Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Proposta de um Protocolo de Disseminação de Mensagens em Redes Veiculares Ad Hoc para Aplicações Sensíveis à Acurácia de Posicionamento

Paulo Victor G. Farias

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador
Prof. Dr. Jacir Luiz Bordim

Brasília
2023

Ficha Catalográfica de Teses e Dissertações

Está página existe apenas para indicar onde a ficha catalográfica gerada para dissertações de mestrado e teses de doutorado defendidas na UnB. A Biblioteca Central é responsável pela ficha, mais informações nos sítios:

http://www.bce.unb.br
http://www.bce.unb.br/elaboracao-de-fichas-catalograficas-de-teses-e-dissertacoes

Esta página não deve ser inclusa na versão final do texto.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Proposta de um Protocolo de Disseminação de Mensagens em Redes Veiculares Ad Hoc para Aplicações Sensíveis à Acurácia de Posicionamento

Paulo Victor G. Farias

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof. Dr. Jacir Luiz Bordim (Orientador)
CIC/UnB

Prof. Dr. Eduardo Adilo Pelinson Alchieri
CIC/UnB

Prof. Dr. Jó Ueyama
ICMC/USP

Prof. Dr. Ricardo Pezzuol Jacobi
Coordenador do Programa de Pós-graduação em Informática

Brasília, 22 de setembro de 2023

Dedicatória

Dedico este trabalho aos meus pais, Gleide e Walmir e aos meus irmãos Guilherme, André Felipe, Caio e Ana Luísa.

Agradecimentos

Agradeço ao professor e orientador, Jacir Luiz Bordim, que proporcionou realizar a pesquisa nesta área de conhecimento. Agradeço aos comentários e sugestões da banca de avaliação que auxiliaram na melhoria do trabalho. Agradeço também à Universidade de Brasília, especialmente o Departamento de Ciência da Computação, que trabalham favorecendo o crescimento e aprendizado dos alunos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

As aplicações para prevenção de acidentes em Redes Veiculares Ad Hoc (VANETs) desempenham um papel importante para garantir a segurança de motoristas, passageiros e pedestres. Seu funcionamento depende de requisitos rígidos relacionados a acurácia de posicionamento dos veículos. Para atender essas restrições, as aplicações exigem o envio frequente de mensagens periódicas, também conhecidas como *beacons*, contendo a posição geográfica, velocidade e direção do veículo. Em cenários de tráfego denso, como em engarrafamentos ou interseções, a alta taxa de envio de *beacons* somada a difusão descoordenada de mensagens de alerta pode ocasionar em uma congestão na rede. Durante esse período, o canal de transmissão se torna saturado, aumentando a perda de pacotes por erros ou colisões, bem quanto o atraso de entrega das mensagens. Vários trabalhos têm sido propostos para aliviar a congestão em redes veiculares, porém a maioria não considera a perda de acurácia de posicionamento que pode ocorrer como efeito colateral das estratégias usadas. Nesse trabalho, propõe-se um novo protocolo para a disseminação de mensagens em VANETs chamado *Accurate Positioning Geocast Protocol* (APGP). O APGP utiliza uma arquitetura com três componentes para controlar a congestão causada por *beacons*, monitorar veículos vizinhos e transmitir mensagens de alerta. A abordagem emprega o ajuste de potência de transmissão para o envio de *beacons* para grupos de vizinhos, também chamados de grupos *geocast*, e ajuste de taxa de envio de mensagens. O APGP foi implementado em um ambiente de simulação de redes veiculares e testado sob um cenário veicular urbano. Os resultados indicaram que a proposta conseguiu reduzir o atraso de entrega em quatro vezes, bem quanto melhorias na taxa de entrega de pacotes de 43% e 23% quando comparado com um protocolo baseado no IEEE 802.11p [1] e o protocolo DC-BTRP [2]. Mesmo em cenários propensos a congestão, a taxa de ocupação de canal foi mantida abaixo de 60%, reduzindo em 47% e 15% em relação a outras alternativas. Em todas as condições, o APGP alcançou os níveis prescritos de acurácia de posicionamento para veículos em distâncias mais curtas.

