position System*

HCM *Highway Capacity Manual*

Hz *Hertz*

iCS *iTETRIS Control System*

iTETRIS *Integrates Wireless Communications and Road Traffic Simulation Platforms*

ITS *Intelligent Transportation Systems*

IVC *Inter Vehicle Communication*

JOSM *Java Open Street Map editor*

MAC *Media Access Control*

MANET *Mobile Ad Hoc Network*

RSU *Road Side Unit*

SUMO *Simulation of Urban Mobility*

T_{cl} *Traffic congestion level*

TMC *Traffic Management Center*

TraCI *Traffic Control Interface*

TraNS *Traffic and Network Simulation Environment*

V2I *Vehicle to Infrastructure*

V2V *Vehicle to Vehicle*

VANET *Vehicular Ad hoc Network*

VEINS *Vehicles in Network Simulation*

VOTING *Vehicular Over The Air Information Gathering*

WAVE *Wireless Access Vehicular Environment*

ZOR *Zone Of Relevance*

SUMÁRIO

1	INTRODUÇÃO.....	17
1.1	Motivação e Justificativa	18
1.2	Especificação do Problema.....	18
1.3	Objetivos	19
1.4	Trabalho Realizado e Contribuições	19
1.5	Organização do texto.....	19
2	REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS.....	21
2.1	Redes Veiculares.....	21
2.1.1	<i>WAVE</i>	22
2.1.2	<i>Simulação em Redes Veiculares</i>	23
2.1.3	<i>Rede de Tráfego Rodoviário</i>	30
2.2	Lógica <i>Fuzzy</i>	32
2.3	Trabalhos Relacionados	35
3	METODOLOGIA DE PESQUISA.....	44
4	CARTIM	47
4.1	Introdução	47
4.2	Estimativa Local de Tráfego Veicular	49
4.3	Validação Cooperativa de Congestionamento de Tráfego	53
4.4	Heurística para Minimização de Congestionamento de Tráfego	55
4.5	Gerenciamento e Manipulação de Mensagens na Rede	57
5	SIMULAÇÕES, AVALIAÇÃO E RESULTADOS DO PROTOCOLO..	61
5.1	Comparação do Sistema <i>Fuzzy</i>	62
5.2	Resultados do CARTIM no Manhattan <i>Grid</i>	64
5.2.1	<i>Mensagens de Congestionamento de Tráfego</i>	64
5.2.2	<i>Sobrecarga de Comunicação</i>	65

5.2.3	<i>Tamanho do Congestionamento</i>	65
5.2.4	<i>Tempo de Viagem, Distância Percorrida e Emissão de CO₂</i> ...	67
5.3	Resultados do CARTIM no Mapa Geográfico	68
6	CONCLUSÕES E TRABALHOS FUTUROS	72
6.1	Conclusões	72
6.2	Trabalhos Futuros	73
	REFERÊNCIAS	75
	APÊNDICE A – CÓDIGO SISTEMA <i>FUZZY</i> CARTIM	81
	APÊNDICE B – PUBLICAÇÕES	85

1 INTRODUÇÃO

O transporte sempre desempenhou um papel de destaque na história da humanidade. Durante as últimas décadas, a densidade de veículos no transporte rodoviário no Brasil tem aumentado consideravelmente (DENATRAN, 2014). Esse aumento deu origem ao fato de que, atualmente, os congestionamentos de tráfego nas vias urbanas e estradas são um dos fenômenos mais comuns que os motoristas têm que enfrentar. Além de ser uma experiência estressante e desagradável para motoristas, os congestionamentos de tráfego também causam um impacto negativo na economia e no meio ambiente.

Cotidianamente, os usuários de uma rede veicular trafegam de casa para o trabalho, esperando chegar aos seus destinos dentro de um tempo estimado. Com esse propósito, tomam decisões individuais sobre quais rotas seguirem, sem nenhuma informação atualizada do tráfego, o que não auxilia no uso balanceado das vias. Entretanto, um estado mais balanceado é de interesse coletivo, já que tende a reduzir os tempos médios de viagem, beneficiando o sistema como um todo (AMARANTE; BAZZAN, 2012).

Hoje em dia, grandes avanços tecnológicos vêm sendo aprimorados e embarcados em veículos automotores, para melhorar a experiência dos motoristas. Por exemplo, podem ser citados sistemas de frenagem, alarmes de velocidade acima do permitido etc (ALVES et al., 2009). O crescimento populacional e, conseqüentemente, da frota de veículos, torna necessárias redes que conectem esse tipo de ambiente. Veículos com conectividade com outros veículos, compartilhando o mesmo meio (as vias) podem auxiliar motoristas, a fim de proporcionar viagens mais rápidas até seus destinos.

Redes veiculares inspiram vários trabalhos, com propostas de novos serviços e aplicações ITS (*Intelligent Transportation Systems*), que proporcionam não só gerenciamento de tráfego, mas também maior segurança rodoviária, acesso à informação e entretenimento (ALVES et al., 2009). Para contribuir com a solução de parte desses problemas mencionados, a comunicação entre veículos é utilizada, e tal ambiente é conhecido como WAVE (*Wireless Access in Vehicular Environments*) (KARAGIANNIS et al., 2011).

Neste contexto, uma aplicação ITS para identificação e redução de congestionamento pode beneficiar órgãos públicos gestores do tráfego, motoristas, usuários do transporte público e ambientalistas. Os motoristas e usuários do transporte são os principais beneficiados, uma vez que consideram o tempo passado no trânsito como tempo perdido.

1.1 Motivação e Justificativa

Em um documento do Governo Federal Brasileiro (SEMOB – Secretaria Nacional do Transporte e da Mobilidade Urbana, 2007), através do Ministério das Cidades, é apresentada a necessidade de reverter o modelo de mobilidade utilizado em grandes cidades brasileiras, que vivem um momento de crise da mobilidade urbana. Essa crise exige mudanças de paradigma, talvez de forma mais radical do que outras políticas setoriais. As necessidades descritas no documento se tornaram ainda mais críticas atualmente.

O DENATRAN (Departamento Nacional de Trânsito) disponibiliza dados referentes à frota de veículos no Brasil. A Tabela 1 mostra a quantidade de veículos no Brasil entre 2007 e setembro de 2013. Observa-se que, nesse intervalo de tempo, ocorreu um aumento expressivo de aproximadamente 62% na frota de veículos no país. Isso é atribuído, principalmente, aos incentivos fiscais para venda de veículos automotores, concedidos pelo Governo, e, ainda, à deficiência do sistema de transporte público nas cidades brasileiras e grande demanda por mobilidade. As redes veiculares e aplicações ITS podem auxiliar e minimizar problemas de congestionamento, cada vez mais frequentes, contribuindo para a mobilidade urbana em cidades inteligentes.

Na literatura, há algumas propostas para detecção de congestionamento em redes veiculares. Este trabalho se baseia em uma delas, estendendo-a para identificar e minimizar congestionamentos em autoestradas e vias urbanas. A proposta de minimização de congestionamento procura reduzir o tempo de viagem e, conseqüentemente, o *stress* das pessoas e o nível de poluição causado pela emissão de gás carbônico (CO₂) pelos veículos.

Tabela 1 – Frota de Veículos no Brasil

Ano	Número de Veículos
2007	49.644.025
2008	54.506.661
2009	59.361.642
2010	64.817.974
2011	70.543.535
2012	76.137.191
2013	80.179.368

Fonte: Dados do DENATRAN.

1.2 Especificação do Problema

Este trabalho pretende responder à seguinte pergunta: como identificar e minimizar congestionamentos de tráfego utilizando redes veiculares? Esse problema pode ser dividido nas seguintes etapas: (i) identificar um congestionamento de tráfego, (ii) conseguir uma decisão colaborativa do congestionamento estimado, (iii) repassar informações da situação atual do tráfego e (iv) utilizar políticas de alteração de rotas para reduzir o congestionamento.

1.3 Objetivos

A partir disso, o objetivo deste trabalho foi propor um protocolo, o CARTIM (do inglês **C**ooperative vehicul**A**R Traffic congestion **I**dentification and **M**inimization), para detecção e redução de congestionamento em redes veiculares, utilizando o padrão IEEE 802.11p (KARAGIANNIS et al., 2011), (BILGIN; GUNGOR, 2013), para troca de informações, em tempo real, da situação atual do tráfego. O CARTIM foi projetado para ser viável em cenários heterogêneos (autoestrada e vias urbanas), que fazem parte do escopo de redes veiculares, e, por isso, um sistema *fuzzy* para tratamento das informações qualitativas (e.g. velocidade lenta, densidade veicular alta etc) e identificação dos níveis de congestionamentos foi utilizado.

A proposta teve como objetivos específicos buscar reduzir o tempo médio de viagem dos veículos, disseminar informações do tráfego na rede veicular, permitir alterações das rotas através de políticas adequadas e melhorar o fluxo do tráfego de veículos nas vias.

1.4 Trabalho Realizado e Contribuições

As principais contribuições desta dissertação foram a concepção e criação de um protocolo para aplicações ITS de gerenciamento de tráfego veicular.

CARTIM: é uma solução que se apoia na comunicação entre veículos para identificar e reduzir congestionamentos veiculares em autoestradas e vias urbanas. Uma característica que pode ser observada nas aplicações de redes veiculares é a demanda por contextos (exemplos: tipo de estrada, velocidade máxima permitida etc). Assim, o protocolo proposto neste trabalho supera tal demanda, se adaptando aos cenários ou contextos, utilizando um sistema *fuzzy* dinâmico para identificar congestionamentos de tráfego. A técnica CTDiss (*Congestion Traffic Dissemination*), para disseminação de informações de tráfego congestionado na rede veicular, também é uma contribuição deste trabalho. O algoritmo para minimização de congestionamento teve como propósito permitir alterações de rotas, sob condições críticas de tráfego nas vias (baseado em um processo colaborativo definido para CARTIM), buscando auxiliar motoristas na tomada de decisão sobre quais rotas adotarem. A finalidade dessa política é ajudar no uso balanceado das vias, já que, dessa forma, tende-se a reduzir os tempos médios de viagem, beneficiando o sistema como um todo.

1.5 Organização do texto

Este trabalho está organizado em 6 capítulos. O Capítulo 2 contextualiza redes veiculares, simulação em redes veiculares e lógica *fuzzy*. Além disso, é apresentada uma

revisão bibliográfica dos principais trabalhos relacionados. O Capítulo 3 apresenta a metodologia proposta e define o ambiente de simulação. Detalhes do protocolo proposto neste trabalho são discutidos no Capítulo 4. A definição dos cenários, a avaliação e os resultados obtidos nas simulações são abordados no Capítulo 5. Por fim, o Capítulo 6 conclui este trabalho, apresentando os comentários finais e potenciais trabalhos futuros.

2 REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

2.1 Redes Veiculares

Redes veiculares, também chamadas de VANETs (*Vehicular Ad hoc NETWORKS*), são redes formadas por veículos e por equipamentos fixos ou unidades de acostamento (*Road Side Units* - RSUs) localizados às margens das estradas. Essas redes se caracterizam pela alta mobilidade dos nós (veículos), enlaces intermitentes e requisitos estritos de latência. São esses atributos que impedem que alguns protocolos utilizados em redes *ad hoc* (CORDEIRO; AGRAWAL, 2011) sejam utilizados em redes veiculares, pois eles não apresentam desempenho satisfatório (ALVES et al., 2009).

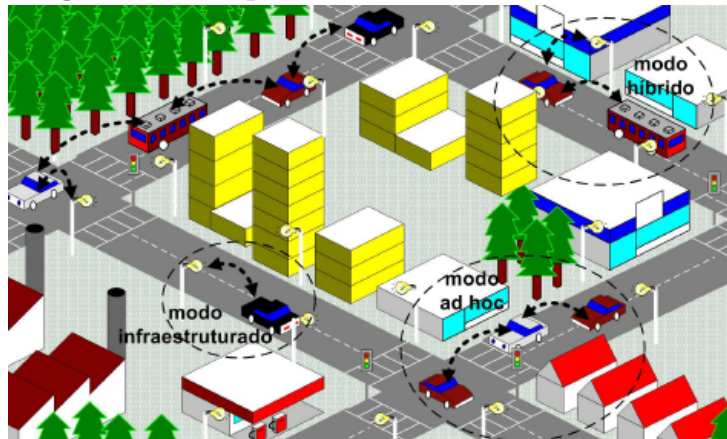
As VANETs, comparadas a outras redes *ad hoc*, possuem algumas peculiaridades, dentre elas as constantes mudanças em sua topologia, provocadas pela alta mobilidade dos nós, o que acarreta desconexões frequentes. No entanto, em virtude dos nós que formam VANETs serem veículos, eles possuem um padrão de mobilidade particular. Uma característica do padrão de mobilidade das VANETs está relacionado com a velocidade dos nós, uma vez que velocidades altas permitem uma janela de comunicação entre veículos muito curta. Outra característica importante e diferente de outras redes *ad hoc*, tais como redes de sensores sem fio (CORDEIRO; AGRAWAL, 2011), é o fato de, nas redes veiculares, a fonte de energia ser considerada "ilimitada", pois durante o período que os veículos estão em movimento, as baterias automotivas são recarregadas. Devido a essa última característica, a capacidade de processamento dos nós nas redes veiculares não é considerada um fator limitante como nas redes de sensores.

A arquitetura das redes veiculares determina a forma como os nós da rede se organizam e se comunicam. As arquiteturas definidas para essas redes são: (i) *ad hoc* puro, (ii) infraestruturada ou (iii) híbrida (ALVES et al., 2009). Na arquitetura (i), também chamada de *Vehicle-to-Vehicle* (V2V) *communication*, os veículos se comunicam sem nenhum suporte externo ou elemento centralizador. No entanto, os veículos funcionam como roteadores e encaminham as mensagens através de múltiplos *hops*. A comunicação V2V tem como vantagem seu baixo custo de implantação e baixa latência de comunicação, mas as desvantagens são o fato da conectividade da rede depender da densidade de veículos vizinhos e do padrão de mobilidade dos veículos. Dessa forma, para evitar problemas de conectividade, a arquitetura (ii) também conhecida como *Vehicle-to-Infrastructure* (V2I) *communication* permite apenas a comunicação entre veículos e infraestrutura. Sua principal desvantagem é o fato de ter que existir uma infraestrutura prévia (RSUs) para permitir

a comunicação entre veículos. A fim de agregar os benefícios das arquiteturas (i) e (ii), a arquitetura (iii) propõe um modo híbrido utilizando comunicação V2V e V2I. A Figura 1 representa as três arquiteturas mencionadas de VANETs.

Apesar do termo VANET estar relacionado à arquitetura (i) é importante salientar que, atualmente, pesquisadores se referem ao termo VANET de forma generalista para ambientes veiculares, mesmo em cenários com a infraestrutura presente.

Figura 1 – Arquiteturas das redes veiculares



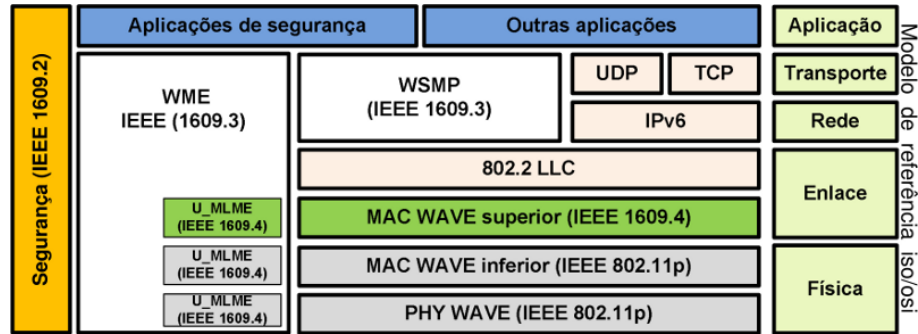
Fonte: Alves et al. (2009).

Com os benefícios e facilidade de comunicação da arquitetura (ii), há a possibilidade de integração desse ambiente veicular com outras redes, como por exemplo as redes de Terceira Geração (3G) (JANG et al., 2009) e Quarta Geração (4G) (HUANG et al., 2012), para assegurar a existência de tal infraestrutura necessária à arquitetura V2I. No entanto, na realidade, não é possível assegurar que sempre existirá tal infraestrutura para gerenciar a conectividade em VANET. Além disso, precisamos considerar o *delay* da comunicação utilizando uma infraestrutura. Algumas aplicações e mensagens críticas na rede veicular podem não suportar tal atraso na comunicação, como por exemplo, mensagens de segurança na via, etc.

2.1.1 WAVE

Wireless Access in Vehicular Environments (WAVE), também conhecido por IEEE 802.11p, é um padrão, baseado no IEEE 802.11, que define as camadas físicas e de controle de acesso ao meio (MAC – *Media Access Control*) para redes veiculares, suportando velocidades de até 200 km/h em aplicações ITS. Além disso, a arquitetura WAVE designa uma família de padrões que não se restringem à camada MAC e física, como apresentado na Figura 2. Os padrões da família IEEE 1609 (KARAGIANNIS et al., 2011) definem outras camadas da pilha de protocolos (incluindo uma camada de rede alternativa à camada IP), características de segurança e operações em múltiplos canais de comunicação (ALVES et al., 2009).

Figura 2 – A pilha de protocolos WAVE

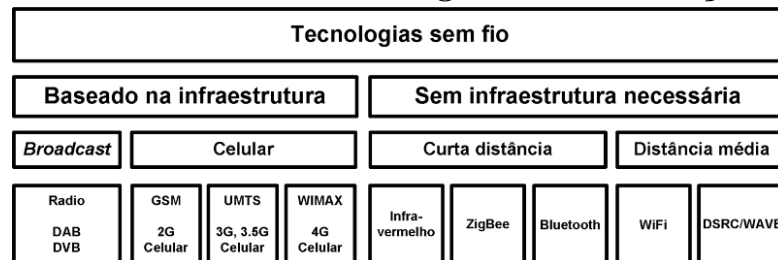


Fonte: Adaptada de Alves et al. (2009).

Os primeiros esforços de padronização das redes veiculares começaram nos Estados Unidos. Em 1999, a FCC (*Federal Communications Commission*) alocou 75 MHz de espectro de frequência, na faixa de 5.9 GHz, para aplicações DSRC (*Dedicated Short Range Communications*). As DSRC são tecnologias de comunicação sem fio, sob regiões ou zonas de tamanho médio (até 1 km²), com taxas de transferência de dados elevadas e baixa latência de comunicação. Espera-se que aplicações ITS DSRC para VANETs operem com taxas de transferência de até 27 Mbps. Em outras partes do mundo, como por exemplo na Europa e no Japão, a banda de frequência reservada deverá ser de 5.8 GHz (GRILLI, 2010). A Figura 3 ilustra uma taxonomia, apresentada por Sommer (2011), de tecnologias de comunicação sem fio.

Nos trabalhos (KARAGIANNIS et al., 2011), (ALVES et al., 2009) e (GRILLI, 2010) mais detalhes da arquitetura e pilha de protocolos WAVE são apresentados e discutidos.

Figura 3 – Taxonomia de tecnologias de comunicação sem fio



Fonte: Adaptada de Sommer (2011).

2.1.2 Simulação em Redes Veiculares

Nas redes veiculares, devido à natureza especial do ambiente, a simulação deve considerar dois aspectos importantes que caracterizam esse tipo de rede: a alta mobilidade dos nós e a comunicação via rádio entre os nós. Como a mobilidade e a comunicação via rádio possuem características bem distintas, tipos diferentes de simuladores para redes veiculares são utilizados. Em (MARTINEZ et al., 2011) os simuladores são divididos em três tipos: (i) simuladores de rede, (ii) geradores de mobilidade, (iii) simuladores de VANET. Contudo, alguns simuladores dos tipos (i) e (ii) têm trabalhado de forma isolada, o que não permite simular cenários mais próximos do ambiente real. Com a necessidade

de comunicação *online* entre esses simuladores, alguns *frameworks* foram implementados. Tal ambiente integrado, conhecido como simulador de VANET, proporciona resultados mais satisfatórios e próximos da realidade de redes veiculares.

Simuladores de Rede

Realizar um conjunto de testes completos, que contenham vários nós na rede, roteadores, links de dados e até uma infraestrutura para validar e verificar o comportamento de um determinado protocolo de rede, ou um algoritmo específico, é muito caro. Os simuladores de rede, nessas circunstâncias, conseguem reduzir o tempo e os gastos financeiros para se realizar essa tarefa. Simulação de rede é comumente utilizada para verificar, investigar e avaliar o comportamento da rede (MALLAPUR; PATIL, 2012).

No trabalho de Martinez et al. (2011) é apresentado um estudo comparativo dos simuladores de rede (usados em VANETs) ns-2, SNS, GloMoSim, JiST/SWANS e GT-NetS. Em (MALLAPUR; PATIL, 2012) uma pesquisa e análise comparativa acerca dos simuladores de rede (para *Mobile Ad-Hoc Networks* - MANETs) ns-2, OMNeT++, NC-TUns e GloMoSim também é realizada. Nesses trabalhos, características dos simuladores são apresentadas, discutidas e comparadas, bem como são avaliadas as vantagens e as desvantagens de cada um.

No trabalho de Font et al. (2010) uma avaliação e comparação sobre as características dos simuladores rede ns-2 e ns-3, do ponto de vista de desenvolvimento, é apresentada. O objetivo dos autores é auxiliar na escolha do simulador. Em (IKEDA et al., 2011) uma análise comparativa de simulações de MANET (usando ns-2 e ns-3) é realizada (com objetivo de atestar a escalabilidade), em que o tempo de execução das simulações em ambos os simuladores de rede são avaliados e comparados. Nesse trabalho específico foi verificado que ns-3 possui escalabilidade maior que o ns-2.

Em (GAMESS; MAHGOUB; RATHOD, 2012), um estudo comparativo de dois simuladores de rede (JiST/SWANS e OMNET++/INETMANET), com ênfase na escalabilidade, é apresentado. Diferentemente de outros trabalhos, que concentram em comparações analíticas de ferramentas de simulação (tais como: curva de aprendizagem, configuração, protocolos suportados, tipo de licença, documentação, ambiente gráfico etc), o trabalho de Gamess, Mahgoub e Rathod (2012) concentra na avaliação do tempo total de simulação e consumo de memória. Mas esse trabalho também tem como objetivo auxiliar e orientar pesquisadores na seleção de uma ferramenta adequada para grandes cenários de simulação. Apesar dos resultados apresentados apontarem um desempenho melhor do simulador JiST/SWANS (em relação ao OMNET++/INETMANET), os próprios autores ressaltam as limitações do simulador JiST/SWANS, como por exemplo falta de protocolos implementados etc.

Estes trabalhos de pesquisa, *surveys*, e estudos comparativos auxiliam na escolha do simulador, para garantir que a ferramenta usada seja apropriada, confiável, eficiente e atenda às necessidades, uma vez que, as funcionalidades dos simuladores de rede variam. Existem simuladores mais complexos e simuladores mais simples, dependendo da aplicação que se deseja simular. No mínimo, um simulador de rede deve permitir que o usuário represente uma topologia de rede, definindo os cenários, especificando os nós da rede, configurando os enlaces e o tráfego entre os nós. Ferramentas mais complexas podem permitir que o usuário especifique todos os parâmetros e protocolos de uma determinada camada. Alguns simuladores de rede também possuem uma interface GUI (*Graphical User Interface*), que melhora a usabilidade e deixa a ferramenta mais intuitiva para implementações futuras do usuário.

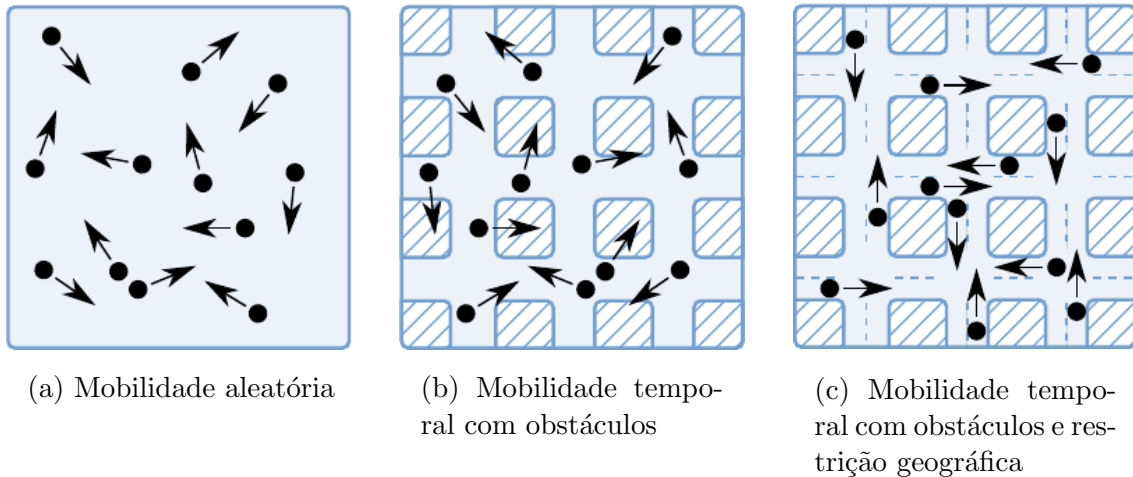
Simuladores de Mobilidade

Muitos erros e resultados tendenciosos surgem quando se utiliza apenas simuladores de rede para simulação de redes veiculares. Isso ocorre porque os simuladores de rede usam padrões de mobilidade determinísticos, quando, na realidade, o comportamento dos veículos possui grande variabilidade aleatória e depende do ambiente operacional. Além disso, os simuladores de rede não representam as características específicas do ambiente veicular, como por exemplo: na rede veicular os veículos se movem sobre uma infraestrutura (ruas, avenidas etc) previamente conhecida e, em um cruzamento, cada veículo pode escolher dentre algumas saídas, mudar de faixa etc. Além disso, em determinados cruzamentos os veículos têm que aguardar a temporização do semáforo. Nesses cenários, os veículos ainda podem apresentar situações imprevisíveis, ocasionando acidentes, congestionamentos etc.

Em (BAI; HELMY, 2006), vários modelos de mobilidade, encontrados na literatura, com características distintas e específicas para diferentes tipos de MANETs, são avaliados. Uma categorização também é proposta pelos autores e se baseia em características específicas como: (i) aleatoriedade, (ii) dependência temporal, (iii) dependência espacial e (iv) restrição geográfica. Modelos aleatórios (i) são modelos em que os nós móveis se movem aleatoriamente e livremente em uma área previamente estabelecida, sem nenhuma restrição. O destino, a velocidade e a direção são escolhidos arbitrariamente e independente dos outros nós na rede. Para modelos com dependência temporal (ii), a movimentação de um nó pode ser limitada pela aceleração, velocidade e mudança de direção. A velocidade atual de cada nó depende de sua velocidade anterior e, por isso, estão correlacionadas, dando, assim, o aspecto temporal. Nos modelos com dependência espacial (iii), o nó não está sozinho no ambiente, outros nós e obstáculos no cenário o influenciam, como por exemplo: em uma estrada, para evitar acidentes, a velocidade de um veículo não deve ultrapassar a velocidade do veículo à sua frente. Para os modelos

com restrição geográfica (iv), a mobilidade dos nós na rede ainda está sujeita a restrições geográficas, isto é, os nós na rede se movimentam apenas sobre segmentos de estrada e faixas previamente conhecidos, como por exemplo vias urbanas.

Figura 4 – Modelos de mobilidade aplicados à movimentação veicular



Fonte: Adaptada de Sommer (2011).

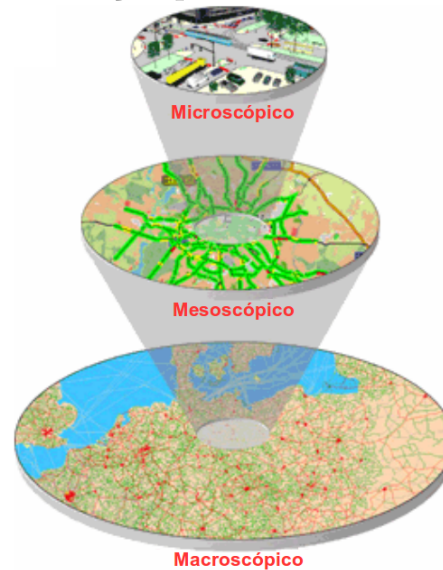
Assim, uma análise e escolha criteriosa acerca de modelos de mobilidade para redes veiculares é necessária, pois existem diversos modelos, com diferentes características, para diversos tipos de MANETs, e alguns não são apropriados ou suficientes para ambientes veiculares, conforme mostram as Figuras 4 (a) e (b).

Os modelos com mobilidade temporal, restrição geográfica e obstáculos (Figura 4 (c)), são os mais adequados para simulações de VANETs, em que a movimentação dos veículos deve ser restrita sobre uma rede rodoviária (mapa) previamente conhecida. Os obstáculos, nesse caso, representam estruturas como prédios etc, que impossibilitam a passagem dos nós e também dificultam a comunicação entre eles.

Outra classificação específica para os simuladores de mobilidade é apresentada em (HARTENSTEIN; LABERTEAUX; EBRARY, 2010), (CRISTEA et al., 2009) e (SOMMER, 2011). Sob âmbito geográfico, esses simuladores são classificados em modelos: (i) microscópico, (ii) mesoscópico e (iii) macroscópico (Figura 5). Os simuladores microscópicos possuem mais detalhes e enfatizam o comportamento local de veículos individuais, representando informações tais como velocidade, aceleração e posicionamento dos nós em cada instante da simulação. Os simuladores macroscópicos capturam as condições do tráfego global, representando, por exemplo, o tráfego através do fluxo de veículos (veículos/hora). Os simuladores mesoscópicos apresentam características de ambos os modelos de simulação microscópico e macroscópico, mas ainda modelam o fluxo de veículos baseado em modelos matemáticos sob uma região do mapa.

Os simuladores de mobilidade macroscópicos e mesoscópicos, por terem uma visão mais generalista, não são suficientes para permitirem um estudo completo de determi-

Figura 5 – Classificação para simuladores de mobilidade



Fonte: Adaptada de Lazaro et al. (2008).

nadas aplicações ITS. Por isso, um simulador microscópico é necessário para simulações mais realistas (simulações de VANET). Entretanto, os simuladores de mobilidade microscópicos possuem baixa escalabilidade quando comparados aos simuladores mesoscópicos e macroscópicos.

Em (MARTINEZ et al., 2011) os simuladores de mobilidade SUMO (*Simulation of Urban MObility*), MOVE, STRAW, FreeSim e CityMob foram apresentados, analisados e comparados quanto a portabilidade, interface, usabilidade, modo de importação de mapas rodoviários reais, modelos de mobilidade e *traces* suportados para integração com outros tipos de simuladores. Em (KRAJZEWICZ et al., 2012) uma descrição e análise detalhada acerca do estado atual do SUMO (simulador de tráfego e mobilidade microscópico de código aberto e multiplataforma) é apresentada, e as principais ferramentas disponibilizadas pelo pacote, bem como a evolução e a função de cada uma, também são abordadas nesse trabalho.

Simuladores de VANET

Na literatura, diversos trabalhos apresentam diferentes simuladores e ferramentas para IVC (*Inter-Vehicle Communication*). Em (DRESSLER et al., 2011), uma análise e uma avaliação dos simuladores mais utilizados para IVC, os modelos, as técnicas utilizadas e as armadilhas para simulação em VANETs são apresentados.

O trabalho de Dressler et al. (2011) concentrou na análise de simuladores para redes veiculares em três aspectos: (i) integração com modelos de mobilidade microscópicos, (ii) métricas de avaliação e (iii) impacto do comportamento humano, que sob o ponto de vista dos autores tem uma forte influência sobre o grau de realismo e portanto na confiabilidade dos resultados. O Quadro 1 mostra um resumo dos principais *frameworks* utilizados para simulação de VANET.

Quadro 1 – Sumário dos *frameworks* para simulações de VANETs

Ferramentas	Simulador de Rede	Simulador de Mobilidade	Simulador de Tráfego	Comportamento Humano
VEINS	OMNET++	SUMO	SUMO	sim
TraNS	ns-2	SUMO	SUMO	sim
iTETRIS	ns-3	SUMO	SUMO	sim
VGSim	JiST/SWANS	Nagel-Schreckenberg	-	-
VSimRTI	JiST/SWANS	VISSIM	VERSIT+	sim
NCTUns	proprietário	proprietário	proprietário	-
SWANS++	JiST/SWANS	STRAW	-	-
GrooveNet	proprietário	Roadnav	-	-
ASH	JiST/SWANS	IDM/MOBIL	-	-
Vanet-Highway	ns-3	IDM/MOBIL	-	-

Fonte: Adaptado de Dressler et al. (2011).

No Quadro 1, três simuladores classificados como simuladores de VANET merecem destaque: **VEINS** – *VEhicles In Network Simulation* (SOMMER; GERMAN; DRESSLER, 2011), (VEINS, 2014); **TraNS** – *Traffic and Network Simulation Environment* (PIÓRKOWSKI et al., 2008), (TraNS, 2014); e **iTETRIS** – *Integrates Wireless Communications and Road Traffic Simulation Platforms* (LAZARO et al., 2008), (iTETRIS, 2014). Apesar da ferramenta VSimRTI (PROTZMANN; SCHUNEMANN; RADUSCH, 2011), (VSimRTI, 2014) também ser classificada como um simulador de VANET completo, o simulador de rede utilizado (JiST/SWANS) é limitado (GAMESS; MAHGOUB; RATHOD, 2012), não possui por exemplo implementações para IEEE 802.11p.

Em (SOMMER; GERMAN; DRESSLER, 2011), detalhes da arquitetura do VEINS, que utiliza comunicação bidirecional entre o simulador de rede e o simulador de tráfego e mobilidade, é apresentado. VEINS, no entanto, é um *framework* de código aberto para IVC (baseado no *framework* MiXiM¹), que já possui implementado o padrão IEEE 802.11p. VEINS, resumidamente, é composto pelo simulador de rede OMNET++² e o simulador de mobilidade e tráfego rodoviário SUMO³.

Para realizar as simulações de VANETs, os simuladores de rede e mobilidade são executados em paralelo e interligados através de um *socket* TCP. O protocolo TraCI (*Traffic Control Interface*) (WEGENER et al., 2008) é utilizado para comunicação bidirecional de tráfego veicular e tráfego de rede.

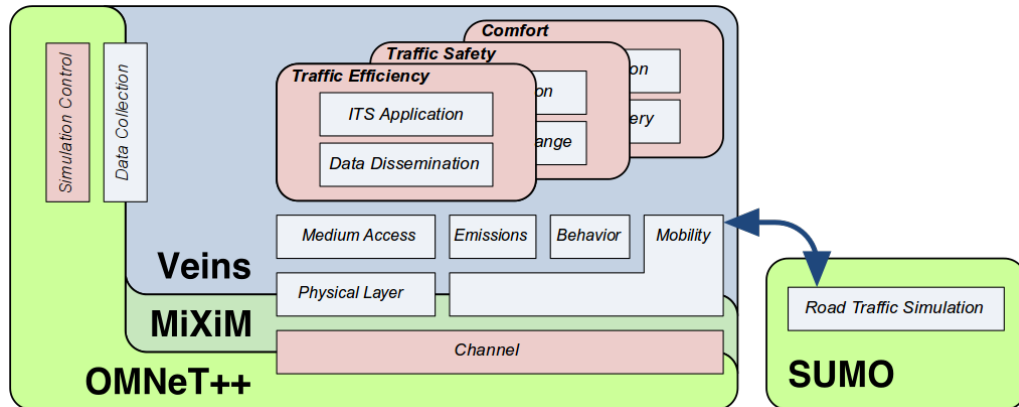
No VEINS, o movimento dos veículos no tráfego rodoviário é exibido no simulador de rede OMNET++, que reflete o movimento dos nós (veículos) no simulador de mobilidade SUMO. Outra característica importante do *framework* é a possibilidade de interação da simulação de tráfego veicular com a simulação de rede em tempo real. Com o VEINS, um determinado evento pode acarretar a alteração da rota de um determinado nó, ou

¹Disponível em <http://mixim.sourceforge.net/>

²Disponível em <http://www.omnetpp.org/>

³Disponível em <http://sumo-sim.org/>

Figura 6 – Arquitetura do VEINS

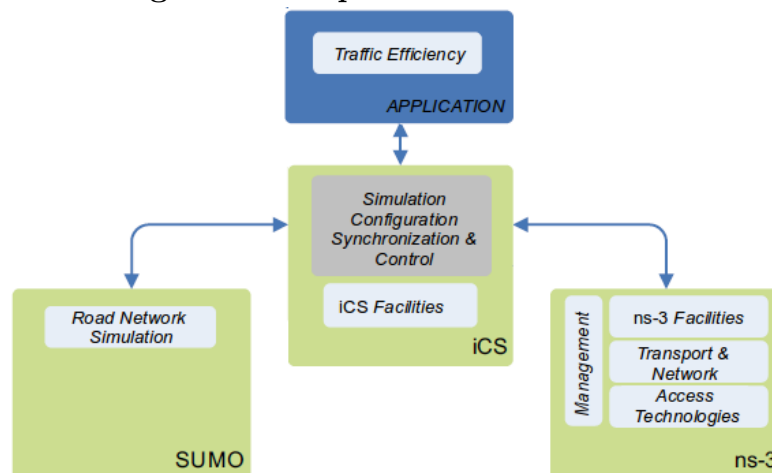


Fonte: Adaptada de VEINS (2014).

conjunto de nós na rede veicular. Podem ser citados como eventos: acidentes, congestionamentos de tráfego veicular etc. A Figura 6 ilustra a arquitetura do VEINS. Nela é possível verificar toda estrutura e as camadas do *framework*.

A outra ferramenta avaliada por este trabalho para simulação de redes veiculares, que possui um ambiente integrado com um simulador de mobilidade (SUMO) e de rede (ns-3) é o iTETRIS. Em (RONDINONE, 2010) e (LAZARO et al., 2008) detalhes da arquitetura do iTETRIS (Figura 7) são analisados e discutidos. O iCS (iTETRIS *Control System*) integra o simulador de mobilidade com um simulador de rede, através do protocolo TraCI. Em (KUMAR et al., 2010), problemas de escalabilidade da plataforma do iTETRIS são abordados, e algumas estratégias são apresentadas e discutidas.

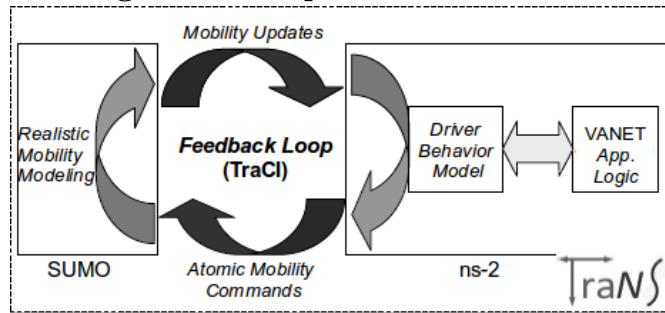
Figura 7 – Arquitetura do iTETRIS



Fonte: Adaptada de Rondinone (2010).

O outro simulador de VANET, muito citado na literatura, é o TraNS. Essa ferramenta possui um ambiente que integra o simulador de mobilidade (SUMO) e o simulador de rede (ns-2). O TraNS, por ter sido pioneiro na integração de um simulador de mobilidade e um simulador de rede para VANETs, foi utilizado inclusive como base para o desenvolvimento do iTETRIS. No entanto, o aspecto negativo do TraNS é o fato do projeto ter sido descontinuado. A última versão de 2008 não permite integração com as

Figura 8 – Arquitetura do TraNS



Fonte: Adaptada de Piórkowski et al. (2008).

versões mais recentes do SUMO. Em (PIÓRKOWSKI et al., 2008) a arquitetura do TraNS (Figura 8) para simulações de VANETs é apresentada e discutida. Na Figura 8 é possível observar três módulos distintos, que correspondem aos respectivos simuladores de mobilidade, rede e VANET. Essa figura também mostra a comunicação bidirecional (realizada pelo protocolo TraCI) entre o simulador de mobilidade e o simulador de rede.

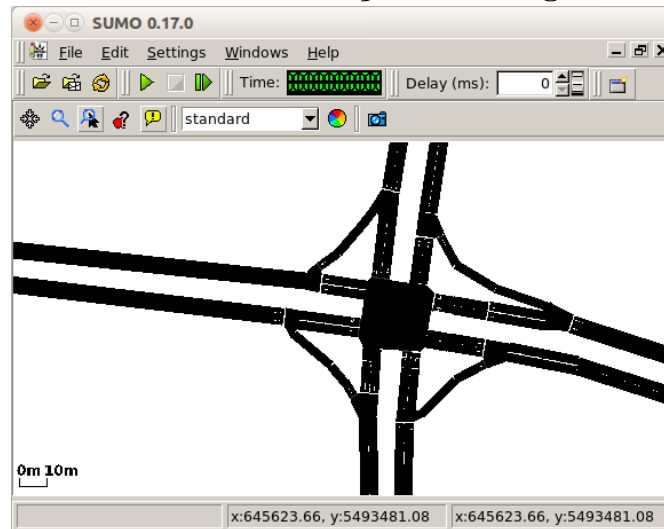
2.1.3 Rede de Tráfego Rodoviário

A rede rodoviária é discriminada basicamente em dois tipos distintos: **autoestrada** e **vias urbanas** (MENEZES, 2013), (MACHADO, 2009). Essa rede rodoviária, influencia na mobilidade e no tráfego dos nós na rede veicular. Nesse contexto, existem várias ferramentas na literatura com o objetivo de auxiliar na configuração dos cenários, na geração de mobilidade e tráfego rodoviário. Essas ferramentas são necessárias quando se trabalha com redes complexas, facilitando por exemplo: a geração da rede rodoviária, importação de mapas reais de cidades, edição de mapas rodoviários, geração de tráfego rodoviário, geração de mobilidade (baseado em: uma aleatoriedade, uma demanda ou rotas específicas) etc. Abaixo são listadas algumas dessas ferramentas e suas funcionalidades:

SUMO – Ambiente Rodoviário: o SUMO (apresentado na Seção 2.1.2) também permite simulação de tráfego rodoviário, sendo considerado um simulador de tráfego e mobilidade veicular. A ferramenta, desenvolvida e mantida pelo Instituto de Sistemas e Transporte do Centro Aeroespacial Alemão (DLR, 2014), basicamente permite que cada veículo seja modelado de forma explícita, com um percurso próprio, se movendo individualmente através de uma rede de estradas. Apesar do ambiente gráfico do SUMO ser opcional, a Figura 9 mostra uma tela desse ambiente para um cenário de simulação de tráfego rodoviário.

Nessa ferramenta, os fluxos de tráfego veicular podem ser atribuídos manualmente, calculados com base em dados de demanda (matriz OD – **O**rigem **D**estino), ou gerados aleatoriamente, de acordo com a rede rodoviária definida. Cada segmento de estrada no SUMO pode ter várias faixas, velocidade máxima permitida e ser restrito a tráfego de certas classes de veículos. Faixas individuais podem ter qualquer configuração e podem

Figura 9 – SUMO - Simulação de tráfego rodoviário



Fonte: Criada pelo autor através do SUMO.

ser interligadas umas às outras por cruzamentos, através da aplicação de regras simples de trânsito. Tais cruzamentos ainda podem regular a passagem do tráfego de veículos usando semáforos (fixos ou por demanda).

Entretanto, em cenários mais realistas que utilizam redes rodoviárias reais, somente o SUMO para geração de tráfego rodoviário não é suficiente, uma vez que existem várias informações a ser inseridas no mapa. Nesse caso, para geração de redes complexas, o SUMO utiliza a importação de outros tipos de dados, como por exemplo arquivos *osm* e *shape*, que possuem um formato mais adequado para armazenamento de informações e atributos associados a sistemas de localização geográfica (GIS - *Geographic Information System*).

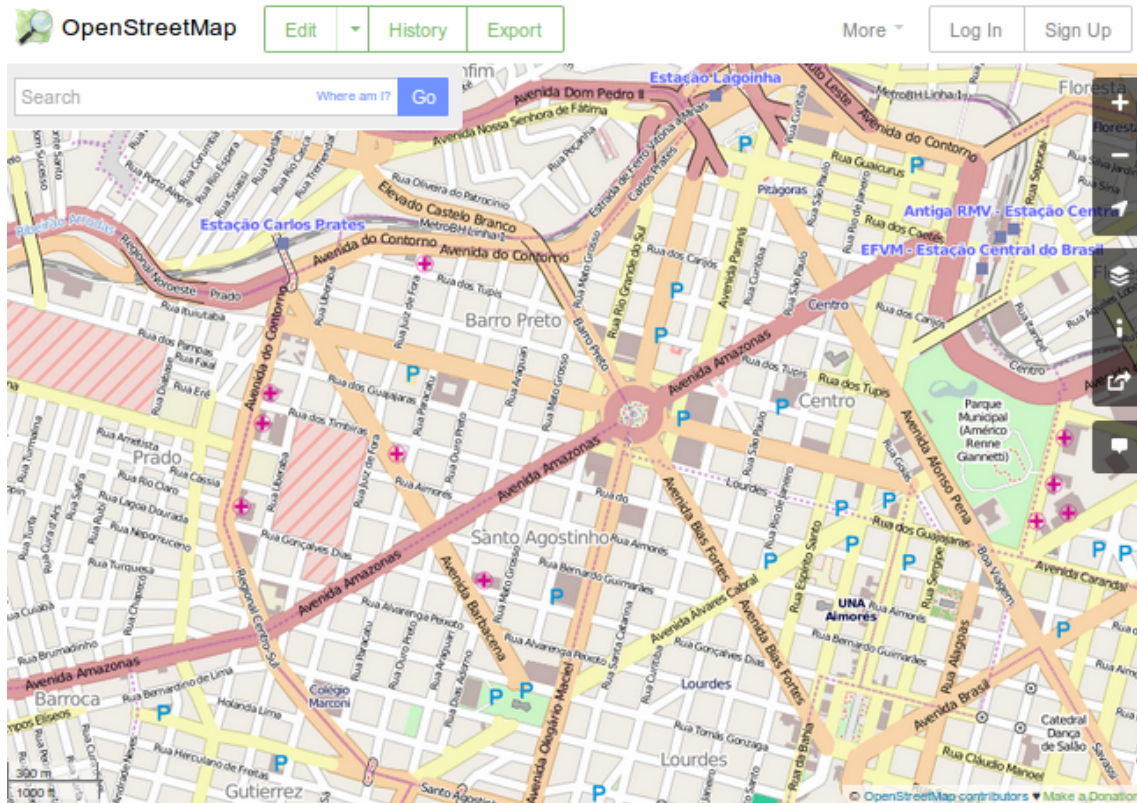
OpenStreetMap⁴ (OSM): é o nome do projeto e fundação de apoio ao projeto, que tem como objetivo coletar e fornecer dados geográficos gratuitamente, permitindo acesso livre às imagens e à base de dados dos mapas (HAKLAY; WEBER, 2008). Existem outras alternativas em relação ao OpenStreetMap, mas a maioria delas são pagas. Assim, o OpenStreetMap é usado para aumentar o realismo de simulações de VANETs, exportando mapas e dados rodoviários reais para importação no SUMO (MACHADO, 2009). A Figura 10 ilustra a interface web do OpenStreetMap, que mostra o mapa geográfico da região central da cidade de Belo Horizonte (BH), MG – Brasil.

O OpenStreetMap também disponibiliza uma ferramenta (JOSM⁵ – *Java Open Street Map editor*) que auxilia na edição de mapas (exportados da plataforma WEB) e informações do tráfego rodoviário. O JOSM é escrito em Java, possui código aberto e licença livre. Em (JOSM, 2014), um guia detalhado para utilização da ferramenta é apresentado.

⁴Disponível em <http://www.openstreetmap.org/>

⁵Disponível em <http://josm.openstreetmap.de/>

Figura 10 – Interface WEB do OpenStreetMap sobre uma região central em Belo Horizonte, MG



Fonte: Criada pelo autor através do OpenStreetMap.

TrafficModeler: avaliado em (PAPALEONDIU; DIKAIKOS, 2009), essa ferramenta de código aberto, com interface gráfica e modelagem rápida (em alto nível), é usada para geração de tráfego veicular, sobre um determinado mapa rodoviário (previamente definido). O TrafficModeler possui um conjunto de algoritmos de geração de mobilidade direcionado para o SUMO. O objetivo é reduzir consideravelmente o tempo e esforço necessário para geração de mobilidade em simulações de VANETs.

2.2 Lógica Fuzzy

Esta seção explica conceitos de lógica *fuzzy* (LF, difusa, ou nebulosa). A teoria dos conjuntos *fuzzy* definida por Zadeh (1965) procura dar um tratamento matemático a certos termos linguísticos subjetivos como: *aproximadamente*, *em torno de*, *baixo*, *médio*, *alto* etc. Pode-se dizer que tal teoria tem como objetivo criar uma lógica de raciocínio aproximado, que busca representar um primeiro passo no sentido de se programar e armazenar conceitos vagos computacionalmente, tornando possível a produção de cálculos sobre informações imprecisas (ENGELBRECHT, 2007).