Palavras-chave: redes ad hoc, redes veiculares, transmissão de mensagens, potência de transmissão

Abstract

Safety applications in Vehicular Ad Hoc Networks (VANETs) play an important role in ensuring the safety of drivers, passengers, and pedestrians. Their operation depends on strict requirements related to vehicle positioning accuracy. To meet these restrictions, applications require the frequent transmission of periodic messages, also known as beacons, containing the geographic position, speed, and direction of a vehicle. In dense traffic scenarios, the high beacon transmission rate summed with the uncoordinated broadcast of alert messages can lead to network congestion. During a congestion, the transmission channel becomes saturated, increasing packet loss by errors and collisions, as well as message delivery delay. Several works have been proposed to alleviate congestion in vehicular networks, however most do not consider the loss of positioning accuracy that can occur as a side effect of the strategies used. This work contributes with a study about the limitations of the transmission channel in vehicular networks, as well as detailed definition of the congestion issue. A novel protocol is proposed for message dissemination in VANETs called *Accurate Positioning Geocast Protocol* (APGP). APGP employs a three component architecture to control beacon congestion, track neighboring vehicles and deliver safety messages. The approach employs a transmission power adjustment for sending messages to groups of neighbors called geocast groups, and a message transmission rate adjustment. APGP was implemented in a simulation environment for vehicular networks and tested in an urban vehicular scenario. The results indicated that the proposal achieved a reduction in delivery delay of up to four times, as well as improvements in sspacket delivery ratio of around 43% and 23% when compared to a protocol based on IEEE 802.11p [1] and the protocol DC-BTRP [2]. Even in scenarios prone to network congestion, channel busy ratio remained below 60%, exhibiting a reduction of 47% and 15% in comparison to other options. In all conditions, APGP successfully achieved the prescribed levels of positioning accuracy for vehicles at shorter distances.

Keywords: ad hoc networks, vehicular networks, message dissemination, transmission power

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

3GPP *3rd Generation Partnership Project.*

AC3 *Adaptive Transmit Power Cooperative Congestion Control.*

APGP *Accurate Positioning Geocast Protocol.*

API Interface de Programa de Aplicações (do Inglês, *Application Programming Interface.*

ATB *Adaptive Traffic Beacon.*

BCC *Beacon Congestion Control.*

BPR *Bayesian Personalized Ranking.*

BS *Base Stations.*

BSM *Basic Safety Message.*

BTPC *Beacon Transmission Power Control .*

C-V2X *Cellular Vehicle-to-Everything.*

CACC *Channel-Aware Congestion Control Algorithm.*

CAM *Cooperative Awaraness Message.*

CBR *Channel Busy Ratio.*

CCH Canal de Controle (do inglês, *Control Channel*).

CCHI *Control Channel Interval.*

CH *Cluster Head.*

CSMA/CA *Carrier Sense Multiple Access with Collision Avoidance.*

D-FPAV *Distributed Fair Power Adjustment for Vehicular Environments.*

D2D *Device-to-Device Communication.*

DBSMA *Dynamic Broadcast Storm Mitigation Algorithm.*

DC-BTRP *Dynamic Control of Beacon Transmission Rate and Power.*

DSRC Comunicações Dedicadas de Curto Alcance (do Inglês, *Dedicated Short-Range Communications*).

DynB *Dynamic Beaconing.*

EEBL *Electronic Emergency Brake Light.*

ETSI *European Telecommunications Standards Institute.*

EVW *Emergency Vehicle Warning.*

FCW *Forward Collision Warning.*

GPS *Global Positioning System.*

HALL *High Availability Low Latency.*

HERO *Heuristic Routing for Vehicular Networks.*

HMM Modelo Oculto de Markov (do Inglês, *Hidden Markov Model*).

IEEE *Institute of Electrical and Electronics Engineers.*

IPD *Inter-Packet Delay.*

IPv6 Protocolo da Internet versão 6 (do Inglês, *Internet Protocol version 6*).

ITS Sistemas de Transporte Inteligente (do Inglês, *Intelligent Transportation Systems*).