Como a finalidade desse sistema é reproduzir características do raciocínio humano, o conhecimento é representado através das regras ou proposições e as respostas são obtidas através de inferências, que buscam tirar conclusões (deduções) a partir de fatos conhecidos.

A palavra *fuzzy* (do inglês) significa incerto, vago, impreciso, subjetivo, nebuloso. Embora a teoria dos conjuntos *fuzzy* estude casos de incertezas, é importante frisar que tal teoria é muito bem definida em (ZADEH, 1965). O que é incerto é a propriedade que define o conjunto em questão. Engelbrecht (2007) e Tanscheit (2004) explicam detalhadamente lógica *fuzzy*. A Figura 11 ilustra um mecanismo de lógica *fuzzy*, e os conceitos apresentados, resumidamente, a seguir são necessários para o entendimento do funcionamento desse mecanismo:

Conjuntos Fuzzy: na teoria clássica dos conjuntos, um conjunto A do universo X pode ser representado pela função característica f_A :

$$f_A(x) = \begin{cases} 1 \Leftrightarrow x \in A \\ 0 \Leftrightarrow x \notin A \end{cases}$$

A teoria dos conjuntos *fuzzy* propõe uma caracterização mais ampla, generalizando a função característica de modo que ela possa assumir um número infinito de valores no intervalo $[0,1]$, e permitir que um determinado elemento pertença a mais de um conjunto *fuzzy* ao mesmo tempo. Assim, um conjunto *fuzzy* A em X é definido por uma função de pertinência $\mu_A(x): X \rightarrow [0,1]$, e representado por um conjunto de pares ordenados $A = \{ (\mu_A(x_i)/x_i) \mid x_i \in X, i = 1, \dots, n \}$, em que μ_A indica o quanto x é compatível com conjunto A .

Variáveis Linguísticas: são variáveis cujos valores são rótulos (*labels*) de conjuntos *fuzzy* e tem como finalidade fornecer uma maneira sistemática para caracterização aproximada de fenômenos complexos ou mal definidos. Assim, a utilização de descrições linguísticas, como as empregadas por seres humanos (em vez de variáveis quantitativas), permite a análise de sistemas que são muito complexos para serem avaliados através de termos matemáticos convencionais. Por exemplo, a *velocidade* de um veículo pode ser uma variável linguística, assumindo rótulos de *lenta*, *moderada* e *rápida*. Esses rótulos são descritos por intermédio de conjuntos *fuzzy*, e sobre um universo de discurso (X), que nesse exemplo específico da velocidade poderia ser de: 0 a 200 km/h.

Fuzzyficação: nessa fase basicamente ocorre o mapeamento das variáveis de entrada determinísticas (numéricas) para conjuntos de termos ou adjetivos linguísticos. O intuito é determinar os correspondentes graus de pertinência das variáveis de entrada em relação aos conjuntos *fuzzy*.

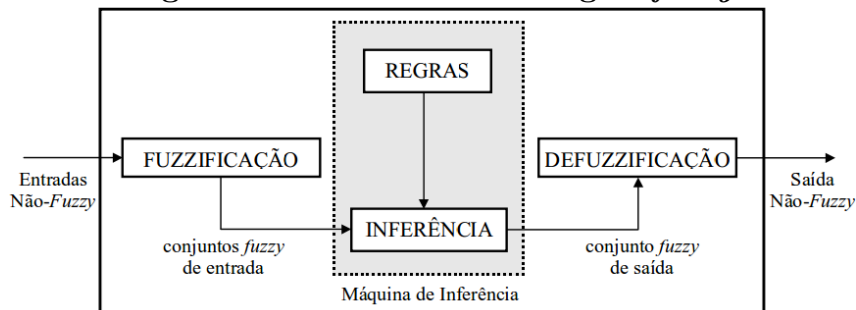
Máquina de Inferências: é considerada o núcleo do mecanismo *fuzzy*. Uma declaração condicional (regra) *fuzzy* da forma *SE ... ENTÃO ...* definida matematicamente por $R: SE\ A\ ENTÃO\ B = A \rightarrow B = A \times B$, sendo A e B conjuntos *fuzzy*, é geralmente fornecida por especialistas do domínio ou extraída de uma base de dados numéricos. Em geral essas regras relacionam *antecedentes* (variáveis de entrada) como *consequentes* (variáveis de saída), podendo ser combinados por meio de diferentes operadores, como verificado abaixo, em que A_i , B_i e C_i são conjuntos *fuzzy*.

$$\begin{aligned} R_1: & \underline{SE}\ x\ \text{é}\ A_1\ \underline{E}\ y\ \text{é}\ B_1\ \underline{ENTÃO}\ z\ \text{é}\ C_1 \\ R_2: & \underline{SE}\ x\ \text{é}\ A_2\ \underline{E}\ y\ \text{é}\ B_2\ \underline{ENTÃO}\ z\ \text{é}\ C_2 \\ & \vdots \\ R_n: & \underline{SE}\ x\ \text{é}\ A_n\ \underline{E}\ y\ \text{é}\ B_n\ \underline{ENTÃO}\ z\ \text{é}\ C_n \end{aligned}$$

Essas regras, juntamente com os conjuntos *fuzzy*, formam a base de conhecimento do sistema, que será consultada pelo mecanismo de inferência. A finalidade dessa base é calcular a saída *fuzzy* e, para isso, ocorrem operações com conjuntos *fuzzy*, através de um procedimento de avaliação das regras *fuzzy* e relacionamento das variáveis linguísticas.

Defuzzificação: é a fase de converter a saída *fuzzy* do sistema para um valor escalar (não-*fuzzy*) que seja preciso. Para isso, o cálculo do centroide (TANSCHKEIT, 2004) é o método mais usado. Tal método calcula o centro de gravidade do conjunto *fuzzy* de saída, produzido pela Máquina de Inferência e, dessa forma, estima o resultado para o sistema.

Figura 11 – Mecanismo de lógica *fuzzy*



Resumidamente: o processo de fuzzificação é responsável por converter os valores numéricos (não-*fuzzy*) de entrada para valores *fuzzy*. Após essa etapa a máquina de inferência, a partir das regras definidas, calcula a saída *fuzzy*, para em seguida, na defuzzificação, ocorrer a redução da saída *fuzzy* do sistema para um valor escalar (não-*fuzzy*).

Fonte: Adaptada de Tanscheit (2004).

Na literatura existem várias implementações para mecanismos *fuzzy*, conforme mostra a Figura 11, como por exemplo: FuzzyLite⁶ (RADA-VILELA, 2013) e jFuzzyLogic⁷ (CINGOLANI; ALCALA-FDEZ, 2012). A vantagem do FuzzyLite é ser desenvolvido em C++, multiplataforma, possuir licença livre e código aberto. O FuzzyLite permite a criação rápida de um sistema *fuzzy*. Para isso, o projeto disponibiliza uma ferramenta gráfica, o *qt fuzzylite*, que auxilia na configuração (através de uma GUI) de todos os componentes de um mecanismo de lógica *fuzzy*. Essa ferramenta, que também é multiplataforma, exporta todos os dados gerados graficamente para a *Fuzzy Control Language* (FCL). Tal linguagem é padronizada pela *International Electrotechnical Commission* (International Electrotechnical Commission (IEC) 61131-7 standard, 2000) e específica para esse domínio, em que só características relacionadas à lógica *fuzzy* são tratadas. O benefício em se usar essa linguagem é separar a parte do código de implementação do mecanismo de lógica *fuzzy*, geralmente em C++ ou Java, da parte de modelagem e configuração do sistema propriamente. Assim, a FCL, além de facilitar a configuração do sistema, agiliza manutenções futuras, em que basta alterar o arquivo FCL para que o sistema *fuzzy* seja automaticamente ajustado.

No contexto de redes veiculares, um sistema *fuzzy* pode ser usado para melhorar a eficiência da detecção de congestionamento veicular em aplicações ITS. Pode-se utilizar por exemplo: (i) a velocidade do veículo e (ii) densidade veicular na via como conjuntos *fuzzy* de entrada do sistema. O (iii) nível de congestionamento seria o conjunto *fuzzy* de saída do sistema. Assim, os respectivos conjuntos fuzzy (i), (ii) e (iii) poderiam ter as respectivas variáveis linguísticas: *lenta*, *moderada*, *rápida*; *baixa*, *média*, *alta*; *livre*, *fraco*, *moderado*, *severo*. Também seria possível supor as seguintes regras *fuzzy*: velocidades *lentas* e *altas* densidades de veículos vizinhos identificam tráfego veicular *severo*; velocidades *rápidas* e densidades *baixas* indicam tráfego *livre* etc. Contudo, somente um sistema *fuzzy* não irá resolver todos os problemas de controle de congestionamento em redes veiculares. Outras abordagens precisam ser avaliadas.

2.3 Trabalhos Relacionados

Mecanismos adequados para controle de congestionamento veicular são essenciais para otimizar o fluxo de veículos em VANETs. As aplicações de controle de congestionamento em redes veiculares podem ser discriminadas em identificação, minimização e prevenção de congestionamento. A identificação de congestionamento veicular representa a eficácia em se caracterizar um congestionamento no trânsito. A minimização corresponde à tentativa de reduzir os congestionamentos detectados. A prevenção de congesti-

⁶Disponível em <http://www.fuzzylite.com/>

⁷Disponível em <http://jfuzzylogic.sourceforge.net/>

onamento é uma abordagem complexa, uma vez que precisa gerenciar fluxos de veículos nas vias, para prevenir que congestionamentos ocorram. Esse problema de prevenção de congestionamento pode ser reduzido em um problema de minimização de fluxos em grafos (DEO, 1974), o que o remete à classe de problemas NP-Completo (RUHE, 1991).

Vários trabalhos, como (FUKUMOTO et al., 2007) e (FAHMY; RANASINGHE, 2008), apresentam propostas de monitoramento e detecção de condições do tráfego utilizando apenas comunicação V2V. Para isso, são transmitidas mensagens do tipo *beacon*. Tais mensagens são bem simples, possuem um baixo *overhead* e são usadas basicamente para anunciar a presença de um veículo aos vizinhos. Outras abordagens, como o ConProVa (SILVA et al., 2013) utilizam apenas a comunicação V2I, o que limita a proposta. No ConProVa, estruturas para lidar com questões relativas ao provimento de contextos em redes veiculares são criadas na infraestrutura. Para isso, o ConProVa implementa um *middleware*, a fim de ajudar na tomada de decisão e resolução de conflitos de interesse sobre a situação real do tráfego relatada pelos veículos, uma vez que os veículos não conseguem validar colaborativamente (comunicação V2V) as condições detectadas na via.

Existem também abordagens como o ECODE (YOUNES; BOUKERCHE, 2013), que utilizam ambas as comunicações V2V e V2I. O ECODE propõe um mecanismo de detecção de congestionamento, disseminação de dados do tráfego e recomendação de um caminho para os veículos, sem sobrecarregar o canal de comunicação. No entanto, o ECODE apesar de utilizar a comunicação V2V depende da infraestrutura (RSU), nas áreas de monitoramento, para identificar congestionamentos. Essa característica limita o protocolo a cenários específicos, e como citado pelos autores, a proposta é direcionada exclusivamente para detecção de congestionamento em centros urbanos. Outra deficiência não tratada no ECODE ocorre ao disseminar informações na rede de um caminho ótimo. Essa abordagem do protocolo tende a criar novos pontos de congestionamento, devido a não utilização de políticas adequadas para auto regular o tráfego nas vias.

Como as abordagens na literatura possuem características distintas, uma classificação para propostas de monitoramento de tráfego veicular foi adaptada para este trabalho, a fim de permitir a identificação e comparação das técnicas analisadas. O Quadro 2 mostra informações sumarizadas e comparativas em relação às referências deste trabalho.

A classificação adotada distingue os trabalhos analisados de acordo com os seguintes aspectos: (i) A identificação de congestionamento retrata a eficácia em mensurar as condições do tráfego na via. (ii) validação cooperativa de tráfego veicular se refere à capacidade da proposta em utilizar a comunicação cooperativa entre veículos para chegar a uma decisão (consensual) sobre a situação do tráfego em uma região. (iii) O tamanho (extensão em metros ou quilômetros) do engarrafamento indica a eficácia em estimar a extensão de um congestionamento. (iv) O *overhead* é a capacidade de se evitar a sobre-

carga do canal de comunicação. (v) A disseminação de informação retrata a característica da proposta em propagar dados da situação do tráfego na rede. (vi) A arquitetura mostra como a rede veicular se organiza. (vii) A mudança de rota indica se existe uma política para os veículos alterarem suas rotas iniciais.

Quadro 2 – Propostas de monitoramento de tráfego veicular avaliadas

Proposta	(i) Identificação de Congestionamento	(ii) Validação Coope- rativa de Tráfego	(iii) Tamanho do En- garrafamento	(iv) Baixo Overhead	(v) Disseminação	(vi) Arquitetura(s)	(vii) Mudança de Rota
COC (FUKUMOTO et al., 2007)	sim ⁽¹⁾	sim	não	não	não	v2v	não
(FAHMY; RANASINGHE, 2008)	sim ⁽¹⁾	não	não	sim	não	v2v	não
ConProVa (SILVA et al., 2013)	sim ⁽¹⁾	não	não	sim	sim	v2i	sim
ECODE (YOUNES; BOUKERCHE, 2013)	sim ⁽¹⁾	não	não	sim	sim	v2x ⁽²⁾	sim
TrafficView (NADEEM et al., 2004)	sim ⁽¹⁾	não	não	não	não	v2v	não
VOTING (PADRON, 2009)	sim ⁽¹⁾	sim	não	sim	sim	v2v	não
(MILLER, 2008)	sim ⁽¹⁾	não	sim	não	não	v2x ⁽²⁾	não
StreetSmart (DORNBUSH; JOSHI, 2007)	sim ⁽¹⁾	sim	sim	sim	não	v2v	não
(LAKAS; CHEQFAH, 2009)	sim ⁽¹⁾	sim	sim	não	não	v2v	sim
(KNORR; SCHRECKENBERG, 2012)	sim ⁽¹⁾	não	não	sim	não	v2v	sim
DTMon (ARBABI; WEIGLE, 2011)	sim ⁽¹⁾	não	não	sim	não	v2x ⁽²⁾	não
(LÜ; CAO, 2003)	sim	não	não	sim	não	v2i	não
(BINGLEI; ZHENG; HONGWEI, 2008)	sim	não	não	sim	não	v2i	não
(SHAN; JIANSHEG; LING, 2012)	sim	não	não	sim	não	v2i	não
COTEC (BAUZA; GOZALVEZ; SANCHEZ-SORIANO, 2010)	sim	sim	sim	sim	sim	v2v	não
CARTIM	sim	sim	sim	sim	sim	v2x ⁽²⁾	sim

1 - Apenas detecta um congestionamento, não classifica e nem estima níveis de congestionamento.

2 - Arquiteturas V2V e V2I.

Fonte: Dados da pesquisa.

O COC (*Content Oriented Communications*), proposta apresentada em (FUKUMOTO et al., 2007), tem como objetivo fornecer informações oportunas de congestionamentos e acidentes veiculares para motoristas. Para isso, os veículos trocam três tipos de informações entre si (comunicação V2V) chamadas de: *Level one* (Lv1), *Level two* (Lv2) e *Level three* (Lv3). Cada tipo de informação possui seu próprio *buffer* para armazenamentos dos dados. Informações Lv1 são utilizadas para enviar a posição (baseado no GPS - *Global Position System*) atual do veículo, velocidade momentânea etc. Informações Lv2 mantêm dados tais como: distância percorrida por segundo e densidade do tráfego (estimada a partir de *broadcast beacons* recebidos de outros veículos). As informações Lv3 são dados agrupados e gerados a partir de informações Lv2 recebidas (de vizinhos), como por exemplo: distância média percorrida do grupo, velocidade média do grupo etc. No COC, cada veículo avalia a situação do tráfego com base nessas informações trocadas. No entanto, a capacidade da proposta em detectar congestionamento de tráfego é obtida com um custo adicional, devido à sobrecarga do canal de comunicação. Para evitar esse *overhead* de comunicação, políticas para disseminação de mensagens na rede devem ser adotadas pelos veículos.

Em (NADEEM et al., 2004), um *framework* (TrafficView) para disseminar e agrupar informações sobre veículos na estrada é apresentado. O TrafficView tem como objetivo propor um sistema que alerte e informe motoristas sobre situações críticas no trânsito e ajude a encontrar uma rota ideal. Para isso cada veículos armazena dados sobre si e sobre veículos vizinhos à frente, como por exemplo: ID (*identification*) do veículo, posição (GPS), velocidade, *broadcast time* etc. Tais informações são trocadas através de *broadcast messages*. No entanto, como mencionado pelos autores, a camada MAC do protocolo IEEE 802.11b (BILGIN; GUNGOR, 2013), utilizado pelo TrafficView, limita a carga de comunicação em 2312 bytes. Dessa forma, abordagens para minimizar a sobrecarga de comunicação devem ser adotadas, uma vez que a quantidade de dados trocados pelos veículos é alta. Para tratar esse problema, TrafficView utiliza uma proposta baseada no agrupamento de dados de veículos vizinhos, pois a compressão de dados exigiria um processamento extra do *hardware* proposto para o sistema.

Outras técnicas utilizam estratégias diferentes para diminuir a carga de comunicação entre veículos. Como por exemplo o VOTING (*Vehicular Over-The-air INformation Gathering*) (PADRON, 2009), que utiliza um algoritmo para veículos enviarem mensagens *beacons* por *broadcast* apenas em intervalos fixos de tempo. Essas mensagens transmitem informações sobre a posição atual do veículo, velocidade, direção do deslocamento etc.

O objetivo do VOTING concentra em detectar congestionamento veicular (em tempo real) utilizando apenas comunicação V2V, sem nenhum elemento centralizador (infraestrutura) presente. Para isso, a proposta segue uma ideia simples, baseada na decisão por maioria, ou seja, a partir da detecção de velocidade abaixo do normal de qualquer veículo no segmento de estrada, antes de disseminar informações na rede sobre o tráfego, os demais veículos no segmento precisam validar a área congestionada. Isto é, começa um processo colaborativo de votação, em que os veículos sinalizam se estão de acordo com as condições anormais do tráfego. Quando o número de veículos no segmento de estrada de acordo com a situação supera o número de veículos em desacordo, o congestionamento veicular na região é validado e transmitido a veículos vizinhos em outros segmentos de estrada. A principal contribuição do VOTING está relacionada justamente a essa abordagem de validação cooperativa, que tende a conduzir a detecções de congestionamentos de tráfego mais confiáveis.

Diferentemente do TrafficView e VOTING, o trabalho de Miller (2008) tenta reduzir a sobrecarga do canal de comunicação apresentando uma técnica individualista, que determina que apenas um único veículo, em cada segmento de estrada, seja responsável por coletar e adicionar dados do tráfego. Essa abordagem, no entanto, apresenta problemas de sobrecarga sobre veículo responsável pelo segmento. Outro problema é a falta de acurácia da situação real de todo o segmento. Daí, surge a necessidade de criar um algoritmo eficiente para a escolha do veículo responsável pelo segmento.

Em (DORNBUSH; JOSHI, 2007), é apresentado o StreetSmart. A abordagem não necessita que todos os veículos adotem o sistema, o mesmo é projetado para executar sob baixa densidade de veículos, sendo inclusive tolerante a desconexões. Sua principal contribuição, em relação às propostas descritas, está na troca de informações (resumidas) limitadas apenas a situações de tráfego inesperado ou anormal, o que possivelmente caracteriza um congestionamento. Tais informações resumidas ainda são combinadas entre pequenos grupos de veículos. Condições anormais no tráfego são inicialmente identificadas através da velocidade do veículos e velocidade permitida na via. O sistema usa o GPS para determinar a localização atual e velocidade dos veículos, permitindo assim que cada veículo monte seu mapa de tráfego local.

Como nos trabalhos de Dornbush e Joshi (2007), Fukumoto et al. (2007) e Nadeem et al. (2004) outras aplicações de controle de congestionamento veicular para VANETs também se baseiam no sistemas de posicionamento global (GPS). No entanto, esse sistema apresenta limitações em cenários específicos, como por exemplo dentro de túneis, dias muito nublados etc. Uma alternativa é discutida em (FAHMY; RANASINGHE, 2008), e consiste em descobrir automaticamente congestionamentos de tráfego com base em informações recebidas apenas de mensagens *beacons*, trocadas entre veículos. A abordagem utiliza para isso um algoritmo baseado em estrutura de dados árvore (CORMEN et al., 2009) e um modelo de filas. O objetivo é detectar um congestionamento e contabilizar o total de nós (veículos) no engarrafamento. Tal trabalho baseou-se nas redes de sensores sem fio para propor a técnica, que, segundo justificativa dos autores, apesar das redes veiculares possuírem alta mobilidade dos nós, em condições de congestionamento, a mesma se comporta de maneira similar a redes de sensores sem fio, com baixas alterações na topologia da rede. Dessa forma, alguns algoritmos consolidados para redes de sensores sem fio poderiam ser customizados e ajustados para resolver o problema. No entanto, o mecanismo de detecção de congestionamento proposto em (FAHMY; RANASINGHE, 2008) é simples, possui deficiências e sob baixa densidade de veículos pode ocasionar falhas na detecção de congestionamento, pois a proposta leva em consideração apenas a quantidade de veículos para caracterizar um congestionamento.

Em (LAKAS; CHEQFAH, 2009), cada veículo continuamente estima um índice de congestionamento, para todo segmento de estrada que deverá passar, e compartilha com outros veículos na mesma ZOR (*Zone Of Relevance*). Esse índice é estimado considerando o tempo de viagem atual e o tempo de viagem esperado (sem congestionamento de tráfego). A proposta também define um procedimento de pergunta e resposta entre os veículos para obter informações sobre o tráfego veicular em uma determinada área. A abordagem utiliza uma adaptação do Algoritmo de Dijkstra (CORMEN et al., 2009), (DEO, 1974) para obter a rota que os veículos devem adotar. O problema dessa abordagem é que ela provoca alta sobrecarga de comunicação e saturação do canal de comunicação, caso vários veículos

respondam à mesma requisição. A grande contribuição desse trabalho é que ele não só detecta o tráfego veicular, mas também tenta evitar que veículos acessem uma região congestionada. No entanto, tende a criar novos pontos de congestionamento no tráfego, uma vez que nenhuma política para mudança das rotas é adotada.

A abordagem apresentada em (KNORR; SCHRECKENBERG, 2012) também se assemelha à ideia proposta por Lakas e Cheqfah (2009), pois além de detectar congestionamentos veiculares, tenta evitar que veículos acessem a região. A proposta utiliza a troca periódica de *beacons* para mensurar as condições do tráfego, através de informações como velocidade, aceleração etc. O protocolo estabelece e utiliza um limiar para velocidade, denotado por V_t (*velocity threshold*), para criar um critério (política) simples de mudança de rota. A principal contribuição do trabalho de Knorr e Schreckenberg (2012), que difere do trabalho de Lakas e Cheqfah (2009), está relacionada a essa política mais elaborada para concessão de alteração de rota dos veículos. Além disso, a técnica empregada evita sobrecarga do canal colaborativo de comunicação por utilizar apenas *broadcast beacons*. No entanto, a proposta possui deficiências. A inexistência de um processo colaborativo de validação das condições detectadas pelos veículos é uma delas. Os resultados obtidos em (KNORR; SCHRECKENBERG, 2012), contudo, foram muito satisfatórios. Verificou-se que, para uma taxa de até 30% dos veículos utilizando comunicação V2V, o congestionamento praticamente desaparece. Mas, análises criteriosas ainda precisam ser feitas, pois em situações de 100% dos veículos utilizando comunicação V2V, o protocolo pode não ser eficiente, e permitir que todos os veículos na ZOR alterem suas rotas. Isso causaria o surgimento de novos pontos de congestionamento.