LCA *Lane Change Advisor.*

LLC Controle de Enlace Lógico (do Inglês, *Logic Link Control*).

LTE *Long Term Evolution.*

MAC *Media Access Control.*

MANET Rede Móvel Ad-Hoc (do Inglês, *Mobile Ad Hoc Network*).

MIB *Management Information Base.*

MLME *MAC Sublayer Management Entity.*

MPBR *Mobility Prediction Based Beacon Rate Adaptation.*

NHTSA *National Highway Traffic Safety Administration.*

NT *Neighbor Tracking.*

OBU *On-Board Unit.*

OCP *Optimised CSMA/CA protocol.*

OMNeT *Objective Modular Network Testbed in C++.*

PCW *Predictive Contention Window.*

PDR *Packet Delivery Ratio.*

PLME *Physical Layer Management Entity.*

PSID *Provider Service Identifier.*

RSS *Receiver Signal Strength.*

RSU *Roadside Unit.*

SBAPC *Speed Based Adaptive Power Control.*

SCH Canal de Serviço (do Inglês, *Service Channel*).

SCHI *Service Channel Interval.*

SDN *Software Defined Networks.*

SESAC *SDN-Enabled Social-Aware Clustering.*

SGD *Stochastic Gradient Descent.*

SIFS *Short Interframe Space.*

SMD *Safety Message Delivery.*

SPDR *Speed and Position aware Dynamic Routing.*

SUMO *Simulation of Urban Mobility.*

TCP *Transfer Control Protocol.*

TRI *Two-Ray Interference.*

UDP *User Datagram Protocol.*

UFC *Unified Framework of Clustering.*

UMBP *Urban Multi-hop Broadcast Protocol.*

V2I Comunicação de Veículo para Infraestrutura (do Inglês, *Vehicle-to-Infrastructure Communication*).

V2V Comunicação de Veículo para Veículo (do Inglês, *Vehicle-to-Vehicle Communication*).

V2V2I Comunicação de Veículo para Veículo para Infraestrutura (do Inglês, *Vehicle-to-Vehicle-to-Infrastructure Communication).*

V2X *Vehicle-to-Everything.*

VANET Rede Veicular Ad-Hoc (do Inglês, *Vehicular Ad-Hoc Network*).

Veins *Vehicles in Network Simulation.*

VSCA *Vehicular Safety Communications - Applications.*

WAVE *Wireless Access in Vehicular Environments.*

WLAN *Wireless Local Area Networks.*

WME *WAVE Management Entity.*

WSA *WAVE Service Advertisement.*

WSM *WAVE Short Message.*

WSMP *WAVE Short Message Protocol.*

XML *Extensible Markup Language.*

ZdI Zona de Interesse.

Capítulo 1

Introdução

À medida que as redes sem fio evoluíram, os Sistemas de Transporte Inteligente (do Inglês, *Intelligent Transportation Systems*) (ITS) foram amplamente pesquisados e implantados com aplicações avançadas, que visam melhorar a gestão do transporte e o fluxo de informações entre veículos [10]. As redes veiculares são uma das principais contribuições dos ITS, pois buscam evitar acidentes, reduzir congestionamentos e garantir conforto para motoristas e passageiros. Inicialmente, o foco principal das redes veiculares era a prevenção de acidentes e essa tecnologia ficou conhecida como Comunicações Dedicadas de Curto Alcance (do Inglês, *Dedicated Short-Range Communications*) (DSRC) [11]. Posteriormente, a emenda IEEE 802.11p, também conhecida como *Wireless Access in Vehicular Environments* (WAVE) [1], definiu uma arquitetura para as comunicações veiculares e padronizou um conjunto de serviços no ambiente veicular.

As redes veiculares também são conhecidas como Rede Veicular Ad-Hoc (do Inglês, *Vehicular Ad-Hoc Network*) (VANET), pois os veículos são equipados com um dispositivo eletrônico chamado *On-Board Unit* (OBU) que permite a troca de mensagens sem depender de uma infraestrutura externa. Os modos de comunicação em VANETs podem ser classificados como Comunicação de Veículo para Veículo (do Inglês, *Vehicle-to-Vehicle Communication*) (V2V) e Comunicação de Veículo para Infraestrutura (do Inglês, *Vehicle-to-Infrastructure Communication*) (V2I) [12].