O DTMon (*Dynamic Traffic Monitoring system*), uma outra abordagem que utiliza as arquiteturas V2V e V2I, é retratada em (ARBABI; WEIGLE, 2011). A técnica proposta consegue monitorar as condições do tráfego apenas em situações sob tráfego livre, ou seja, sem obstrução total do tráfego veicular, o que limita a proposta. A técnica insere novos conceitos e apresenta deficiências, como a necessidade de existir uma infraestrutura prévia para detecção das condições do tráfego. No DTMon, as RSUs são utilizadas para enviar periodicamente tarefas e responsabilidades para veículos específicos, como por exemplo: cálculo de velocidade média, localização, tempo do trajeto em um segmento de estrada etc. Cabe, então, ao veículo selecionado realizar a tarefa requisitada e reportar as informações à respectiva RSU, que solicitou a tarefa. Para isso, o veículo tenta primeiro utilizar as comunicações V2I e V2V para entrega das informações. Caso não consiga, o protocolo propõe um procedimento para contornar momentaneamente a falta de uma infraestrutura, em que o veículo com informações a serem retransmitidas, ao entrar na região de outra RSU adjacente, transmite a respectiva mensagem. O TMC (*Traffic Management Center*) da arquitetura do DTMon é um módulo responsável por interligar todas as RSUs, assegurar a entrega de mensagens entre RSUs adjacentes e gerenciar informações sobre

as condições do tráfego. Com isso, a grande contribuição do DTMon é tentar utilizar os benefícios da comunicação V2I quando a infraestrutura estiver presente. A deficiência é não tratar da disseminação de informações sumarizadas sobre as condições do tráfego.

As propostas analisadas anteriormente ainda possuem uma baixa acurácia na identificação do nível de congestionamento veicular, ou seja, basicamente detectam condições normais ou anormais do tráfego veicular. Não existe uma estimativa da intensidade do congestionamento identificado, o que pode levar a altas taxas de erros em aplicações de VANETs de controle de congestionamento. Os trabalhos discutidos a seguir propõem algoritmos e soluções baseados em inteligência computacional, para identificar e também estimar níveis (intensidade) de congestionamentos de forma mais eficiente.

Trabalhos que utilizam Lógica *Fuzzy*

O trabalho de Lu e Cao (2003) propõe um algoritmo para classificação automática do nível de congestionamento de tráfego veicular, utilizando para isso um mecanismo de inteligência computacional baseado em lógica *fuzzy*. No entanto, a proposta dos autores para identificação de tráfego veicular utiliza apenas a infraestrutura, que coleta dados do tráfego através de sensores instalados na via. Esses dados são tratados e utilizados como entrada no sistema *fuzzy* proposto.

Em (BINGLEI; ZHENG; HONGWEI, 2008), (SHAN; JIANSHEG; LING, 2012) e (BAUZA; GOZALVEZ; SANCHEZ-SORIANO, 2010), a ideia proposta em (LU; CAO, 2003) foi utilizada para aumentar a eficiência na detecção de congestionamentos. Como retratado em (BINGLEI; ZHENG; HONGWEI, 2008), o uso de limites fixos entre categorias de tráfego veicular leva a altas taxas de erros, principalmente quando uma categoria está próxima da fronteira de classificação de outras categorias. Isso ocorre devido à característica do fluxo do tráfego veicular, que não possui limites claros entre tipos de tráfegos distintos. Nesse caso, para problemas de decisão não determinísticos, lógica *fuzzy* obtém bons resultados, por permitir que variáveis de entrada do sistema contenham elementos com grau parcial de adesão, em regiões de fronteira. É essa característica do sistema que ajuda a assegurar a precisão dos resultados.

A proposta apresentada em (SHAN; JIANSHEG; LING, 2012), conforme citado, estima o nível de congestionamento utilizando lógica *fuzzy*. Contudo, a detecção de congestionamento é obtida através da análise (em tempo real) de vídeos do tráfego veicular. As variáveis de entrada do sistema, densidade e velocidade, são extraídas dos vídeos. Daí a necessidade de implantação de uma infraestrutura prévia, para filmar o tráfego veicular nas vias. Essa infraestrutura necessária limita a proposta, mas, como a infraestrutura está presente, a comunicação V2I pode ser utilizada para disseminar informações sobre as condições detectadas no tráfego. A grande contribuição desse trabalho está nos resultados

satisfatórios obtidos (utilizando lógica *fuzzy*) para mensurar o nível de congestionamento.

O COTEC (*COoperative Traffic congestion detECTION*), proposta apresentada em (BAUZA; GOZALVEZ; SANCHEZ-SORIANO, 2010), é a principal referência deste trabalho. Conforme mencionado, ele utiliza lógica *fuzzy* para aumentar a precisão na detecção do nível de congestionamento veicular. O protocolo ainda dispõe de uma abordagem para validação cooperativa (comunicação V2V) das condições do tráfego detectadas pelo sistema *fuzzy*. Essa abordagem também ajuda a reduzir a sobrecarga do canal de comunicação. Os conceitos abordados pelos autores em (BAUZA; GOZALVEZ; SANCHEZ-SORIANO, 2010) foram incorporados a este trabalho.

A abordagem para detecção cooperativa de congestionamento de tráfego do COTEC utiliza um limiar de congestionamento que é previamente definido, e assim que o nível de congestionamento local estimado pelo sistema *fuzzy* (nos veículos) superar esse limiar, o processo de detecção cooperativa de congestionamento do protocolo (que tem por objetivo conseguir uma decisão de consenso, entre veículos vizinhos, sobre a situação do tráfego na região) é ativado. Para esse processo, o COTEC utiliza mensagens de dados, que são diferentes das mensagens *beacons* citadas. Essas mensagens, chamadas de CTE (*Cooperative Traffic Estimation*), são usadas para estimativas de congestionamento de tráfego cooperativo e para veículos na região estimarem a extensão do congestionamento. Dessa forma, veículos localizados na traseira do congestionamento obtêm uma visão global do engarrafamento. Basicamente, cada veículo que estiver transmitindo uma mensagem CTE atualiza informações do tráfego, no pacote a ser retransmitido, com base nas suas próprias estimativas locais.

O mecanismo de transmissão e retransmissão de mensagens CTE no COTEC utiliza informações centradas em veículos, em que apenas veículos com suspeita de congestionamento veicular participam do processo de transmissão da mensagem. A retransmissão é finalizada quando a mensagem CTE chega em uma área não congestionada. Veículos localizados fora do engarrafamento são encarregados de disseminar informações resumidas de congestionamento à veículos em outras regiões.

Porém, no COTEC, o procedimento de encaminhamento das mensagens CTE utiliza um temporizador inversamente proporcional ao nível do sinal, ou seja, veículos que recebem mensagens CTE com nível de sinal alto precisam esperar um tempo maior para retransmitir a mensagem, e quando forem retransmitir, caso haja algum veículo mais atrás transmitindo mensagens CTE, a transmissão do veículo à frente é cancelada. Essa abordagem pode ser comparada com a proposta de (FAHMY; RANASINGHE, 2008), que possibilita o funcionamento do protocolo mesmo em ambientes restritos, onde não seja possível utilizar o GPS. Contudo, essa abordagem em ambientes urbanos apresenta problemas, uma vez que nesses cenários existem várias vias paralelas e, dessa forma, esse

processo de retransmissão tenderia a produzir dados não confiáveis de uma ZOR e não de uma determinada via. Para determinadas aplicações de minimização de congestionamento veicular essa abordagem pode ser um problema que precisa ser tratado.

O COTEC ainda possui outras deficiências mais graves que limitam aspectos positivos da proposta, tais como: (i) a ineficiência em evitar que um veículo alcance uma área congestionada e a (ii) falta de uma política de permissão de alteração de rotas. (iii) Novas configurações no sistema *fuzzy* também são necessárias para que tal técnica não se limite apenas ao cenário ideal (autoestrada) proposto pelos autores.

Nesse contexto, este trabalho propõe um protocolo de identificação e redução de congestionamento veicular, que possua todas as características apresentadas no Quadro 2. O CARTIM, aqui proposto, supera as limitações apresentadas nas referências avaliadas neste trabalho. Com todas as características listadas no Quadro 2, o CARTIM é uma proposta ciente de contexto viável para cenários heterogêneos (autoestrada e vias urbanas), que fazem parte do escopo de redes veiculares.

3 METODOLOGIA DE PESQUISA

Este capítulo apresenta a metodologia proposta, e, a Figura 12, mostra o fluxograma das etapas que foram seguidas para realização deste trabalho.

A técnica de simulação foi adotada como estratégia para modelagem computacional, capacitação e avaliação do protocolo proposto. Na simulação, as variáveis do sistema apresentam um comportamento semelhante ao ambiente real, acarretando uma sucessão de eventos cronológicos.

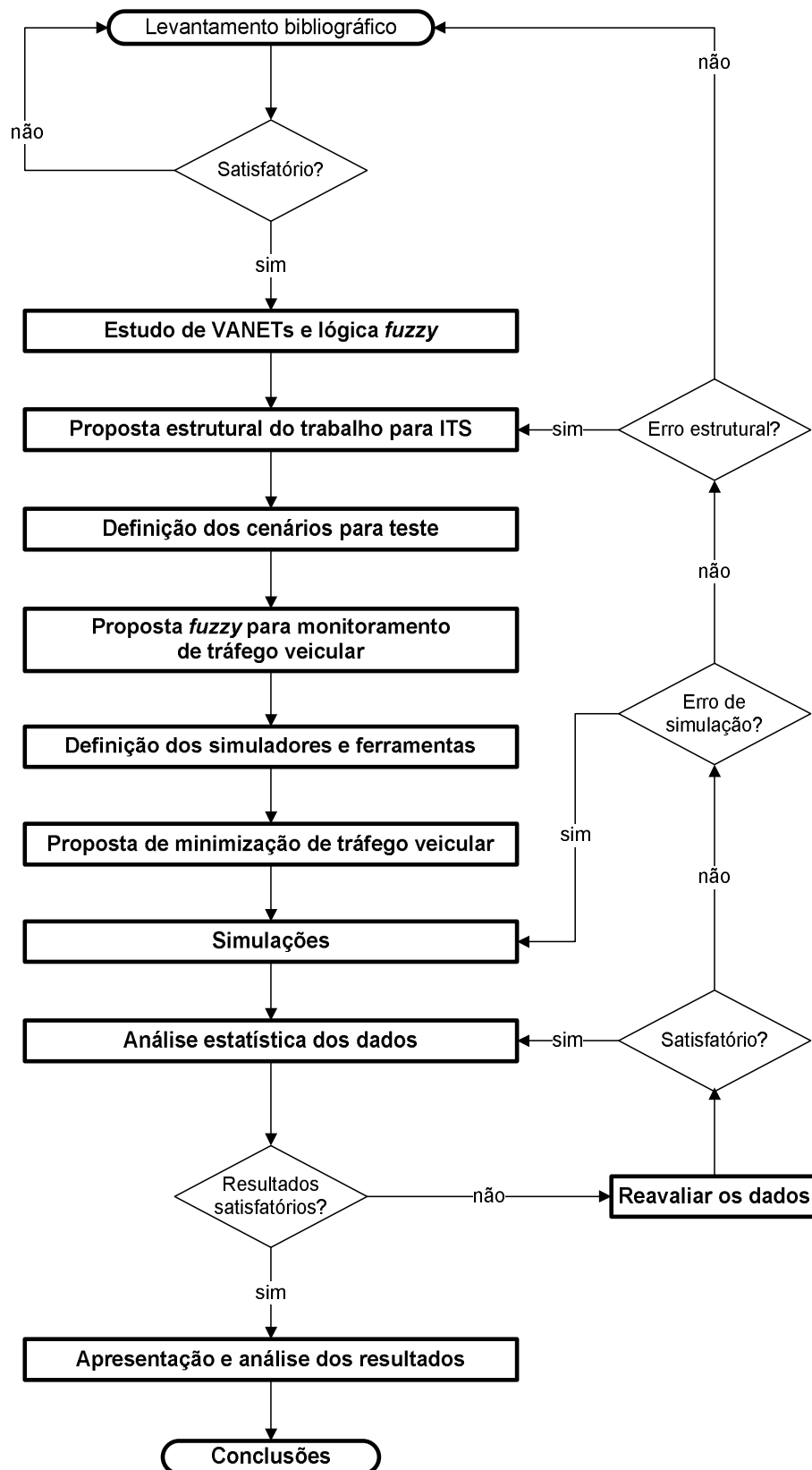
Para simulações de VANETs o ambiente foi composto pelas seguintes ferramentas: OMNET++ (VARGA; HORNIG, 2008) para simulação de rede, SUMO (KRAJZEWICZ et al., 2012) para simulação de tráfego e mobilidade, e o *framework* VEINS para comunicação bidirecional entre esses dois simuladores.

A tese de doutorado de Sommer (2011) apresenta toda arquitetura do VEINS e o grande benefício do *framework* é ter implementado o padrão IEEE 802.11p. Algoritmos para calcular a emissão de CO₂ dos veículos também são propostos e avaliados para o *framework* VEINS em (SOMMER et al., 2010). Esse cálculo se baseia em vários fatores, tais como a velocidade e a aceleração dos veículos, de acordo com a fórmula definida por Capiello et al. (2002). Também contribuiu para escolha do VEINS o fato do *framework* utilizar os simuladores de rede (OMNET++) e mobilidade (SUMO), para simulações de VANETs. A vantagem do OMNET++ é ser uma ferramenta para simulação de rede de grande escalabilidade (GAMESS; MAHGOUB; RATHOD, 2012), com boa documentação e curva de aprendizagem rápida, se comparado com ns-2 e ns-3. O SUMO, por ser um simulador de tráfego e mobilidade microscópico, e por implementar o protocolo TraCI (que permite a comunicação bidirecional com simulador de rede), é quase uma unanimidade dentre os trabalhos encontrados na literatura para simulação de tráfego e mobilidade veicular em VANETs. A Figura 13 ilustra o ambiente completo usado para simulações de VANETs.

Para criação de redes rodoviárias reais e configuração de cenários para simulações de VANETs, as seguintes ferramentas foram utilizadas: OpenStreetMap, JOSM e Traffic-Modeler. Essas ferramentas auxiliaram principalmente na geração das redes rodoviárias e da mobilidade veicular, reduzindo o tempo e os esforços necessários.

Todas as simulações e cenários propostos neste trabalho seguiram quase os mesmos parâmetros de simulação para comunicação (sem fio) entre veículos e infraestrutura. As variações nos cenários simulados foram apenas na rede rodoviária (mapa), tempo de simu-

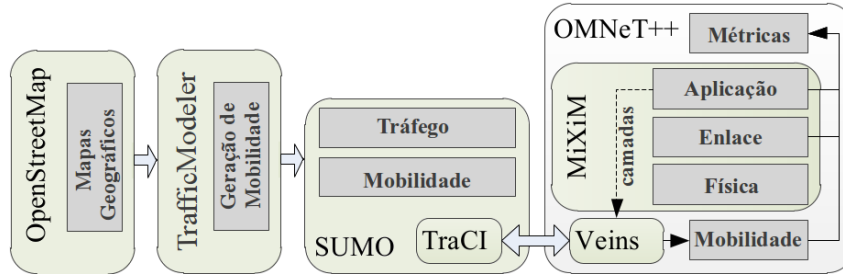
Figura 12 – Fluxograma das etapas da metodologia proposta



Fonte: Elaborada pelo autor.

lação, número de veículos e mobilidade veicular. Os cenários usados foram criados com objetivo de atestar a capacidade do protocolo proposto. Para isso, três tipos de mapas foram utilizados: (i) *Freeway*, (ii) *Manhattan Grid*, (iii) Mapa Geográfico.

Figura 13 – Ambiente usado para simulações de VANETs



Fonte: Elaborada pelo autor.

O Quadro 3 apresenta dados sumarizados das configurações realizadas nas simulações, bem como os principais parâmetros de comunicação ajustados no *framework* VEINS para os respectivos módulos, Veículos e RSUs, implementados neste trabalho. A potência de transmissão na comunicação V2V de 0.61 mW (Quadro 3) resulta em um raio de transmissão (r) aproximado de 150 metros. Na comunicação V2I o raio (r) estimado é de 300 metros.

A máquina utilizada para as simulações foi um Intel Core 2 Duo, CPU de 2.33 GHz com 2 processadores nessa configuração, memória cache de 4MB e 8GB de memória RAM. O sistema operacional utilizado foi Ubuntu GNU/Linux 12.10, com OMNET++ 4.3, Sumo 0.17.0, VEINS 2.1 e FuzzyLite 2.0.

Quadro 3 – Principais parâmetros de simulação configurados no *framework* VEINS

Veículos e RSUs	
Parâmetro	Valor
.connectionManager.carrierFrequency	5.890e9 Hz
.mac1609_4.bitrate	6Mbps
.phy80211p.sensitivity	-94dBm
.phy80211p.thermalNoise	-110dBm
Veículos	
.mac1609_4.txPower	0.61mW
.phy80211p.maxTXPower	0.61mW
RSUs	
.mac1609_4.txPower	2.20mW
.phy80211p.maxTXPower	2.20mW

Fonte: Elaborado pelo autor.

4 CARTIM

Este capítulo apresenta o CARTIM (***C**ooperative vehicul**AR** Traffic congestion Identification and **M**inimization*) em cinco seções: (i) introdução, (ii) estimativa local de tráfego veicular, (iii) validação cooperativa de congestionamento de tráfego, (iv) heurística para minimização de congestionamento de tráfego, (v) gerenciamento e manipulação de mensagens na rede.

4.1 Introdução

Sistemas de monitoramento de tráfego e detecção de congestionamentos veiculares podem utilizar uma rede centralizada, usando infraestrutura, ou distribuída, usando os próprios veículos para coletar, analisar e divulgar informação a respeito do tráfego veicular. A abordagem distribuída para detecção de congestionamento é mais eficiente, pois fornece maior grau de confiabilidade e não necessita de grande investimento em infraestrutura. CARTIM se baseia em uma solução distribuída para detecção de congestionamento veicular, em que 100% dos veículos devem implementar o protocolo. A infraestrutura, quando presente, é usada apenas para disseminar mensagens de congestionamento em uma ZOR maior que as alcançadas na comunicação V2V. Assim, a infraestrutura é transparente para o protocolo.

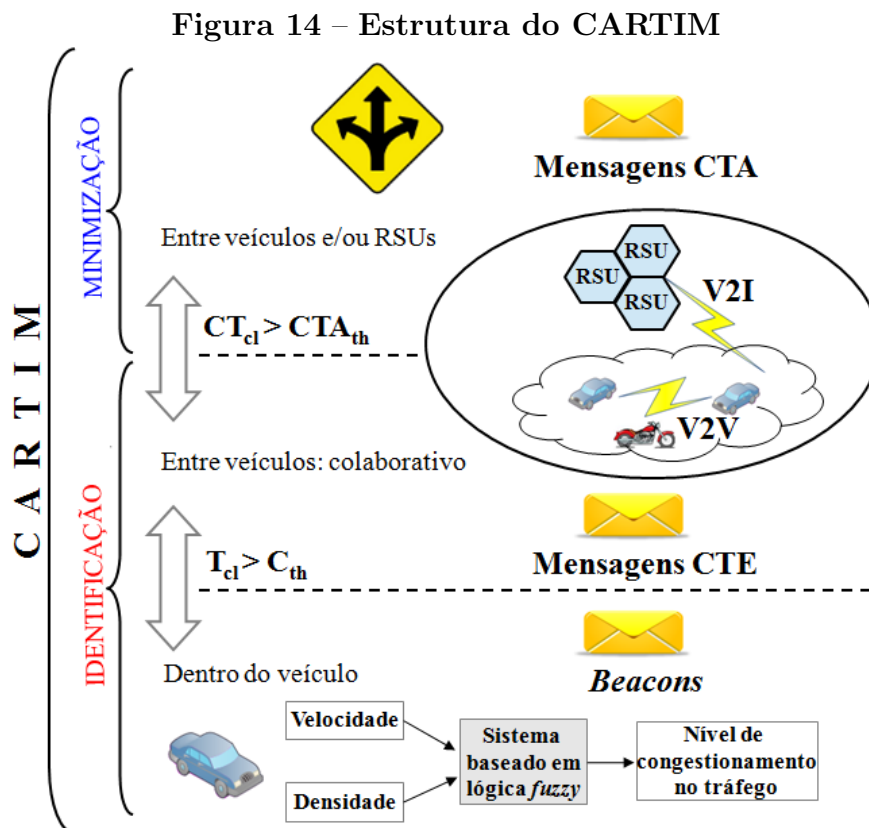
Para troca de informações entre veículos e validação cooperativa de congestionamento, o CARTIM usa dois tipos de mensagens: (i) *beacon* e (ii) *data message*. As mensagens *beacons* possuem baixo *overhead* e são usadas para anunciar a presença do veículo aos seus vizinhos. As mensagens de dados possuem *overhead* maior, pois acrescentam dados relativos à situação do tráfego. O CARTIM possui dois tipos de mensagens de dados: (i) CTE (*Cooperative Traffic Estimation*) e (ii) CTA (*Cooperative Traffic Alert*). A carga útil de um *beacon* é de 16 bytes, e no caso das mensagens de dados a carga útil é de 128 bytes. Para evitar sobrecarga do canal de comunicação, o CARTIM adota estratégias diferentes para disseminação das mensagens *beacons* e de dados na rede.

Existem duas propostas muito utilizadas na literatura para disseminação de mensagens *beacons* na rede: *flooding* e *diffusion* (MUHLETHALER; LAOUITI; TOOR, 2007), (SOMMER; TONGUZ; DRESSLER, 2011). No mecanismo de *flooding*, cada veículo emite informações sobre si mesmo, e sempre que um veículo recebe uma mensagem por *broadcast*, ele a armazena e encaminha (*store and forward*) imediatamente por *rebroadcast*. Esse método não é escalável (por ter mensagens inundando a rede) e, por isso, inadequado

para cenários com alta densidade de veículos. No outro mecanismo, *diffusion*, cada veículo também transmite informações por *broadcast* sobre si mesmo mas, periodicamente, em intervalos específicos de tempo. E, sempre que um veículo recebe mensagens por *broadcast*, ele apenas atualiza suas informações para o próximo período de transmissão. Portanto, o mecanismo de difusão é escalável, pois o número de mensagens transmitidas na rede é limitada, o que não ocorre no *flooding*, que pode causar sobrecarga do canal de comunicação. O CARTIM usa apenas o mecanismo *diffusion* para transmitir *beacons*.

As mensagens de dados usadas pelo CARTIM, para validar colaborativamente condições de tráfego congestionado, são transmitidas em intervalos fixos de tempo e apenas quando um determinado veículo detecta situação de tráfego anormal. Sob condições de tráfego normal não existe transmissão desses tipos de mensagens, evitando assim sobrecarregar o canal de comunicação.

A Figura 14 ilustra sucintamente, em alto nível, a estrutura do CARTIM. É possível observar a proposta de minimização, identificação local e colaborativa de congestionamento do protocolo. Para detecção local e identificação (do nível) de congestionamento veicular o CARTIM implementa um sistema baseado em lógica *fuzzy*. Quando esse nível de congestionamento de tráfego (T_{cl} – *Traffic congestion level*) superar o limiar de congestionamento local (C_{th} – *Congestion threshold*), o protocolo inicia o processo colaborativo de validação das condições detectadas.



Fonte: Elaborada pelo autor.

As mensagens CTE são utilizadas para estimar níveis de congestionamento de tráfego cooperativo (CT_{cl} – *Cooperative Traffic congestion level*), e ao superarem o limiar cooperativo de tráfego (CTA_{th} – *CTA threshold*), essas mensagens se transformam em mensagens CTA. Veículos no raio de comunicação ao escutarem tais mensagens na rede alteram suas rotas iniciais, evitando passar na via congestionada. Quando a situação do tráfego não estiver mais sob condições severas, as mensagens CTA param de ser disseminadas. E, dessa forma, o protocolo regula dinamicamente a política de alteração de rota. Nas seções subsequentes os detalhes da estrutura do CARTIM serão apresentados.

4.2 Estimativa Local de Tráfego Veicular

O CARTIM utiliza a arquitetura V2V para estimar o nível de congestionamento do tráfego. As informações são obtidas através da troca periódica de mensagens *beacons* entre os veículos, em que cada veículo informa aos seus vizinhos sua velocidade e sua localização atual. Para não sobrecarregar o canal de comunicação com mensagens *beacons*, a frequência de transmissão dessas mensagens no protocolo é ajustada para 0.5 Hz, ou seja, a cada 2 segundos o veículo envia um *beacon* por *broadcast*. Como mostra a Equação 1, em que f representa a frequência e T o período (em segundos) para transmissão de uma mensagem *beacon* na rede.

$$f(Hz) = \frac{1}{T(s)} \quad (1)$$

O processo de identificação de congestionamento veicular não é uma tarefa simples. Informações qualitativas são tratadas como, por exemplo, densidade baixa de veículos. Os trabalhos de (LU; CAO, 2003), (BINGLEI; ZHENG; HONGWEI, 2008), (SHAN; JIANSHENG; LING, 2012) e (BAUZA; GOZALVEZ; SANCHEZ-SORIANO, 2010) utilizam lógica *fuzzy* para detecção do nível de congestionamento de tráfego. Fundamentado nesses trabalhos, o CARTIM também utiliza lógica *fuzzy*.