Nas comunicações V2V, veículos trocam mensagens entre si com o objetivo principal de evitar acidentes. Entre as aplicações desse modo é possível citar *Electronic Emergency Brake Light* (EEBL), *Forward Collision Warning* (FCW) e *Lane Change Advisor* (LCA) [13]. Nas comunicações V2I, veículos podem trocar mensagens com uma infraestrutura localizada próxima à pista conhecida como *Roadside Unit* (RSU), que provê serviços como monitoramento de tráfego e sistemas inteligentes para estacionamento. A Comunicação de Veículo para Veículo para Infraestrutura (do Inglês, *Vehicle-to-Vehicle-to-Infrastructure Communication)* (V2V2I) combina as características de V2V e V2I para prover uma forma

de comunicação híbrida [14].

Para repassar as informações de eventos críticos nas comunicações V2V, os veículos utilizam um mecanismo conhecido como difusão ou *broadcast*, onde a transmissão de uma mensagem é feita para todos os vizinhos no alcance. Dessa forma, torna-se possível conscientizar todos os veículos em um raio de alcance sobre algum evento crítico. Um exemplo comum de uma aplicação V2V é o *Emergency Vehicle Warning* (EVW), onde um veículo especial, como uma ambulância, transmite mensagens de alerta com antecedência para indicar sua necessidade de utilizar uma rota específica. Essas mensagens podem então ser repassadas para veículos mais a frente com intuito de facilitar a sua passagem e poupar tempo.

A necessidade de comunicações com baixa latência e alta confiabilidade tornam as redes veiculares um sistema crítico, onde o bom desempenho das aplicações está intrinsecamente ligado com os protocolos e algoritmos de comunicação utilizados. Por este motivo, torna-se essencial garantir que os padrões atuais permitam que as mensagens sejam transmitidas com o menor atraso possível e com informações acuradas para garantir a segurança da vida de motoristas, passageiros e pedestres.

1.1 Definição do Problema

O foco desse trabalho é nas comunicações de veículo para veículo ou simplesmente V2V no padrão IEEE 802.11p [1]. Nesse modo, os veículos não têm o auxílio de uma infraestrutura externa para coordenar a transmissão de mensagens ou como elas devem ser roteadas. Por esse motivo, cada nó presente na rede é responsável por enviar mensagens periódicas para seus vizinhos com informações como sua posição de GPS, velocidade e direção [15]. Essas mensagens podem receber o nome de *beacons* e são enviadas para nós vizinhos a um salto de distância. Sua principal utilidade é conscientizar outros veículos para prevenir acidentes e auxiliar na construção da topologia da rede.

Aplicações projetadas para prevenção de acidentes em VANETs estabelecem que *beacons* devem ser enviados em altas taxas, variando entre 10 a 50 por segundo [16]. Esse comportamento é justificado em virtude da natureza extremamente móvel e imprevisível do ambiente veicular, que rapidamente torna obsoleta as informações de posicionamento após pequenas janelas de tempo. As aplicações em VANETs também devem transmitir mensagens de alerta, que muitas vezes são repassadas por veículos para outros vizinhos mais distantes numa zona de interesse.

Os *beacons* são essenciais para encontrar a rota com menor atraso para transmitir e rotear essas mensagens de alerta. Entretanto, esse processo gera uma carga adicional ao canal de transmissão. Em ambientes com tráfego denso, o quantitativo de *beacons* e men-

sagens de alerta geradas simultaneamente pode facilmente ocasionar em uma saturação do canal de transmissão [17]. Assim como nas redes sem fio do padrão IEEE 802.11 [18], VANETs também estão propensas a perda de pacotes por erros ou colisões durante um período de congestão do canal. Quanto mais mensagens são perdidas, pior será a percepção do veículo sobre o posicionamento de seus vizinhos. Em alguns casos, as mensagens de alerta sobre eventos críticos serão perdidas ou enviadas com atraso, colocando vidas em risco.