Como em qualquer sistema de decisão baseado em lógica *fuzzy*, as variáveis possuem valores linguísticos e são separadas em conjuntos *fuzzy*. A classificação desses conjuntos no CARTIM se baseia nas referências do HCM (*Highway Capacity Manual*) (BOARD, 2000). O manual de capacidade das estradas contém conceitos, diretrizes e procedimentos computacionais para medir a capacidade, o desempenho e a qualidade do tráfego, baseado na velocidade e densidade de veículos nas vias.

O sistema *fuzzy* proposto para o CARTIM também utiliza como parâmetros de entrada (valores não-*fuzzy*) a velocidade instantânea (V_{inst}) do veículo e a densidade veicular (ρ) da via. A velocidade é um atributo simples de se obter e pode ser calculada através da

distância percorrida ($\Delta D = D - D_0$) em um intervalo curto de tempo ($\Delta t = t - t_0$), conforme pode ser observado na Equação 2. No CARTIM, esse intervalo curto de tempo Δt é definido automaticamente pelo protocolo, pois está vinculado à frequência de transmissão de *broadcast beacons* do protocolo (0.5 Hz no caso), e, portanto $\Delta t = 2(s)$.

$$V_{inst}(m/s) = \frac{\Delta D(m)}{\Delta t(s)} \quad (2)$$

A obtenção da densidade de veículos vizinhos na via é uma tarefa complexa. Com base no número de mensagens *beacons* recebidas, cada veículo estima a quantidade de veículos vizinhos (N) em um intervalo de tempo (Δt). Assim, é possível calcular a densidade de veículos, que corresponde ao número de veículos por metro (unidade de simulação), por faixa (F). A Equação 3 mostra a fórmula usada para obtenção da densidade de vizinhos. Nela, r corresponde a um raio de comunicação V2V de aproximadamente 150 metros, de acordo com o Quadro 3.

$$\rho = \frac{N}{r(m) \times F} \quad (3)$$

As classificações das variáveis linguísticas dos conjuntos *fuzzy* de entrada do sistema para a velocidade são: *muito lenta*, *lenta*, *moderada*, *rápida*; e para densidade são: *baixa*, *média*, *alta*, *muito alta*. O conjunto *fuzzy* de saída varia entre $[0,1]$ e, no CARTIM, esse conjunto de saída adota a seguinte classificação: 0 - equivale a fluxo *livre*, 1/3 - equivale a fluxo *fraco*, 2/3 - equivale a fluxo *moderado*, 1 - equivale a condições *severas* de tráfego.

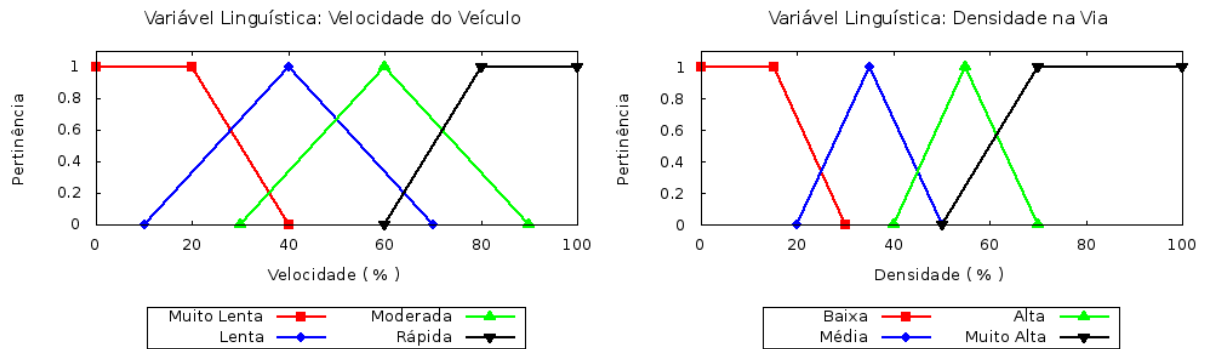
Quadro 4 – Regras *fuzzy* para o CARTIM

Nível de Congestionamento	Densidade Veicular			
	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>	<i>Muito Alta</i>
<i>Muito Lenta</i>	fraco	moderado	moderado	severo
<i>Lenta</i>	livre	fraco	moderado	moderado
<i>Moderada</i>	livre	fraco	fraco	moderado
<i>Rápida</i>	livre	livre	livre	fraco

Fonte: Adaptado de Bauza, Gozálvez e Sanchez-Soriano (2010).

O Quadro 4 apresenta as regras *fuzzy* adotadas para obtenção do nível de congestionamento do sistema, de acordo com o respectivo conjunto *fuzzy* de entrada. As Figuras 15 (a) e (b) mostram as funções de pertinência das variáveis linguísticas associadas aos conjuntos *fuzzy* de entrada do sistema, em que é possível verificar o grau parcial de adesão dessas funções, nas regiões de fronteira. O conjunto *fuzzy* de saída, ilustrado na Figura 16, também mostra as funções de pertinência da variável linguística congestionamento. A Figura 17 permite visualizar a estrutura para detecção do nível de congestionamento. Por exemplo, condições *severas* de congestionamento ocorrem apenas quando a velocidade do veículo está *muito lenta* e a densidade de vizinhos na via está *muito alta*.

Figura 15 – Funções de pertinência dos conjuntos *fuzzy* de entrada



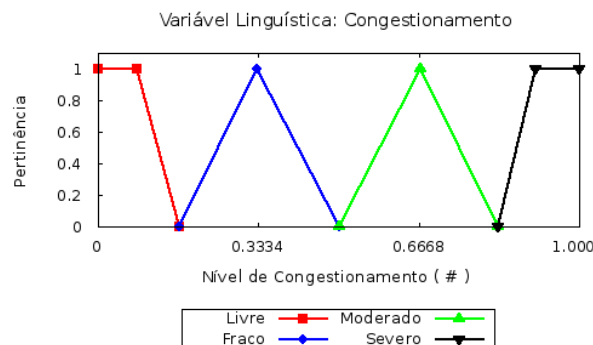
(a) Conjunto *fuzzy* de entrada – velocidade

(b) Conjunto *fuzzy* de entrada – densidade

Fonte: Elaborada pelo autor.

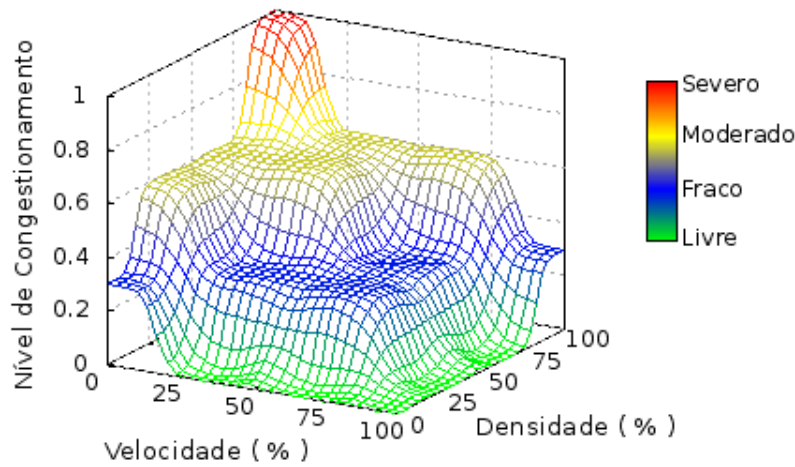
Como a definição das funções de pertinência das variáveis linguísticas do sistema *fuzzy* não é uma tarefa trivial, este trabalho se baseou na referência (BOARD, 2000) e no processo desenvolvido em (SKYCOMP, 2008) para classificar e caracterizar a intensidade (nível) dos congestionamentos veiculares. Para analisar os dados do tráfego veicular de diferentes rodovias, recolhidos através de levantamentos aéreos, a Skycomp Incorporated (SKYCOMP, 2014) propõe uma classificação para as condições do tráfego veicular baseada nos níveis de serviços (LOS – *Level Of Service*) apresentados em (BOARD, 2000). A métrica (LOS) representa uma estimativa para descrever a qualidade das condições do tráfego, e há seis categorias diferentes para ela, de *A* a *F*. Os níveis de serviços *A* até *E* representam condições distintas do tráfego, mas sob fluxo livre (não congestionado). O LOS *F* é o único que descreve situações concretas de congestionamento do tráfego veicular, mas a categorização proposta pelo HCM não faz distinção entre os diferentes níveis de congestionamento ocorridos no LOS *F*. Por outro lado, a abordagem proposta pela Skycomp estende a classificação do HCM, mais especificamente para o LOS *F* e, assim, estipula métricas (baseadas na velocidade e densidade veicular) para avaliar e diferenciar níveis de congestionamento.

Figura 16 – Funções de pertinência para nível de congestionamento



Fonte: Elaborada pelo autor.

Figura 17 – Nível de congestionamento obtido pelo sistema



Fonte: Elaborada pelo autor.

A Tabela 2 mostra a classificação estendida (do LOS F) proposta pela Skycomp. Nessa tabela, as métricas usadas para classificação das condições do tráfego são apresentadas. Como um dos objetivos deste trabalho era identificar congestionamento veicular, a proposta apresentada pela Skycomp foi adaptada para configuração das funções de pertinência das variáveis linguísticas do sistema *fuzzy* proposto para CARTIM.

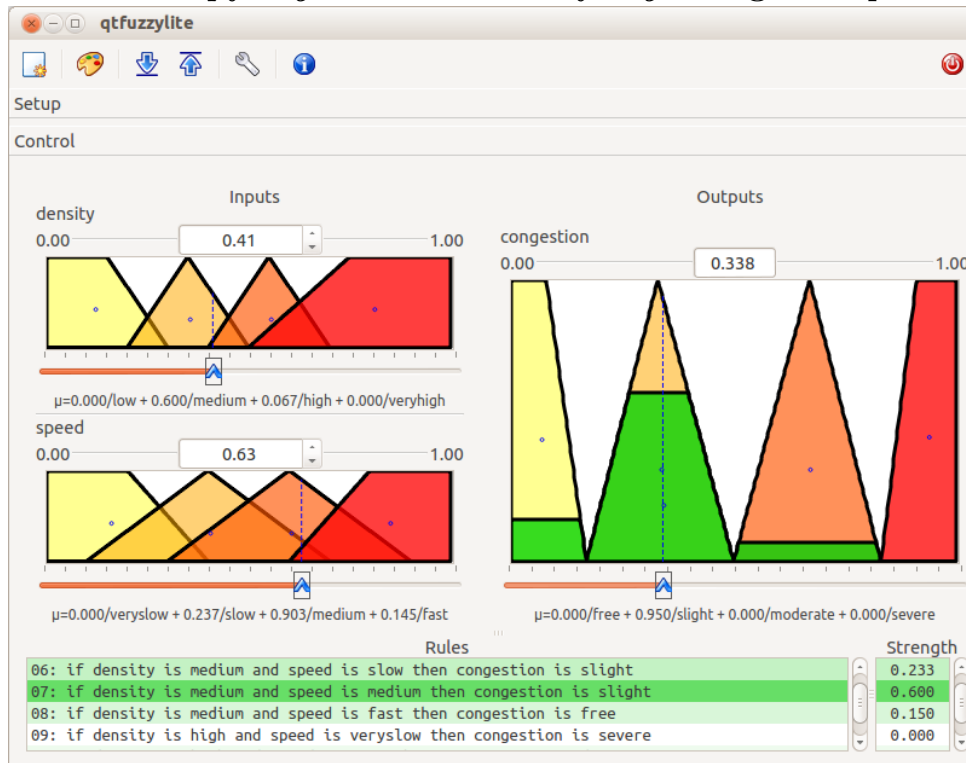
Contudo, um grande benefício do sistema *fuzzy* proposto para o CARTIM, e ilustrado na Figura 17, está relacionado à conversão dos valores das variáveis linguísticas de entrada do sistema para porcentagem. Desse modo, cada veículo estima o valor da velocidade baseado na razão entre sua velocidade momentânea e a velocidade máxima permitida na via. A densidade de veículos vizinhos no CARTIM se baseia na ocupação do segmento de estrada que o veículo percorre no momento. Utilizando o raio de transmissão (alcançado na comunicação V2V para disseminação de mensagens), em metros, estima-se a quantidade máxima de veículos por faixa no segmento. Assim, cada veículo, de acordo com a razão entre o número de vizinhos momentaneamente no segmento e a quantidade máxima de veículos suportada no local, consegue estimar a ocupação do segmento em porcentagem. A utilização dessa abordagem pelo CARTIM faz com que o protocolo consiga se adaptar e ajustar o sistema *fuzzy* dinamicamente de acordo com o ambiente (contexto).

Tabela 2 – Nível de congestionamento proposto pela Skycomp

Nível de Congestionamento		Velocidade		Densidade	
		mi/h	km/h	veí/mi/ln	veí/km/ln
L. O. S. F.	Leve	30-50	48-81	46-60	29-37
	Fraco	15-40	24-64	60-80	37-50
	Moderado	10-25	16-40	80-100	50-63
	severo	≤ 25	≤ 40	≥ 100	≥ 63

Fonte: Adaptada de Skycomp (2008).

Figura 18 – Tela do *qtffuzzylite* com sistema *fuzzy* configurado para CARTIM



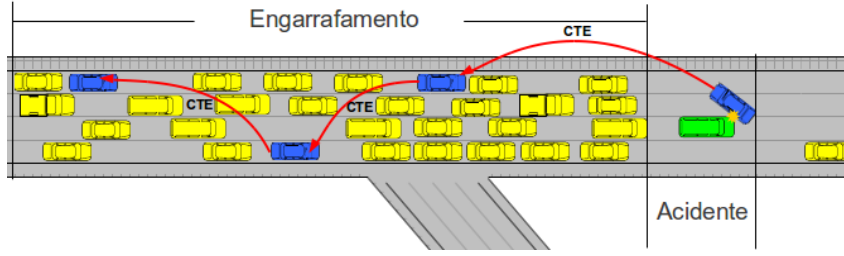
Fonte: Criada pelo autor através do qtffuzzylite.

A Figura 18 mostra a tela do *qtffuzzylite* com o sistema *fuzzy* configurado para o CARTIM, com as devidas variáveis linguísticas, funções de pertinência, conjuntos e regras *fuzzy*. Nessa figura (sob um cenário qualquer de exemplo) podemos observar o comportamento do sistema, em que, com velocidade de 63% da máxima permitida na via e uma ocupação (densidade) de 41% da via, a saída (não-*fuzzy*) do sistema é de aproximadamente 1/3 (0.338), que equivale a condições de fluxo fraco, de acordo com a classificação proposta pelo CARTIM.

4.3 Validação Cooperativa de Congestionamento de Tráfego

O CARTIM usa um processo colaborativo (comunicação V2V) para validar o nível local de congestionamento detectado pelo sistema *fuzzy*. Para isso, são utilizadas mensagens de dados – chamadas CTE (*Cooperative Traffic Estimation*), conforme mencionado anteriormente. Essas mensagens apresentadas em (BAUZA; GOZALVEZ; SANCHEZ-SORIANO, 2010), usam um limiar de congestionamento (C_{th} - *Congestion threshold*) que é estipulado previamente, e, assim que atingido, um processo colaborativo de validação das condições detectadas se inicia. A transmissão das mensagens cooperativas ocorre exclusivamente da extremidade dianteira do congestionamento para a extremidade traseira, através de um encaminhamento *multi-hop*, *unicast*, como pode ser observado no exemplo ilustrado na Figura 19.

Figura 19 – Disseminação de mensagens CTE em um cenário de exemplo sob tráfego congestionado



Fonte: Elaborada pelo autor.

Para estimar de forma cooperativa o nível de congestionamento veicular, o CARTIM usa apenas veículos que retransmitem a mensagem CTE, em que cada veículo usa sua quantidade de vizinhos (N) para calcular a média ponderada do nível de congestionamento. Dessa forma, a proposta procura representar as condições detectadas pelos demais veículos na via, sem aumentar o *overhead* de comunicação. A Equação 4, proposta para o CARTIM, mostra o cálculo realizado para se estimar cooperativamente o nível de congestionamento ($\bar{x}_p$). Nela, x_i corresponde ao nível de congestionamento entre $[0,1]$ estimado pelo veículo que participa do processo de retransmissão.

$$\bar{x}_p = \frac{\sum_{i=1}^n (N \times x_i)}{\sum_{i=1}^n N} \quad (4)$$

As mensagens CTE também são utilizadas para estimar o tamanho de um congestionamento, de acordo com a posição inicial do primeiro veículo que a transmitiu. Assim, a descoberta do veículo na dianteira do congestionamento é uma tarefa importante para o funcionamento do CARTIM. Tal veículo, além de ser o responsável pela frequência (Hz) de geração de mensagens CTE na via congestionada, serve como referencial aos demais veículos na pista, para cálculo da extensão do engarrafamento.

A alta mobilidades dos nós, as mudanças na topologia da rede e as desconexões frequentes são características já mencionadas de VANETs, que podem levar a altas taxas de perdas no envio de mensagens *unicast*. Além disso, pelo dinamismo e para não sobrecarregar a rede, o CARTIM não utiliza mensagens ACK (*acknowledgement*) para confirmação do recebimento das mensagens CTE. Portanto, elevadas taxas de perda de mensagens comprometem o funcionamento do protocolo, uma vez que a comunicação fica interrompida. Para envio de mensagens *unicast*, em situações de tráfego congestionado, o CARTIM propõe uma nova técnica (**CTDiss** – *Congestion Traffic Dissemination*) para disseminação de mensagens CTE na rede veicular de forma mais eficiente.

A CTDiss realiza uma avaliação para identificar qual o veículo vizinho, atrás do congestionamento, é o mais adequado para encaminhar a mensagem *unicast*. A Figura 20 (a) ilustra um cenário em que o veículo 8 é o mais distante em relação ao veículo 1, considerando-se o raio (r) de comunicação V2V. Tal veículo, no entanto, não é apontado

como o veículo ideal para retransmissão da mensagem CTE. O veículo 6, considerado ideal, é obtido pela técnica através da Equação 5, que calcula dinamicamente a distância ótima (λ) do veículo, para retransmissão da mensagem. Essa distância, ilustrada pela Figura 20 (b), obtida em metros (unidade na simulação), é calculada subtraindo-se do tamanho do raio de transmissão (r) dos veículos a razão entre a velocidade máxima permitida na via (m/s) e a frequência (Hz) de transmissão das mensagens *beacons*.

$$\lambda(m) = r(m) - \frac{v(m/s)}{f(Hz)} \quad (5)$$

Por exemplo, veículos com a frequência de transmissão de *broadcast beacons* ajustada em 0.5 Hz e com raio de transmissão de 150 metros, trafegando em uma via qualquer, cuja velocidade máxima permitida seja de 16,67 m/s (60 km/h), obtêm a distância ótima de aproximadamente 116,67 metros. No CARTIM, esse cálculo é necessário para encontrar o veículo vizinho mais próximo dessa distância de referência, a fim de retransmitir a mensagem *unicast*. Apesar das mensagens cooperativas de tráfego serem *unicast*, veículos no raio de transmissão escutam tais mensagens e obtêm seu conteúdo.

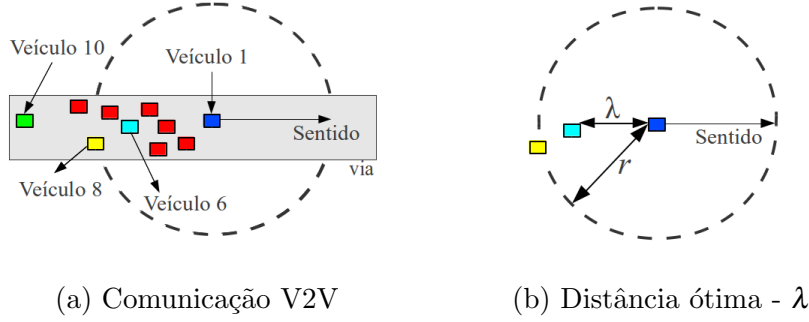
Utilizando essa abordagem, o CARTIM minimiza problemas causados pelas constantes mudanças na topologia da rede. O protocolo considera que, mesmo que o veículo selecionado para retransmissão da mensagem se desloque na via com a velocidade máxima permitida, ele ainda tem alta probabilidade de estar dentro do raio de transmissão do veículo atual, durante o intervalo de tempo de um ciclo de transmissão de mensagem *beacon*. Esse intervalo de tempo é o mesmo utilizado para descoberta dos vizinhos.

As mensagens CTE possuem um alto *overhead* quando comparadas às mensagens *beacons*. Para evitar sobrecarga do canal de comunicação, tais mensagens só são enviadas quando o nível de congestionamento local no veículo é superior ao C_{th} e em uma frequência de 0.1 Hz (ou seja, a cada 10s). Além disso, apenas veículos com nível de congestionamento maior que C_{th} participam do processo de retransmissão e, caso já exista algum veículo à frente transmitindo, o veículo de trás cancela sua transmissão. O processo de retransmissão na rede termina quando as mensagens chegam em uma região onde o nível de congestionamento local detectado pelos veículos é inferior ao C_{th} .

4.4 Heurística para Minimização de Congestionamento de Tráfego

Uma heurística para redução de congestionamento de tráfego é adotada pelo CARTIM. O objetivo da heurística é tentar reduzir o tempo gasto no trânsito pelos motoristas, sugerindo alterações nas rotas quando começam a ocorrer condições severas de congestionamento em uma determinada via. Para evitar o surgimento de novos pontos de congestionamento no tráfego, o protocolo adota uma política para alteração das rotas.

Figura 20 – CTDiss, processo de retransmissão de mensagens unicast

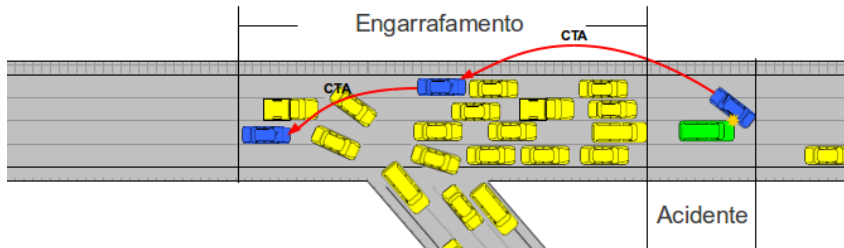


Fonte: Elaborada pelo autor.

A proposta de minimização de congestionamento também utiliza o processo colaborativo para validação das condições detectadas. A mensagem CTE, ao atingir nível de congestionamento cooperativo maior que o limiar CTA_{th} (*CTA threshold*), se transforma em mensagem CTA (*Cooperative Traffic Alert*). Veículos no raio de transmissão, ao capturarem tais mensagens na rede, poderão alterar suas rotas iniciais, evitando passar na via sob condições severas. A minimização de congestionamento do CARTIM pode utilizar as arquiteturas V2V e V2I. Contudo, a infraestrutura, composta pelas RSUs, é transparente para o protocolo. Caso esteja presente, a função da RSU é escutar mensagens CTA na rede e enviar *broadcast* dessas mensagens em uma região maior que as alcançadas na comunicação V2V. Caso a infraestrutura não esteja presente, apenas a comunicação V2V é utilizada. A Figura 21 ilustra um cenário (sem infraestrutura) em que veículos ao escutarem mensagens CTA na rede alteram suas rotas iniciais. Para mudança e geração da nova rota, o CARTIM adota o algoritmo de Dijkstra (CORMEN et al., 2009) com a finalidade de encontrar o caminho (até o destino) mais curto na rede rodoviária.

No CARTIM, quando o nível de congestionamento em uma via não estiver mais sob condições severas, as mensagens CTA param de ser disseminadas. Assim, o protocolo regula dinamicamente os alertas para alteração das rotas, procurando, desse modo, evitar o surgimento de novos pontos de congestionamento no tráfego.

Figura 21 – Disseminação de mensagens CTA em um cenário de exemplo sob tráfego severo

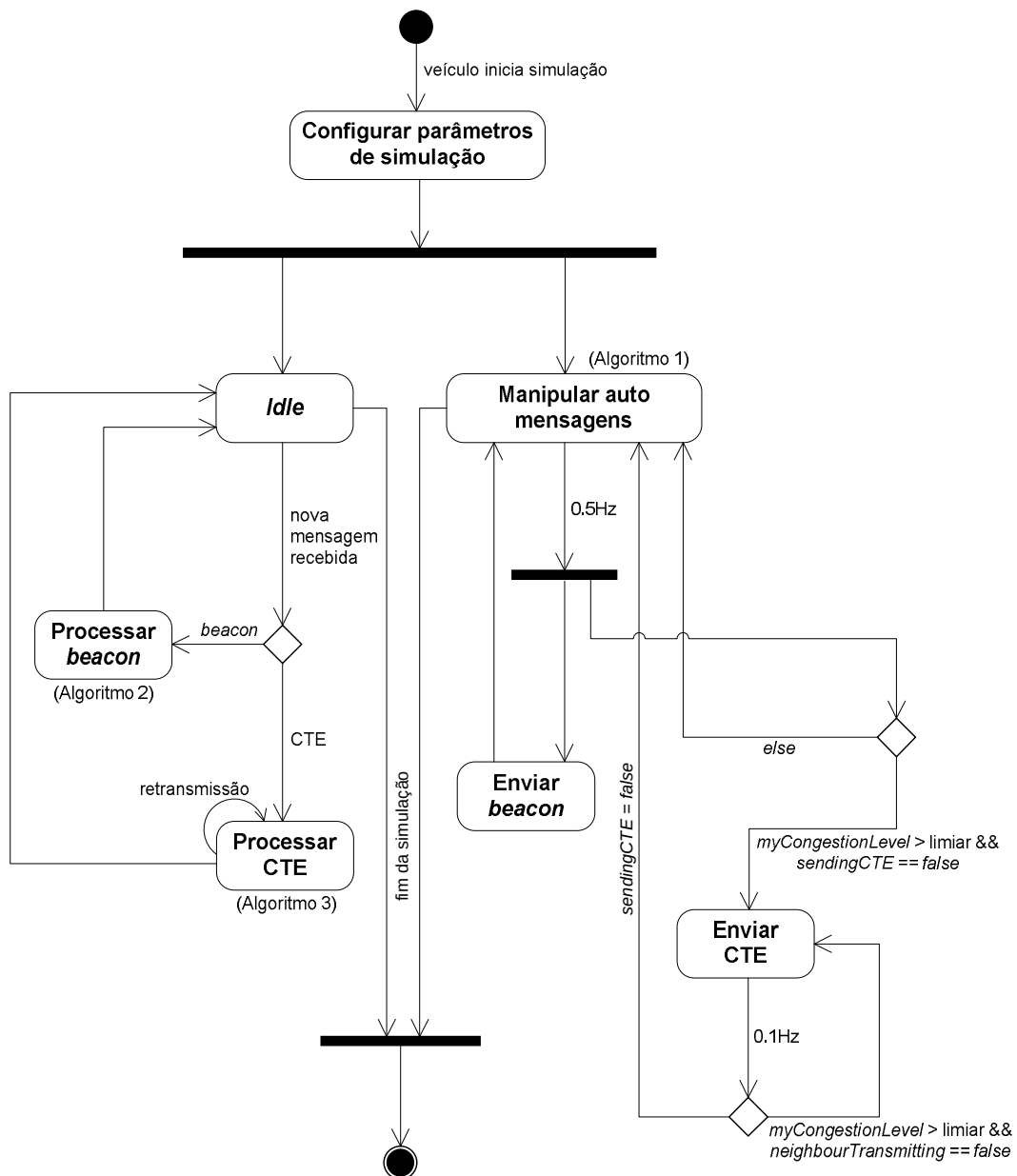


Fonte: Elaborada pelo autor.

4.5 Gerenciamento e Manipulação de Mensagens na Rede

Para gerenciar e controlar o processo de envio e recebimento de mensagens *beacons* e CTE, o CARTIM segue o diagrama de atividades da UML ilustrado na Figura 22. Nesse diagrama, o fluxo de atividades mostra, de forma abstrata, como o protocolo reage a diferentes eventos relacionados a troca de mensagens na rede. A atividade mais complexa, manipular auto mensagens, deve seguir o Algoritmo 1 (*HandleSelfMsg*) para implementação no OMNET++. Os Algoritmos 2 e 3 também estão identificados nesse diagrama.

Figura 22 – Diagrama de atividades - Controle de mensagens na rede



Fonte: Elaborada pelo autor.

O Algoritmo 1 (*HandleSelfMsg*), em alto nível, codifica o processo de manipulação de auto mensagens ilustrado na Figura 22. Nesse algoritmo, os veículos constantemente (a cada 0.5 Hz) enviam *beacons* por *broadcast* na rede (informando basicamente sua localização atual), e caso o nível de congestionamento estimado pelo sistema *fuzzy* seja superior ao limiar previamente estabelecido, se inicia o processo de envio de mensagens CTE. No entanto, se algum veículo já estiver à frente do congestionamento transmitindo mensagens CTE, o veículo atual cancela o envio dessas mensagens.

Algoritmo 1: *HandleSelfMsg*

```

1: Event: new message schedule
2: Input: message msg
3: Output:  $\emptyset$ 
4: typeOfMsg = msg  $\rightarrow$  getkind();
5: switch typeOfMsg do
6:   case SEND_BEACON_EVT :
7:     beacon  $\leftarrow$  createBeaconMessage();
8:     sendMessage(beacon);
9:     myDensity  $\leftarrow$  getTrafficDensity(beaconFrequency);
10:    mySpeed  $\leftarrow$  getVehSpeed();
    // Este método retorna o nível de congestionamento baseado em LF
11:    myCongestionLevel  $\leftarrow$  fuzzyLogic.getCongLevel(mydensity, myspeed);
    // Se nível estimado for maior que o limiar de congestionamento
    // e o veículo atual não estiver enviando mensagens CTE na rede
12:    if myCongestionLevel > cth && !sendingCTE then
13:      sendingCTE  $\leftarrow$  true;
14:      scheduleAt(simtime(), sendCTEEvt);
15:    end if
16:    scheduleAt(simtime() + beaconFrequency, sendBeaconEvt);
17:  end case
18:  case SEND_CTE_EVT :
    // Se nível estimado for maior que o limiar de congestionamento
    // e não houver nenhum veículo afrente enviando mensagens CTE
19:    if myCongestionLevel > cth && !neighbourTransmitting() then
20:      CTEMessage  $\leftarrow$  createCTEMessage();
21:      sendMessage(CTEMessage);
22:      scheduleAt(simtime() + cteFrequency, sendCTEEvt);
23:    else
24:      sendingCTE  $\leftarrow$  false;
25:    end if
26:  end case
27: end switch

```

Fonte: Elaborado pelo autor.

Os Algoritmos 2 e 3 são executados quando veículos escutam uma nova mensagem na rede. Esses veículos, nesse exato momento, saem do estado inativo (*idle*) e processam a nova mensagem recebida, *beacon* ou CTE.

Algoritmo 2: Processar Mensagem *beacon*

```

1: Event: beacon B received from neighbors
2: Output:  $\emptyset$ 
3: neighbour N is type of data;
4: myNeighbours[] is a neighbours list;
5:  $N \rightarrow id = B \rightarrow \text{getSenderAddress}();$ 
6:  $N \rightarrow Position = B \rightarrow \text{getSenderPosition}();$ 
7:  $N \rightarrow time = B \rightarrow \text{getSenderTime}();$ 
   // Armazena os dados do vizinho na base de dados
8:  $\text{myNeighbours}[N \rightarrow id] = N;$ 

```

Fonte: Elaborado pelo autor.

Algoritmo 3: Processar Mensagem CTE

```

1: Event: data message CTE received from neighbors
2: Output:  $\emptyset$ 
3: if  $\text{myRoad} == \text{CTE} \rightarrow \text{getRoad}()$  then
   // Veículos estimam o tamanho total do congestionamento
4:    $\text{myCongestionLenght} = \text{calcDistance}(\text{msg} \rightarrow \text{getPosition}(), \text{currentPosition});$ 
   // Se o destinatário da mensagem for o veículo atual
5:   if  $\text{CTE} \rightarrow \text{getDestination}() == \text{myId}$  then
   // Se nível estimado for maior que limiar de congestionamento
6:     if  $\text{myCongestionLevel} > c_{th}$  then
       // Obtêm veículo ideal para retransmitir a mensagem
       // CTE, baseado na técnica CTDissemination
7:        $\text{dstId} = \text{CTDiss}();$ 
8:        $\text{CTE} \rightarrow \text{setDestination}(\text{dstId});$ 
       // Calcula nível de congestionamento cooperativamente
9:        $\text{CooperativeCongestionEstimation} = \text{estimateCongestion}(\text{cte});$ 
       // Atualiza nível de congestionamento cooperativo na mensagem CTE
10:       $\text{CTE} \rightarrow \text{setCCE}(\text{CooperativeCongestionEstimation});$ 
11:       $\text{sendMessage}(\text{CTE});$ 
12:     end if
13:   else
       // Armazena e atualiza os dados do veículo afrente
       // transmitindo mensagens CTE na via
14:      $\text{saveNeighbourTransmitting}(\text{CTE});$ 
15:   end if
16: end if
   // Verifica política para alteração da rota inicial
17: if  $\text{politicChangeRoute}(\text{CTE} \rightarrow \text{getCCE}())$  then
18:    $\text{myRoute} = \text{commandChangeRoute}(\text{CTE} \rightarrow \text{getRoad}());$ 
19: end if

```

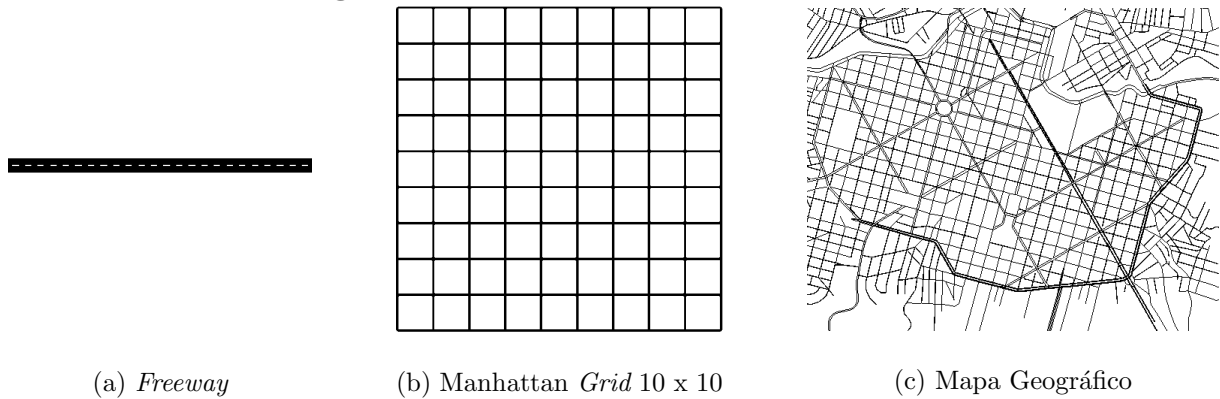
Fonte: Elaborado pelo autor.

O Algoritmo 2 é o responsável por processar mensagens *beacons*, atualizando a base de dados com informações sobre os vizinhos. O Algoritmo 3 processa apenas mensagens CTE, atualizando informações sobre a situação do tráfego, como por exemplo tamanho total do congestionamento na via e nível de congestionamento cooperativo. Caso o veículo atual seja o responsável pela retransmissão da mensagem CTE recebida, ele chama a função CTDiss, que identifica o vizinho ideal para encaminhamento da mensagem. No Algoritmo 3, caso a política adotada pelo CARTIM para alteração das rotas seja satisfeita, o veículo altera sua rota inicial.

5 SIMULAÇÕES, AVALIAÇÃO E RESULTADOS DO PROTOCOLO

Simulações foram conduzidas para comparar o sistema *fuzzy* do CARTIM, apresentado no Capítulo 4, com o do COTEC. A finalidade é verificar o comportamento dos sistemas e atestar a eficácia de cada um em detectar congestionamentos na rede. Para testes e validação do CARTIM foi criada a rede rodoviária *Freeway* (Figura 23 (a)). Em seguida, simulações foram realizadas para mostrar como a adoção do CARTIM pode auxiliar na minimização de congestionamentos de tráfego veicular. Para isso, duas outras redes rodoviárias foram criadas: *Manhattan Grid* e Mapa Geográfico. A Figura 23 ilustra as três redes rodoviárias usadas nos cenários simulados.

Figura 23 – Redes rodoviárias simuladas



Fonte: Criada pelo autor através do SUMO.

Os parâmetros usados e ajustados no *framework* VEINS para comunicação V2V e V2I foram discutidos no Capítulo 3 e apresentados no Quadro 3. O Quadro 5 mostra os parâmetros configurados especificamente sobre o CARTIM, para realização das simulações. Os cenários utilizados para simulação tiveram como escopo verificar se os objetivos propostos por este trabalho foram alcançados. Nas seções subsequentes, são apresentados os cenários simulados, as redes rodoviárias, as métricas usadas para comparação e avaliação do CARTIM juntamente com os resultados obtidos.

Quadro 5 – CARTIM, parâmetros de simulação

CARTIM	
Parâmetro	Valor
<i>Beacon frequency</i>	0.5 Hz
<i>CTE frequency</i>	0.1 Hz
<i>Congestion threshold</i> (C_{th})	0.5
<i>CTA threshold</i> (CTA_{th})	0.8
<i>Change route</i>	Active
<i>Vehicle average space</i>	10 m

Fonte: Elaborado pelo autor.

5.1 Comparação do Sistema *Fuzzy*

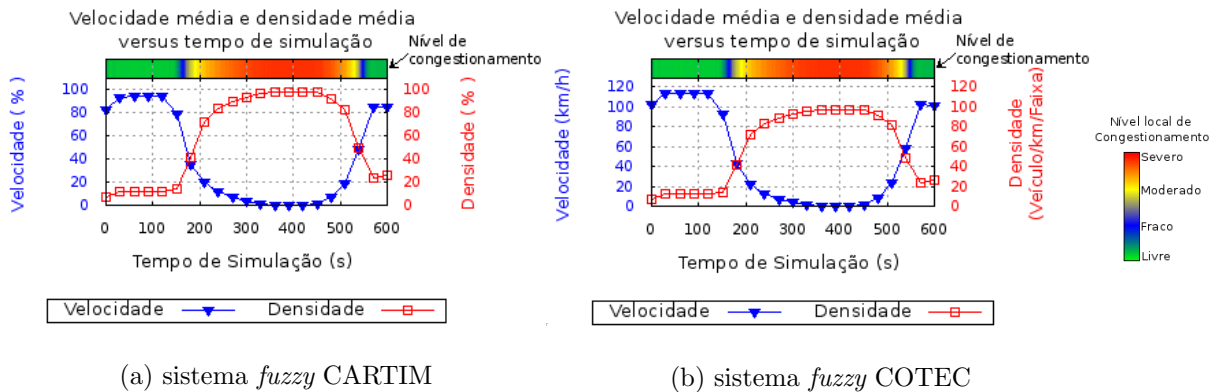
Para comparar o sistema *fuzzy* dos protocolos CARTIM e COTEC, dois Cenários (*A* e *B*) foram propostos para avaliação dos protocolos. Em ambos os cenários, uma autoestrada (*freeway* – Figura 23 (a)) de dois quilômetros, com duas faixas, foi utilizada. Os mapas dos cenários foram criados no SUMO, cuja taxa de chegada de veículos corresponde a 1 veículo por segundo. O Cenário *A* é similar ao cenário utilizado pelos autores do COTEC. Tal cenário possui velocidade máxima na via de 120 km/h. O Cenário *B* apresenta velocidade máxima permitida de 40 km/h, que corresponde à velocidade máxima em algumas vias urbanas. Na simulação, com a intenção de gerar congestionamento, foi forçado um acidente entre alguns veículos, começando a partir do tempo de simulação $t = 150$ s, obstruindo a via durante 5 minutos (300 s) em ambos os cenários. Os dados obtidos, nas simulações, foram para avaliar veículos em 800 metros da autoestrada, na região do acidente. Cada cenário foi simulado 33 vezes e avaliado por 10 minutos (600 s) durante o período do acidente. O tempo médio gasto, para simulação de cada cenário, foi de aproximadamente 17 minutos.

Os gráficos das Figuras 24 e 25 têm como objetivo mostrar o comportamento dos protocolos em estimar localmente (nos veículos) o tráfego veicular. Para isso, os conjuntos *fuzzy* de entrada do sistema também foram avaliados. Esses gráficos representam as velocidades e densidades médias, obtidas pelos veículos durante as simulações. Os níveis de congestionamento local inferidos pelos sistemas *fuzzy*, durante as simulações, também são ilustrados nesses gráficos. Essas informações agrupadas em um gráfico permitem visualizar o nível de congestionamento estimado pelo sistema, à medida que a velocidade e a densidade de veículos vizinhos variam.

Nos gráficos do Cenário *A* (Figura 24), se observa que o CARTIM obtém resultados similares ao COTEC, pois ambos os sistemas *fuzzy* estimaram níveis de congestionamento na mesma intensidade durante as simulações. O comportamento similar dos protocolos neste Cenário *A* valida a proposta *fuzzy* do CARTIM. De acordo com a Figura 24, ainda se verifica que, à medida que os conjuntos *fuzzy* de entrada do sistema variam, os níveis de congestionamento estimados também variam (de acordo com as regras *fuzzy* definidas).

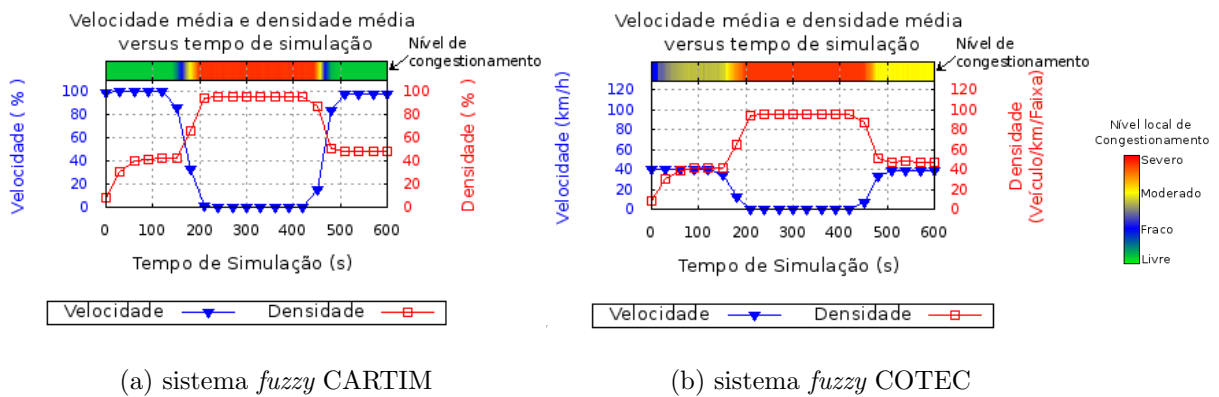
A Figura 25 (a) mostra os níveis de congestionamento estimados pelo CARTIM no Cenário *B*, bem como as velocidades e densidades médias dos veículos durante as simulações. Contudo, o COTEC, neste Cenário *B*, apresentou uma diferença considerável nos níveis de congestionamento, Figura 25 (b), comparado com o CARTIM. Isso é atribuído ao *fuzzy* estático do protocolo, que fica desconfigurado em cenários distintos do proposto para o protocolo. Assim, cenários com mais faixas (nas vias) afetariam ainda mais a estimativa de tráfego do COTEC.

Figura 24 – Nível de congestionamento no Cenário A, aferido pela velocidade média e densidade média de veículos na *Freeway*, versus tempo de simulação



Fonte: Dados da pesquisa.

Figura 25 – Nível de congestionamento no Cenário B, aferido pela velocidade média e densidade média de veículos na *Freeway*, versus tempo de simulação



Fonte: Dados da pesquisa.

As simulações mostraram que a proposta *fuzzy* dinâmica do CARTIM se adapta melhor ao cenário (contexto), obtendo informações mais precisas da situação do tráfego. É essa abordagem que ajuda no funcionamento do protocolo em cenários heterogêneos, comuns em redes veiculares. Além disso, ao estimar o tráfego veicular inferindo níveis de congestionamento mais elevados, o COTEC inicia antecipadamente o processo colaborativo (por se atingir o C_{th}) para validar as condições detectadas na via. Isso, no entanto, provoca um *overhead* desnecessário na rede e pode ser observado na Tabela 3, que mostra o total de mensagens CTE transmitidas e retransmitidas no Cenário B. Nessa tabela se verifica que o COTEC, comparado com CARTIM, teve um aumento de 534%.

Tabela 3 – Total de mensagens CTE no Cenário B

Protocolo	Quantidade (#)
CARTIM	134
COTEC	716

Fonte: Dados da pesquisa.

5.2 Resultados do CARTIM no Manhattan *Grid*

Para avaliar o CARTIM, simulações foram realizadas em um cenário Manhattan *Grid* (10 x 10), sob uma região de 20,25 km². Esse tipo de mapa é comumente usado em redes veiculares, tendo como objetivo avaliar protocolos em ambientes urbanos. A interseção das ruas no mapa representa um cruzamento e cada rua possui extensão de 500 m, faixa única, mão, contramão e velocidade máxima permitida variando entre 50 e 60 km/h em ruas perpendiculares.

O Quadro 6 apresenta os cinco cenários utilizados para avaliar o CARTIM nessa rede rodoviária. O número de veículos caracteriza a quantidade de nós no instante inicial da simulação. Com destinos aleatórios, cada veículo realizou uma viagem com distância mínima de 4 km. Os obstáculos representam estruturas (e.g. prédios etc) nos cenários, que podem dificultar a comunicação via rádio entre os nós. A mudança de rota corresponde à proposta de minimização de congestionamento do CARTIM. A arquitetura da rede define como a rede veicular se organiza. As RSUs, presentes no Cenário 5, foram instaladas em todos os cruzamentos.

O tempo de simulação de cada cenário foi de 10 minutos, sendo que cada cenário foi simulado 33 vezes e o tempo total para simular um cenário variou em função da quantidade de veículos. Os cenários com 1000 veículos gastaram aproximadamente 50 minutos e os cenários com 2000 veículos cerca de 1 hora e 40 minutos para serem simulados. Os gráficos desta seção que representam valores médios possuem intervalos de confiança de 95%. Contudo, alguns intervalos de confiança estão bastante próximos do valor médio, o que dificulta a visualização nas figuras.

Quadro 6 – Cenários de simulação

Cenário	Nº de Veículos	Obstáculos	Mudança de rota	Arquitetura(s)
1	1000	não	não	v2v
2	1000	sim	não	v2v
3	2000	sim	não	v2v
4	2000	sim	sim	v2v
5	2000	sim	sim	v2x ⁽¹⁾

1 - Arquiteturas V2V e V2I.

Fonte: Dados da pesquisa.

5.2.1 Mensagens de Congestionamento de Tráfego

A técnica (CTDiss) de retransmissão de mensagens de congestionamento de tráfego usada pelo CARTIM, discutida na Seção 4.3 e ilustrada na Figura 20, foi avaliada nos cenários simulados. A Tabela 4 discrimina a quantidade de mensagens de congestionamento de tráfego transmitidas, retransmitidas e recebidas durante as simulações. Os Cenários 1 e 2, com menor densidade de veículos, tiveram maiores perdas de mensagens.

Tabela 4 – Mensagens de congestionamento por cenário

Cenário	1	2	3	4	5
Transmitidas	811	799	3413	3386	3378
Retransmitidas	54	53	1634	1604	1908
Recebidas	770	741	4599	4502	4766
Perdidas (%)	10.98	13.03	8.87	9.78	9.84

Fonte: Dados da pesquisa.

Os Cenários 1 e 2 foram criados apenas para comparar e avaliar o processo de retransmissão de mensagens do CARTIM com presença de obstáculos. A Tabela 4 mostra que o Cenário 2 teve um aumento aproximado de 2.05% na taxa de mensagens perdidas, comparado com o Cenário 1. Isso ocorreu em razão dos obstáculos. No entanto, devido à especificidade das redes veiculares, as taxas alcançadas não comprometeram o funcionamento do CARTIM. Além disso, à medida que os congestionamentos aumentam, a densidade de veículos na região também aumenta, o que diminui a mobilidade dos nós na rede, reduzindo, assim, a perda de mensagens. Essa afirmação é verificada na Tabela 4, comparando as taxas de mensagens perdidas nos Cenários 1 e 2 com os Cenários 3, 4, 5.

A heurística de minimização de congestionamento do CARTIM, adotada nos Cenários 4 e 5, provocou uma ligeira elevação na taxa de perda de mensagens, quando comparados com o Cenário 3. Isso ocorreu porque tal heurística, ao permitir mudanças de rotas, gera mais alterações na topologia da rede. A Tabela 4 ainda mostra que, no Cenário 5, houve número maior de mensagens retransmitidas em relação aos Cenários 3 e 4. Isso ocorreu pois, devido à infraestrutura presente no Cenário 5, as RSUs passaram a retransmitir mensagens CTA ao escutá-las.

5.2.2 Sobrecarga de Comunicação

É necessário avaliar o impacto da adoção do CARTIM em relação às mensagens de congestionamento de tráfego. A Tabela 4 é utilizada para contabilizar o total de mensagens enviadas e retransmitidas pelos veículos. Essas mensagens possuem uma carga útil de aproximadamente 128 bytes (1024 bits) no protocolo. Nos Cenários 3, 4 e 5, um total de 5047, 4990 e 5286 mensagens foram disseminadas, respectivamente, durante as simulações, o que corresponde às taxas médias de 8.4, 8.3 e 8.8 mensagens por segundo. Porém, essas taxas de mensagens disseminadas durante as simulações não foram suficientes e significativas para sobrecarregar a rede, uma vez que a taxa de transferência era de 6 Mbit/s (6×10^6 bits/segundo). Nessas circunstâncias, a rede suportaria a transmissão de quase 5900 mensagens de congestionamento por segundo.

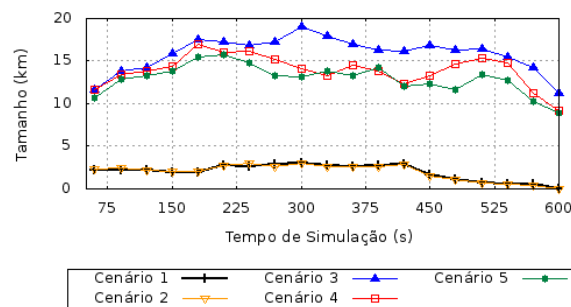
5.2.3 Tamanho do Congestionamento

A Figura 26 mostra o tamanho total dos congestionamentos nos cenários por unidade de tempo. Essa métrica é avaliada por órgãos gestores do tráfego para quantificar

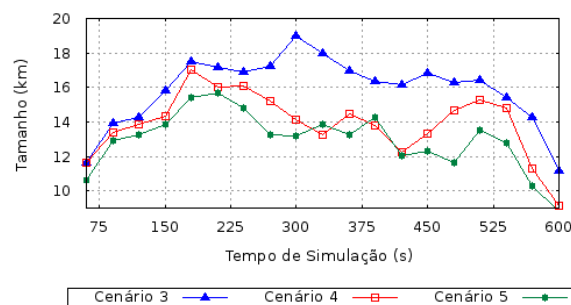
a situação de momento no trânsito. Neste trabalho, tal métrica é utilizada para medir e avaliar a influência do CARTIM nos cenários simulados.

A Figura 26 (b) mostra que o CARTIM ajudou a reduzir o tamanho do congestionamento geral nos Cenários 4 e 5 durante as simulações. O Cenário 3, que só utiliza a abordagem de identificação de congestionamento, obteve engarrafamentos maiores. Tal fato ocorre pois veículos nos Cenários 4 e 5, assim que escutam mensagens CTA na rede, obtêm aprovação para mudar de rota, evitando passar em vias sob condições severas de tráfego já validadas. O Cenário 5, que utiliza as arquiteturas V2V e V2I, de acordo com o gráfico ilustrado na Figura 26 (b), ainda obteve congestionamento geral durante boa parte da simulação menor que os verificados no Cenário 4, que usa apenas a arquitetura V2V. Esse fato é atribuído à infraestrutura (RSUs), que aumenta o raio de comunicação ao retransmitir mensagens CTA a veículos mais distantes.

Figura 26 – Congestionamento geral por cenário versus tempo de simulação



(a) Todos os cenários

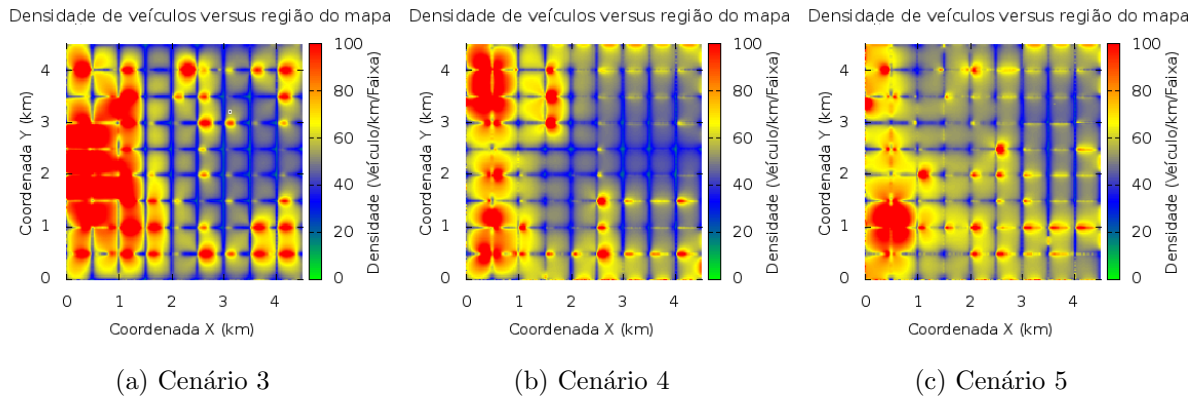


(b) Cenários com 2000 veículos

Fonte: Dados da pesquisa.

A Figura 27 ilustra a densidade de veículos por região (inferida pelo sistema *fuzzy*) durante as simulações, nos cenários com 2000 veículos. O objetivo é observar o fluxo de veículos no mapa. A Figura 27 (a) mostra uma grande região (em vermelho) com alta densidade de veículos no Cenário 3, em que, em comparação com os Cenários 4 e 5 (Figuras 27 (b) e (c)), se verifica um espalhamento menor dos veículos sobre o mapa. Tal observação é feita em razão da minimização de congestionamento do CARTIM, que permite mudanças de rotas perante condições severas e, assim, tende a melhorar o fluxo de veículos nas vias.

Figura 27 – Mapa de calor com a densidade de veículos por região do mapa, estimado pelos veículos durante as simulações usando o CARTIM

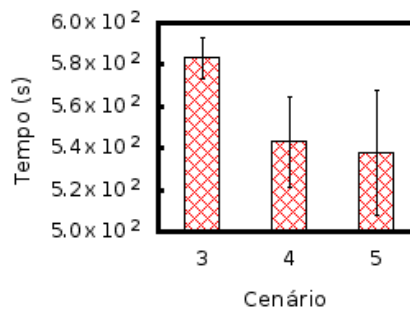


Fonte: Dados da pesquisa.

5.2.4 Tempo de Viagem, Distância Percorrida e Emissão de CO_2

O tempo de viagem é a métrica de interesse dos usuários das redes veiculares. A Figura 28 ilustra o tempo médio de viagem dos veículos por cenário. Em outras palavras, a figura mostra quanto tempo na média um veículo gastou no seu trajeto completo. Os dados apresentados na Figura 28 apontam que o CARTIM reduziu o tempo médio de viagem nos Cenários 4 e 5 (em relação ao 3), pois nesses cenários a minimização de congestionamento foi utilizada pelos veículos. No Cenário 5, como a infraestrutura estava presente, também é verificado no gráfico da Figura 28 que os veículos realizaram os trajetos no menor tempo médio.

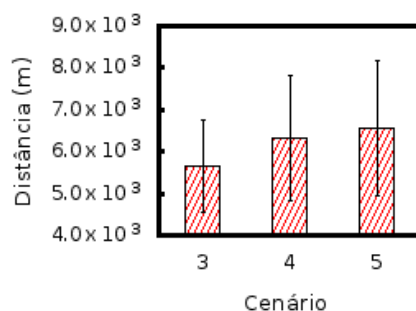
Figura 28 – Tempo médio de viagem versus cenário



Fonte: Dados da pesquisa.

A distância percorrida e a emissão de CO_2 são métricas utilizadas neste trabalho para avaliar a adoção do CARTIM. A Figura 29 mostra que a distância média percorrida pelos veículos nos Cenários 4 e 5 aumentou (em razão das mudanças de rotas). Mas como o principal objetivo é reduzir o tempo de viagem, o aumento do trajeto não compromete os benefícios apresentados do CARTIM.

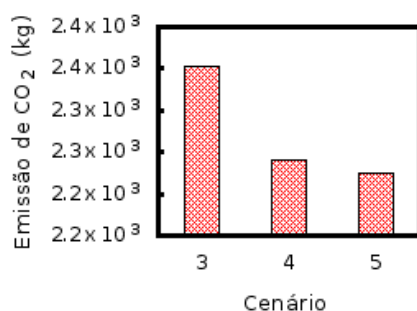
A emissão de CO_2 mede a quantidade de gás carbônico originado dos veículos que foi depositada no ar durante o trajeto dos mesmos. Essa informação é relevante nos

Figura 29 – Distância média de viagem versus cenário

Fonte: Dados da pesquisa.

dias atuais devido à grande preocupação com a qualidade do ar e a sustentabilidade. Na cidade de Belo Horizonte, MG – Brasil, no ano de 2010, foram emitidos 3,75 milhões de toneladas de CO₂ no ar, sendo que 71% desse montante foram provenientes de veículos. Esse número é 18% maior se comparado com o ano de 2007. Portanto, é de interesse da sociedade que se tenha soluções para diminuir a emissão de gases poluentes na atmosfera (SILVA et al., 2013).

Neste trabalho, a emissão de CO₂ foi usada como métrica e calculada com base no *framework* VEINS. Como apresentado na Figura 30, os Cenários 4 e 5, que adotam a abordagem para minimização de congestionamento do CARTIM, reduziram a emissão de CO₂, comparados com Cenário 3. Isso ocorreu devido à redução do tempo médio de viagem dos veículos nesses cenários (Figura 28). Essa diminuição da emissão de CO₂ ocorre mesmo com os veículos tendo que percorrer uma distância maior, como mostra a Figura 29. O aumento da distância percorrida é devido às rotas alternativas (geradas pelo algoritmo de Dijkstra) serem, em geral, mais longas que as rotas iniciais.

Figura 30 – Emissão de CO₂ versus cenário

Fonte: Dados da pesquisa.

5.3 Resultados do CARTIM no Mapa Geográfico

Para avaliar a adoção do CARTIM em um cenário real, simulações com protocolo foram conduzidas em um mapa geográfico, usando, para isso, o modelo real de tráfego da cidade de Belo Horizonte. Esse modelo real foi obtido junto à BHTRANS (Empresa

de Transportes e Trânsito de Belo Horizonte) e adaptado para as simulações, em que algumas das principais vias da cidade (Tabela 5) que fazem acesso a região central foram mapeadas. A Figura 31 (a) ilustra o mapa geográfico da região central de Belo Horizonte. Essa região corresponde a um perímetro urbano de aproximadamente 12 km. A Figura 31 (b) mostra a mesma região, criada para a simulação de VANETs, no simulador de tráfego e mobilidade SUMO.

**Tabela 5 – Contagem volumétrica de veículos (CVV)
em direção a região central de Belo Horizonte, MG**

Local	CVV (veículo/hora) entre 8:00 e 9:00
Av. Amazonas com Av. do Contorno	3919
Av. Brasil com Av. Francisco Sales	1437
Elevado Castelo Branco com Av. Dom Pedro II	2721
Av. Nossa Senhora do Carmo com Av. do Contorno	5894
Av. Prudente de Moraes com Av. do Contorno	1321
Av. Raja Gabaglia com Av. do Contorno	1674
Av. Tereza Cristina com Av. do Contorno	2319

Fonte: BHTRANS (2011).

No contexto da Tabela 5, mais de 19.000 veículos partem dos bairros rumo a região central da cidade entre 8:00 e 9:00 da manhã de um dia útil. Para avaliação do protocolo, sobre esse mapa geográfico, dois cenários foram propostos: (i) CARTIM e (ii) COMUM. No Cenário COMUM, nenhuma das funcionalidades do CARTIM foram utilizadas, e esse cenário corresponde a um ambiente sem VANETs (consequentemente sem monitoramento de tráfego). Os Quadros 3 e 5 (apresentados anteriormente) mostram os parâmetros de simulação utilizados no Cenário CARTIM.

Os Cenários CARTIM (i) e COMUM (ii) foram simulados uma única vez, e, como, o volume de veículos nas vias rumo ao centro da cidade (Tabela 5) é muito grande, o número de faixas (dessas vias) foram reduzidos em apenas uma, representando os respectivos volumes de veículos, como mostra a Tabela 6. Com isso, foi possível computacionalmente simular esse cenário real. A simulação termina quando todos os veículos chegam ao seu destino final, que corresponde a cerca de 21 horas e 35 minutos para o Cenário CARTIM. O Cenário COMUM, que corresponde a um cenário real e sem VANET, foi simulado mais rápido em aproximadamente 50 minutos (devido ao menor uso de memória por não ter nenhum protocolo implementado).

Tabela 6 – Número médio de veículos por faixa por via

Local	Número de Veículos por faixa
Av. Amazonas com Av. do Contorno	1960
Av. Brasil com Av. Francisco Sales	479
Elevado Castelo Branco com Av. Dom Pedro II	680
Av. Nossa Senhora do Carmo com Av. do Contorno	1474
Av. Prudente de Moraes com Av. do Contorno	660
Av. Raja Gabaglia com Av. do Contorno	837
Av. Tereza Cristina com Av. do Contorno	580

Fonte: Dados da pesquisa.

Figura 31 – Mapas da região central de Belo Horizonte, MG



(a) Mapa geográfico

(b) Mapa rodoviário - SUMO

Fonte: Criada pelo autor.

A Tabela 7 mostra as médias, os desvios padrão e os intervalos de confiança de 95% sobre os tempos de viagens e distâncias percorridas pelos veículos em ambos os cenários. A emissão total de CO₂ também foi medida em cada cenário simulado. Os resultados apresentados nessa tabela também mostram a capacidade do protocolo CARTIM em minimizar congestionamentos em cenários reais.

O tempo médio que a população gasta em deslocamentos urbanos casa-trabalho na região metropolitana de Belo Horizonte é de 34,4 minutos (PEREIRA; SCHWANEN, 2013). Nas simulações, o tempo médio gasto pelos veículos, no percurso até a região central de BH, no Cenário CARTIM (i) foi de 1380,10 segundos (aproximadamente 23 minutos), 16% mais rápido comparado com os veículos no Cenário COMUM (ii), cujo tempo do percurso foi de aproximadamente 27 minutos (1635,32 segundos). No entanto, como a proposta de minimização de congestionamento do CARTIM busca um estado mais balanceado da rede, foi verificado, por essa razão, um aumento médio do percurso dos veículos até os respectivos destinos, conforme apresentado na Tabela 7. Nessa tabela, o Cenário CARTIM (i) mostra um aumento médio de aproximadamente 4% no percurso dos veículos em relação ao Cenário COMUM (ii). Contudo, como o objetivo específico é reduzir o tempo de viagem, esse aumento médio do trajeto realizado é esperado, uma vez que a proposta de minimização de congestionamento do protocolo CARTIM busca um estado mais balanceado da rede rodoviária para reduzir o tempo de viagem.

Em cenários mais realistas, com dados reais do tráfego veicular, o CARTIM também conseguiu reduzir a emissão de CO₂ pelos veículos (Tabela 7). Com aumento da frota de veículos e, consequentemente, aumento da poluição gerada, propostas que reduzam a emissão de gás carbônico pelos veículos serão necessárias. O CARTIM alcança esse objetivo específico indiretamente, reduzindo o tempo médio de viagem dos veículos.

Tabela 7 – Desempenho do protocolo CARTIM no cenário real

Cenário	CARTIM	COMUM
Tempo de viagem por veículo		
média (<i>s</i>)	1380,10	1635,32
desvio padrão (<i>s</i>)	612,10	889,47
intervalo de confiança de 95% (<i>s</i>)	[1362,25 ; 1397,95]	[1609,38 ; 1661,26]
erro ou tolerância (<i>s</i>)	17,85	25,94
Distância percorrida por veículo		
média (<i>m</i>)	2881,46	2770,45
desvio padrão (<i>m</i>)	219,71	210,02
intervalo de confiança de 95% (<i>m</i>)	[2873,77 ; 2889,14]	[2764,32 ; 2776,57]
erro ou tolerância (<i>m</i>)	7,69	6,16
Emissão total de CO₂ (<i>kg</i>)	10927,18	11749,66

Fonte: Dados da pesquisa.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta as principais conclusões e contribuições alcançadas durante a pesquisa. Ao final desse capítulo são feitas algumas sugestões de trabalhos futuros.

6.1 Conclusões

As redes veiculares possuem uma série de desafios para sua adoção em larga escala. Dentre os principais desafios estão particularidades como a alta mobilidade dos nós, o dinamismo dos cenários e a escalabilidade em termos do número de nós. A perda de conectividade durante a transmissão dos dados e o tempo reduzido em que dois nós permanecem em contato são outros desafios. As aplicações de redes veiculares devem continuar evoluindo e buscando soluções para estes desafios, visto que um estado mais balanceado das vias são de interesse coletivo, pois auxiliam na redução dos problemas no trânsito.

Um dos maiores desafios em VANETs é o desenvolvimento de protocolos eficientes para identificação e minimização de congestionamento de tráfego, bem como o desenvolvimento de algoritmos eficientes para comunicação e disseminação de informações sobre o tráfego na rede, utilizando múltiplos saltos.

A principal contribuição deste trabalho é o CARTIM, um protocolo de identificação e minimização de congestionamento de tráfego. O CARTIM foi planejado considerando as características e as realidades das redes veiculares, buscando ser adaptável ao contexto e, assim, viável em cenários heterogêneos (autoestrada e vias urbanas). Essas características mencionadas distinguem o CARTIM das propostas encontradas na literatura, que geralmente propõem soluções para problemas de VANETs para cenários específicos. Além disso, o protocolo aqui proposto não depende da infraestrutura para monitoramento do tráfego veicular. As RSUs, quando presentes, são transparentes para o protocolo, sua função é apenas aumentar o raio de disseminação de mensagens CTA.

As seguintes métricas foram definidas para comparar e avaliar o protocolo apresentado neste trabalho: nível de congestionamento aferido pela velocidade e pela densidade média, quantidade de mensagens CTE disseminadas, sobrecarga de comunicação, tamanho do congestionamento, tempo médio de viagem, distância média percorrida e emissão de CO₂.

Essas métricas foram avaliadas em três cenários distintos: (i) *Freeway*, (ii) *Manhattan Grid* (10 x 10) e (iii) *Mapa Geográfico*. No cenário *Freeway* (i), os resultados

mostraram a eficácia do sistema *fuzzy* dinâmico em detectar congestionamentos veiculares e comprovaram a capacidade do CARTIM em se adaptar ao contexto. Os resultados obtidos no cenário Manhattan *Grid* (ii) mostraram a capacidade do protocolo de minimizar congestionamentos, melhorando o fluxo de veículos e consequentemente conseguindo um estado mais balanceado sobre a rede rodoviária. Dessa forma, o CARTIM atingiu a principal finalidade deste trabalho: reduzir o tempo médio de viagem dos veículos no tráfego, procurando aumentar a satisfação dos usuários da rede veicular. O protocolo ainda reduziu a emissão de CO₂ na atmosfera pelos veículos durante as simulações, mesmo as distâncias médias percorridas terem aumentado. Por fim, os resultados obtidos no mapa da região central de Belo Horizonte (iii), com dados reais do tráfego veicular na região, também mostraram como o CARTIM pode auxiliar na minimização de congestionamentos, e consequentemente, na redução dos tempos médios de viagem em cenários urbanos reais.

A técnica CTDis, proposta para CARTIM, para transmissão de mensagens *unicast* na rede, em situações de tráfego congestionado, também foi avaliada nesse trabalho, e obteve resultados satisfatórios para o dinamismo desse tipo de MANET. Essa abordagem permitiu a transmissão e disseminação de mensagens de congestionamento veicular na rede. A grande dificuldade de transmissão e retransmissão de mensagens *unicast* em VANETs é devido a alta mobilidade dos nós e as constantes mudanças de topologia da rede. A curta janela de transmissão também pode acarretar elevadas taxas de perda de mensagens. Por isso, as taxas alcançadas pela CTDis foram aceitáveis e suficientes para funcionamento do protocolo proposto.

6.2 Trabalhos Futuros

Como trabalhos futuros, podem ser citados:

- Comparação do CARTIM com outras propostas encontradas na literatura para controle de congestionamento veicular, que inclusive, não necessariamente se apoie em inteligência computacional como lógica *fuzzy*.
- Criação de outras propostas de minimização de congestionamento veicular, que devem ser avaliadas e comparadas com a proposta inicial apresentada neste trabalho.
- Novos algoritmos de roteamento baseados por exemplo nos níveis de congestionamento estimados cooperativamente pelos veículos (nas vias congestionadas). Essa nova abordagem ainda deve ser comparada com a proposta inicial apresentada nesse trabalho, que se baseia no algoritmo de Dijkstra, conforme discutido no Capítulo 2.

- Protocolos de disseminação de mensagens *geocast* também podem ser implementados nas RSUs, afim de evitar uma possível sobrecarga na rede.
- Novas avaliações no aumento dos raios de comunicação V2V e V2I devem ser realizadas, pois o aumento desses raios pode acarretar em um aumento exponencial na sobrecarga do canal de comunicação.
- Mais avaliações sobre a técnica CTDis em cenários específicos, como por exemplo cenários sob baixa densidade de veículos etc, também se fazem necessárias para verificar a eficácia e eficiência dessa abordagem, inclusive em outras aplicações de VANET para disseminação de informações na rede.
- A utilização de microcontroladores e sensores como acelerômetro, GPS (*Global Positioning System*) etc podem viabilizar testes do CARTIM em cenários reais, com baixo custo financeiro por veículo.
- Outras hipóteses do mundo real também devem ser levantadas, avaliadas e testadas no CARTIM para verificar por exemplo: como a proposta de minimização de congestionamento do protocolo e a técnica CTDis se comportam em ambientes reais de teste. Possíveis ajustes no protocolo serão necessários em função da tecnologia utilizada.

REFERÊNCIAS

- ALVES, R. d. S.; CAMPBELL, I. d. V.; COUTO, R. d. S.; CAMPISTA, M. E. M.; MORAES, I. M.; RUBINSTEIN, M. G.; COSTA, L. H. M. K.; DUARTE, O. C. M. B.; ABDALLA, M. **Redes Veiculares: Princípios, Aplicações e Desafios**. *Minicursos do XXVII Simpósio Brasileiro de Redes de Computadores*, Chapter V, p. 199–254, maio 2009.
- AMARANTE, M. d. B. d.; BAZZAN, A. L. C. **Um Processo para Avaliação de Balanceamento de Carga em Redes de Transporte Veicular Utilizando Adaptação Heurística**. In: *Anais do VIII Simpósio Brasileiro de Sistemas de Informação*. São Paulo, Brasil: SBC, 2012. p. 73–78.
- ARBABI, H.; WEIGLE, M. **Monitoring Free Flow Traffic Using Vehicular Networks**. In: *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*. Las Vegas, USA: IEEE, 2011. p. 272–2760.
- BAI, F.; HELMY, A. **A Survey of Mobility Models in Wireless Adhoc Networks**. *Wireless Adhoc Networks*, Chapter I, p. 1–30, 2006.
- BAUZA, R.; GOZALVEZ, J.; SANCHEZ-SORIANO, J. **Road Traffic Congestion Detection Through Cooperative Vehicle-to-Vehicle Communications**. *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, p. 606–612, october 2010. ISSN 0742-1303.
- BHTRANS. **Pesquisa de Contagem Classificada de Veículos**. 2011. Disponível em <<http://www.bhtrans.pbh.gov.br/>>. Acesso em: Janeiro 2014.
- BILGIN, B. E.; GUNGOR, V. C. **Performance Comparison of IEEE 802.11p and IEEE 802.11b for Vehicle-to-Vehicle Communications in Highway, Rural, and Urban Areas**. *International Journal of Vehicular Technology*, 2013.
- BINGLEI, X.; ZHENG, H.; HONGWEI, M. **Fuzzy-logic-based Traffic Incident Detection Algorithm for Freeway**. *Machine Learning and Cybernetics, 2008 International Conference on*, v. 3, p. 1254–1259, 2008.
- BOARD, N. R. C. U. T. R. **HCM2000**. Washington, D.C, USA: Transportation Research Board, 2000. (HCM 2000: Highway Capacity Manual). ISBN 9780309067461.
- CAPPIELLO, A.; CHABINI, I.; NAM, E.; LUE, A.; ZEID, M. A. **A statistical model of vehicle emissions and fuel consumption**. *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, p. 801–809, 2002.
- CINGOLANI, P.; ALCALA-FDEZ, J. **jFuzzyLogic: A Robust and Flexible Fuzzy-Logic Inference System Language Implementation**. *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, p. 1–8, June 2012. ISSN 1098-7584.

CORDEIRO, C. de M.; AGRAWAL, D. **Ad Hoc and Sensor Networks: Theory and Applications**. 2nd. ed. Singapore: World Scientific, 2011. ISBN 9789814338882.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms**. 3rd. ed. Massachusetts, USA: The MIT Press, 2009. ISBN 0262033844, 9780262033848.

CRISTEA, V.; GRADINESCU, V.; GORGORIN, C.; DIACONESCU, R.; IFTODE, L. **Automotive Informatics and Communicative Systems: Principles in Vehicular Networks and Data Exchange**. Hershey, USA: IGI Global, 2009. Chapter XIV (Simulation of Vanet Applications) 258–276 p. ISBN 9781605663388.

DENATRAN. **Frota de veículos**. 2014. Disponível em <<http://www.denatran.gov.br/frota.htm>>. Acesso em: Fevereiro 2014,.

DEO, N. **Graph Theory with Applications to Engineering with Computer Science**. Delhi, India: Prentice-Hall Of India Pvt. Limited, 1974. (Prentice-Hall series in automatic computation). ISBN 9788120301450.

DLR. **Institute of Transportation Systems**. 2014. Disponível em <<http://www.dlr.de/ts/en/>>. Acesso em: Janeiro 2014.

DORNBUSH, S.; JOSHI, A. **StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's**. In: *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*. Dublin, Ireland: IEEE, 2007. p. 11–15. ISSN 1550-2252.

DRESSLER, F.; SOMMER, C.; ECKHOFF, D.; TONGUZ, O. **Toward Realistic Simulation of Intervehicle Communication**. *IEEE Vehicular Technology Magazine*, v. 6, n. 3, p. 43–51, set. 2011. ISSN 1556–6072.

ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. 2nd. ed. Chichester, UK: Wiley Publishing, 2007. ISBN 0470035617.

FAHMY, M.; RANASINGHE, D. N. **Discovering automobile congestion and volume using vanet's**. In: *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*. Phuket, Thailand: IEEE, 2008. p. 367–372.

FONT, J. L.; ÑIGO, P.; DOMÍNGUEZ, M.; SEVILLANO, J. L.; AMAYA, C. **Architecture, Design and Source Code Comparison of Ns-2 and Ns-3 Network Simulators**. In: *Proceedings of the 2010 Spring Simulation Multiconference*. San Diego, CA, USA: Society for Computer Simulation International, 2010. (SpringSim '10), p. 109:1–109:8. ISBN 978-1-4503-0069-8.

FUKUMOTO, J.; SIROKANE, N.; ISHIKAWA, Y.; WADA, T.; OHTSUKI, K.; OKADA, H. **Analytic method for real-time traffic problems by using Contents Oriented Communications in VANET**. *2007 7th International Conference on ITS Telecommunications*, IEEE, jun. 2007.

GAMESS, E.; MAHGOUB, I.; RATHOD, M. **Scalability evaluation of two network simulation tools for Vehicular Ad hoc Networks**. In: *Wireless Advanced (WiAd), 2012*. London, UK: IEEE, 2012. p. 58–63.

GRILLI, G. **Data Dissemination in Vehicular Networks**. 198 p. Tese (Doutorado) — University of Rome, 2010.

HAKLAY, M. M.; WEBER, P. **OpenStreetMap: User-Generated Street Maps**. *IEEE Pervasive Computing*, IEEE Educational Activities Department, Piscataway, USA, v. 7, n. 4, p. 12–18, october 2008. ISSN 1536-1268.

HARTENSTEIN, H.; LABERTEAUX, K.; EBRARY, I. **VANET: Vehicular Applications and Inter-Networking Technologies**. New Jersey, USA: John Wiley Sons, 2010. 472 p. ISBN 978-0-470-74056-9.

HUANG, J.; QIAN, F.; GERBER, A.; MAO, Z. M.; SEN, S.; SPATSCHECK, O. **A close examination of performance and power characteristics of 4G LTE networks**. In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. New York, NY, USA: ACM, 2012. (MobiSys '12), p. 225–238. ISBN 978-1-4503-1301-8.

IKEDA, M.; KULLA, E.; BAROLLI, L.; TAKIZAWA, M.; MIHO, R. **Performance Evaluation of Wireless Mobile Ad-Hoc Network via NS-3 Simulator**. In: *Network-Based Information Systems (NBIS), 2011 14th International Conference on*. Tirana, Albânia: IEEE, 2011. p. 135–141. ISSN 2157-0418.

International Electrotechnical Commission (IEC) 61131-7 standard. **Fuzzy control language specification**. 2000.

iTETRIS. **Integrates Wireless Communications and Road Traffic Simulation Platforms**. 2014. Disponível em <<http://www.ict-itetris.eu/>>. Acesso em: Janeiro 2014.

JANG, K.; HAN, M.; CHO, S.; RYU, H.-K.; LEE, J.; LEE, Y.; MOON, S. B. **3G and 3.5G wireless network performance measured from moving cars and high-speed trains**. In: *Proceedings of the 1st ACM workshop on Mobile internet through cellular networks*. New York, NY, USA: ACM, 2009. (MICNET '09), p. 19–24. ISBN 978-1-60558-753-0.

JOSM. **Java Open Street Map Editor – User Guide**. 2014. Disponível em <<http://wiki.openstreetmap.org/wiki/JOSM/Guide/>>. Acesso em: Janeiro 2014.

KARAGIANNIS, G.; ALTINTAS, O.; EKICI, E.; HEIJENK, G.; JARUPAN, B.; LIN, K.; WEIL, T. **Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions**. *Communications Surveys Tutorials, IEEE*, v. 13, n. 4, p. 584–616, 2011. ISSN 1553-877X.

KNORR, F.; SCHRECKENBERG, M. **Influence of inter-vehicle communication on peak hour traffic flow**. *Physica A Statistical Mechanics and its Applications*, v. 391, p. 2225–2231, 2012.

KRAJZEWICZ, D.; ERDMANN, J.; BEHRISCH, M.; BIEKER, L. **Recent Development and Applications of SUMO - Simulation of Urban MObility**. *International Journal On Advances in Systems and Measurements*, v. 5, n. 3&4, p. 128–138, 2012.

KUMAR, V.; LIN, L.; KRAJZEWICZ, D.; HRIZI, F.; MARTINEZ, O.; GOZALVEZ, J.; BAUZA, R. **iTETRIS: Adaptation of ITS Technologies for Large Scale Integrated Simulation**. In: *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*. Taipei, Taiwan: IEEE, 2010. ISSN 1550-2252.

LAKAS, A.; CHEQFAH, M. **Detection and Dissipation of Road Traffic Congestion Using Vehicular communication**. *Mediterranean Microwave Symposium (MMS)*, Tangiers, Morocco, november 2009.

LAZARO, O.; ROBERT, E.; LAN, L.; GOZALVEZ, J.; TURKSMA, S.; FILALI, F.; CARTOLANO, F.; URRUTIA, M. A.; KRAJZEWICZ, D. **iTETRIS: An Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions**. In: *21st WWRP 2008*. Stockholm, Sweden: IEEE, 2008. p. 465–471. ISSN 1663-4918.

LU, J.; CAO, L. **Congestion Evaluation from Traffic Flow Information Based on Fuzzy Logic**. *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, v. 1, p. 50–53, 2003.

MACHADO, R. J. F. **Adaptação de Mapas Geográficos e Planeamento de Rotas na Simulação de Tráfego de Veículos em Ambiente Cidadino**. 103 p. Dissertação (Mestrado) — Universidade de Coimbra, setembro 2009.

MALLAPUR, S.; PATIL, S. **Survey on Simulation Tools for Mobile Ad-Hoc Networks**. *International Journal of Computer Networks and Wireless Communications (IJCNCW)*, v. 2, n. 2, p. 11–15, april 2012. ISSN 2250-3501.

MARTINEZ, F. J.; TOH, C. K.; CANO, J.-C.; CALAFATE, C. T.; MANZONI, P. **A Survey and Comparative Study of Simulators for Vehicular Ad Hoc Networks (VANETs)**. *Wireless Communications and Mobile Computing*, John Wiley and Sons, v. 11, n. 7, p. 813–828, jul. 2011.

MENEZES, J. G. M. D. **Soluções de Disseminação de Dados para Redes ad hoc sem fio**. 123 p. Tese (Doutorado) — Universidade Federal de Minas Gerais – UFMG, dezembro 2013.

MILLER, J. **Vehicle-to-vehicle-to-infrastructure (V2V2I) intelligent transportation system architecture**. In: *Intelligent Vehicles Symposium*. Eindhoven, Netherlands: IEEE, 2008. p. 715–720. ISSN 1931-0587.

MUHLETHALER, P.; LAOUITI, A.; TOOR, Y. **Comparison of Flooding Techniques for Safety Applications in VANETs**. *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, p. 1–6, 2007.

NADEEM, T.; DASHTINEZHAD, S.; LIAO, C.; IFTODE, L. **TrafficView: Traffic Data Dissemination Using Car-to-Car Communication**. *SIGMOBILE Mob. Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 8, n. 3, p. 6–19, jul. 2004. ISSN 1559-1662.

PADRON, F. M. **Traffic Congestion Detection Using VANET**. 113 p. Dissertação (Mestrado) — Florida Atlantic University, april 2009.

PAPALEONDIU, L. G.; DIKAIKOS, M. D. **TrafficModeler: A Graphical Tool for Programming Microscopic Traffic Simulators through High-Level Abstractions**. In: *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*. Seoul, Korea: IEEE, 2009.

PEREIRA, R. H. M.; SCHWANEN, T. Discussion Papers, **Tempo de Deslocamento Casa - Trabalho no Brasil (1992-2009): Diferenças Entre Regiões Metropolitanas, Níveis de Renda e Sexo**. 2013. Disponível em <<http://EconPapers.repec.org/RePEc:ipe:ipetds:1813>>. Acesso em: Janeiro 2014.

PIóRKOWSKI, M.; RAYA, M.; LUGO, A. L.; PAPADIMITRATOS, P.; GROSSGLAUSSER, M.; HUBAUX, J.-P. **TraNS: Realistic Joint Traffic and Network Simulator for VANETs**. *ACM SIGMOBILE Mobile Computing and Communications Review*, v. 12, n. 1, p. 31, jan. 2008. ISSN 15591662.

PROTZMANN, R.; SCHUNEMANN, B.; RADUSCH, I. **The influences of communication models on the simulated effectiveness of V2X applications**. *Communications Magazine, IEEE*, v. 49, n. 11, p. 149–155, 2011. ISSN 0163-6804.

RADA-VILELA, J. **Fuzzylite: A Fuzzy Logic Control Library in C++**. *Proceedings of the Open Source Developers Conference*, Auckland, NZ, 2013.

RADA-VILELA, J. **A Fuzzy logic control library**. 2014. Disponível em <<https://code.google.com/p/fuzzylite/>>. Acesso em: Janeiro 2014.

RONDINONE, M. **iTETRIS: The Integrated Wireless and Traffic Simulation Platform for Real-Time Road Traffic Management Solutions**. *COMeSafety Newsletter*, v. 8, oct/nov. 2010.

RUHE, G. **Algorithmic Aspects of Flows in Networks**. Heidelberg, Germany: Kluwer Academic Publishers, 1991. (Mathematics and its applications). ISBN 9780792311515.

SEMOB – Secretaria Nacional do Transporte e da Mobilidade Urbana. **Plano de Mobilidade Urbana**. 2007. Disponível em <<http://www.cidades.gov.br/images/stories/ArquivosSEMOB/Biblioteca/LivroPlanoMobilidade.pdf>>. Acesso em: Janeiro 2014.

SHAN, H.; JIANSHEG, W.; LING, X. **Real-time Traffic Congestion Detection Based on Video Analysis**, *Journal of Information Computational Science*, p. 2907–2914, 2012.

SILVA, F. A.; SILVA, T. R.; RUIZ, L. B.; LOUREIRO, A. F. A. **Um Middleware para Provisionamento de Contextos para Redes Veiculares**. *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC*, Chapter XV, p. 615–628, maio 2013.

SKYCOMP. **Freeway Performance Ratings and Arterial Bottleneck Inventory**. 2008. Disponível em <http://www.skycomp.com/VDOT/resources/VDOT_08.pdf>. Acesso em: Janeiro 2014.

SKYCOMP. 2014. Disponível em <<http://www.skycomp.com/>>. Acesso em: Janeiro 2014.

SOMMER, C. **Car-to-X Communication in Heterogeneous Environments**. 209 p. Tese (Doutorado) — University of Erlangen, june 2011.

SOMMER, C.; GERMAN, R.; DRESSLER, F. **Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis**. *IEEE Transactions on Mobile Computing*, v. 10, n. 1, p. 3–15, jan. 2011. ISSN 1536-1233.

SOMMER, C.; KRUL, R.; GERMAN, R.; DRESSLER, F. **Emissions vs. Travel Time: Simulative Evaluation of the Environmental Impact of ITS**. *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, p. 1–5, May 2010. ISSN 1550-2252.

SOMMER, C.; TONGUZ, O.; DRESSLER, F. **Traffic information systems: efficient message dissemination via adaptive beaconing**. *Communications Magazine, IEEE*, v. 49, n. 5, p. 173–179, 2011. ISSN 0163-6804.

SUMO. **Simulation of Urban MObility**. 2014. Disponível em <<http://sumo-sim.org/>>. Acesso em: Janeiro 2014.

TANSCHKEIT, R. **Sistemas Fuzzy**. 2004. Disponível em <<http://www.inf.ufsc.br/~mauro/ine5377/leituras/ICA-Sistemas%20Fuzzy.pdf>>. Acesso em: Janeiro 2014.

TraNS. **Traffic and Network Simulation Environment**. 2014. Disponível em <<http://trans.epfl.ch/>>. Acesso em: Janeiro 2014.

VARGA, A.; HORNIG, R. **An Overview of the OMNeT++ Simulation Environment**. *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, p. 1–10, 2008.

VEINS. **VEhicles In Network Simulation**. 2014. Disponível em <<http://veins.car2x.org/>>. Acesso em: Janeiro 2014.

VSimRTI. **Smart Mobility Simulation**. 2014. Disponível em <http://www.dcaiti.tu-berlin.de/research/simulation/>. Acesso em: Janeiro 2014.

WEGENER, A.; PIÓRKOWSKI, M.; RAYA, M.; HELLBRÜCK, H.; FISCHER, S.; HUBAUX, J.-P. **TraCI: An Interface for Coupling Road Traffic and Network Simulators**. In: *Proceedings of the 11th Communications and Networking Simulation Symposium*. Ottawa, Canada: ACM, 2008. (CNS '08), p. 155–163. ISBN 1-56555-318-7.

YOUNES, M.; BOUKERCHE, A. **Efficient traffic congestion detection protocol for next generation VANETs**. *Communications (ICC), 2013 IEEE International Conference on*, p. 3764–3768, June 2013. ISSN 1550-3607.

ZADEH, L. A. **Fuzzy Sets**. *Information Control*, v. 8, p. 338–353, 1965.

APÊNDICE A – CÓDIGO SISTEMA *FUZZY* CARTIM

O código A.1 apresenta a configuração do sistema fuzzy proposto para CARTIM. Os conjuntos *fuzzy*, as variáveis linguísticas, as funções de pertinência e as regras *fuzzy* são descritos pela *Fuzzy Control Language*. O código em C++ (A.2) mostra a implementação do sistema *fuzzy* do CARTIM.

Código A.1 – Código FCL do sistema *fuzzy*

```

1  /*
2      FCL Sistema Fuzzy CARTIM
3  */
4
5  FUNCTION_BLOCK CARTIM
6
7  VAR_INPUT
8      density: REAL;
9      speed: REAL;
10  END_VAR
11
12  VAR_OUTPUT
13      congestion: REAL;
14  END_VAR
15
16  FUZZIFY density
17  RANGE := (0.000 .. 1.000);
18  TERM low := Trapezoid (0.000, 0.000, 0.150, 0.300);
19  TERM medium := Triangle (0.200, 0.350, 0.500);
20  TERM high := Triangle (0.400, 0.550, 0.700);
21  TERM veryhigh := Trapezoid (0.500, 0.750, 1.000, 1.000);
22  END_FUZZIFY
23
24  FUZZIFY speed
25  RANGE := (0.000 .. 1.000);
26  TERM veryslow := Trapezoid (0.000, 0.000, 0.200, 0.400);
27  TERM slow := Triangle (0.100, 0.400, 0.700);
28  TERM medium := Triangle (0.300, 0.600, 0.900);
29  TERM fast := Trapezoid (0.600, 0.800, 8.100, 8.200);
30  END_FUZZIFY
31
32  DEFUZZIFY congestion
33  RANGE := (0.000 .. 1.000);
34  TERM free := Trapezoid (0.000, 0.000, 0.080, 0.170);

```

```

35 TERM slight := Triangle (0.170, 0.330, 0.500);
36 TERM moderate := Triangle (0.500, 0.670, 0.830);
37 TERM severe := Trapezoid (0.830, 0.910, 1.000, 1.000);
38
39 METHOD : COG;
40 ACCU : MAX;
41 DEFAULT := inf | NC;
42 END_DEFUZZIFY
43
44 RULEBLOCK
45 AND : MIN;
46 OR : MAX;
47 ACT : MIN;
48
49 RULE 1 : if density is low and speed is veryslow then congestion is slight
50 RULE 2 : if density is low and speed is slow then congestion is slight
51 RULE 3 : if density is low and speed is medium then congestion is free
52 RULE 4 : if density is low and speed is fast then congestion is free
53 RULE 5 : if density is medium and speed is veryslow then congestion is
      moderate
54 RULE 6 : if density is medium and speed is slow then congestion is slight
55 RULE 7 : if density is medium and speed is medium then congestion is slight
56 RULE 8 : if density is medium and speed is fast then congestion is free
57 RULE 9 : if density is high and speed is veryslow then congestion is severe
58 RULE 10 : if density is high and speed is slow then congestion is moderate
59 RULE 11 : if density is high and speed is medium then congestion is
      moderate
60 RULE 12 : if density is high and speed is fast then congestion is slight
61 RULE 13 : if density is veryhigh and speed is veryslow then congestion is
      severe
62 RULE 14 : if density is veryhigh and speed is slow then congestion is
      severe
63 RULE 15 : if density is veryhigh and speed is medium then congestion is
      moderate
64 RULE 16 : if density is veryhigh and speed is fast then congestion is
      moderate
65 END_RULEBLOCK
66
67 END_FUNCTION_BLOCK

```

Código A.2 – Código C++ do sistema *fuzzy*

```

1  /*
2     C++ Sistema Fuzzy CARTIM
3  */
4
5  #include <fl/Headers.h>

```

```

6
7 void fuzzyEngine() {
8
9   fl::Engine* engine = new fl::Engine;
10  engine->setName("CARTIM");
11  engine->addHedge(new fl::Any);
12  engine->addHedge(new fl::Extremely);
13  engine->addHedge(new fl::Not);
14  engine->addHedge(new fl::Seldom);
15  engine->addHedge(new fl::Somewhat);
16  engine->addHedge(new fl::Very);
17
18  fl::InputVariable* inputVariable1 = new fl::InputVariable;
19  inputVariable1->setName("density");
20  inputVariable1->setRange(0.000, 1.000);
21
22  inputVariable1->addTerm(new fl::Trapezoid("low", 0.000,0.000,0.150,0.300));
23  inputVariable1->addTerm(new fl::Triangle("medium", 0.200,0.350,0.500));
24  inputVariable1->addTerm(new fl::Triangle("high", 0.400,0.550,0.700));
25  inputVariable1->addTerm(new fl::Trapezoid("veryhigh",
        0.500,0.750,1.000,1.000));
26  engine->addInputVariable(inputVariable1);
27
28  fl::InputVariable* inputVariable2 = new fl::InputVariable;
29  inputVariable2->setName("speed");
30  inputVariable2->setRange(0.000, 1.000);
31
32  inputVariable2->addTerm(new fl::Trapezoid("veryslow",
        0.000,0.000,0.200,0.400));
33  inputVariable2->addTerm(new fl::Triangle("slow", 0.100,0.400,0.700));
34  inputVariable2->addTerm(new fl::Triangle("medium", 0.300,0.600,0.900));
35  inputVariable2->addTerm(new fl::Trapezoid("fast", 0.600,0.800,8.100,8.200))
        ;
36  engine->addInputVariable(inputVariable2);
37
38  fl::OutputVariable* outputVariable1 = new fl::OutputVariable;
39  outputVariable1->setName("congestion");
40  outputVariable1->setRange(0.000, 1.000);
41  outputVariable1->setDefaultValue(std::numeric_limits<scalar>::infinity());
42  outputVariable1->setLockDefuzzifiedValue(true);
43  outputVariable1->setDefuzzifier(new fl::Centroid(500));
44  outputVariable1->output()->setAccumulation(new fl::Maximum);
45
46  outputVariable1->addTerm(new fl::Trapezoid("free", 0.000,0.000,0.080,0.170)
        );
47  outputVariable1->addTerm(new fl::Triangle("slight", 0.170,0.330,0.500));
48  outputVariable1->addTerm(new fl::Triangle("moderate", 0.500,0.670,0.830));

```

```

49  outputVariable1->addTerm(new fl::Trapezoid("severe",
      0.830,0.910,1.000,1.000));
50  engine->addOutputVariable(outputVariable1);
51
52  fl::RuleBlock* ruleblock1 = new fl::RuleBlock;
53  ruleblock1->setName("");
54  ruleblock1->setTnorm(new fl::Minimum);
55  ruleblock1->setSnorm(new fl::Maximum);
56  ruleblock1->setActivation(new fl::Minimum);
57
58  ruleblock1->addRule(fl::MamdaniRule::parse("if density is low and speed is
      veryslow then congestion is slight", engine));
59  ruleblock1->addRule(fl::MamdaniRule::parse("if density is low and speed is
      slow then congestion is slight", engine));
60  ruleblock1->addRule(fl::MamdaniRule::parse("if density is low and speed is
      medium then congestion is free", engine));
61  ruleblock1->addRule(fl::MamdaniRule::parse("if density is low and speed is
      fast then congestion is free", engine));
62  ruleblock1->addRule(fl::MamdaniRule::parse("if density is medium and speed
      is veryslow then congestion is moderate", engine));
63  ruleblock1->addRule(fl::MamdaniRule::parse("if density is medium and speed
      is slow then congestion is slight", engine));
64  ruleblock1->addRule(fl::MamdaniRule::parse("if density is medium and speed
      is medium then congestion is slight", engine));
65  ruleblock1->addRule(fl::MamdaniRule::parse("if density is medium and speed
      is fast then congestion is free", engine));
66  ruleblock1->addRule(fl::MamdaniRule::parse("if density is high and speed is
      veryslow then congestion is severe", engine));
67  ruleblock1->addRule(fl::MamdaniRule::parse("if density is high and speed is
      slow then congestion is moderate", engine));
68  ruleblock1->addRule(fl::MamdaniRule::parse("if density is high and speed is
      medium then congestion is moderate", engine));
69  ruleblock1->addRule(fl::MamdaniRule::parse("if density is high and speed is
      fast then congestion is slight", engine));
70  ruleblock1->addRule(fl::MamdaniRule::parse("if density is veryhigh and
      speed is veryslow then congestion is severe", engine));
71  ruleblock1->addRule(fl::MamdaniRule::parse("if density is veryhigh and
      speed is slow then congestion is severe", engine));
72  ruleblock1->addRule(fl::MamdaniRule::parse("if density is veryhigh and
      speed is medium then congestion is moderate", engine));
73  ruleblock1->addRule(fl::MamdaniRule::parse("if density is veryhigh and
      speed is fast then congestion is moderate", engine));
74  engine->addRuleBlock(ruleblock1);
75
76  }

```

APÊNDICE B – PUBLICAÇÕES

Este apêndice lista os trabalhos publicados durante o mestrado.

1. ARAÚJO, G. B.; QUEIROZ, M. M.; DUARTE-FIGUEIREDO, F. L. P.; TOSTES, A. I. J.; LOUREIRO, A. A. F. **CARTIM: A Proposal Toward Identification and Minimization of Vehicular Traffic Congestion for VANET**. *IEEE 19th Symposium on Computers and Communications - ISCC*, p. 1-6, June 2014.
2. ARAÚJO, G. B.; DUARTE-FIGUEIREDO, F. L. P.; TOSTES, A. I. J.; LOUREIRO, A. A. F. **Um Protocolo de Identificação e Minimização de Congestionamentos de Tráfego para Redes Veiculares**. *XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC*, p. 1-14, maio 2014.