Algoritmos e protocolos têm sido propostos para lidar com o problema de congestão causado por envio excessivo de mensagens em VANETs. Essas estratégias podem ter como foco o envio de *beacons* [19–22], visando melhorar as condições do canal por técnicas como o ajuste da taxa de envio de mensagens [19, 20] e da potência de transmissão [21, 22]. Outras abordagens buscam melhorar como as mensagens de alerta são repassadas [23–26], adotando técnicas reativas [23, 24] ou proativas [25, 26] para a escolha de nós encaminhadores.

Entretanto, observou-se que a maioria das propostas tem como único objetivo melhorar as condições do canal de transmissão no que tange a recepção de pacotes e ocupação do canal. Porém, não consideram a perda de acurácia de posicionamento que as técnicas utilizadas podem gerar como efeito colateral. Esse problema pode afetar negativamente as aplicações com foco em prevenção de acidentes, que definem requisitos rígidos sobre o posicionamento dos veículos. Além disso, as operações multicanal introduzidas no padrão IEEE 1609.4 [5] limitam ainda mais a capacidade do canal de transmissão e são geralmente ignoradas durante os testes dos trabalhos existentes. Considerando os fatores mencionados, pode-se afirmar que os métodos atuais não são suficientes para atender às necessidades das aplicações com foco na segurança dos usuários. Em especial, aos requisitos de acurácia de posicionamento.

1.2 Objetivos e Requisitos

1.2.1 Objetivos Gerais

Esse trabalho têm como principal objetivo contribuir para o aprimoramento do campo de comunicação em VANETs, oferecendo uma proposta eficaz para melhorar a confiabilidade e a eficiência da troca de informações entre veículos. Primeiramente, identifica-se e categoriza as estratégias presentes na literatura para lidar com o problema de congestão em VANETs, destacando seus pontos positivos e negativos. Após isso, ele investiga e avalia o desempenho do *Accurate Positioning Geocast Protocol* (APGP), uma nova abordagem para a transmissão de mensagens em uma Rede Veicular Ad-Hoc por meio de *geocast*. O

protocolo garantirá que os requisitos de acurácia de posicionamento das aplicações com foco em prevenção de acidentes sejam atendidos mesmo em cenários com uma alta densidade de veículos, onde o canal de transmissão está propenso a falhas de transmissão e colisões.

1.2.2 Requisitos da Proposta

- **RQ1:** Diminuir a taxa de ocupação do canal com uma estratégia de ajuste de potência consciente, ou seja, que garante que todos os veículos dentro do raio de transmissão estabelecido recebam mensagens.
- **RQ2:** Propor mecanismos que possibilitem alterar a taxa de envio de *beacons*, sem perda da acurácia de posicionamento, considerando uma predição de posição do veículo.
- **RQ3:** Manter uma lista de candidatos (vizinhos) para auxiliar no envio de mensagens de alerta das aplicações sobre possíveis eventos, como acidentes e problemas na pista.
- **RQ4:** Apresentar um mecanismo confiável e com baixo atraso para a transmissão das mensagens de alerta sobre eventos críticos.

1.3 Metodologia

Primeiramente, uma revisão do estado da arte no que tange aos trabalhos com foco em algoritmos e protocolos em VANETs será realizada. Os trabalhos mais recentes e relevantes ao problema serão discutidos, considerando seus pontos positivos e negativos. Após isso, uma proposta denominada *Accurate Positioning Geocast Protocol* (APGP) será desenvolvida em um ambiente virtual de simulação para redes veiculares. O desempenho do protocolo será comparado com outras estratégias via experimentos realizados em um cenário que se assemelha ao tráfego urbano. Após a coleta de dados das simulações, os resultados serão apresentados por gráficos e tabelas para demonstrar o comportamento do protocolo. Serão consideradas métricas relevantes para essa avaliação como a taxa de recepção de pacotes, taxa de ocupação do canal, atraso de entrega e erro de posicionamento.

1.4 Estruturação

O restante desse trabalho é organizado da seguinte forma:

- **Capítulo 2:** trata da fundamentação teórica relativamente às redes veiculares, destacando suas características, aplicações, tecnologias, padrões e simuladores.
- **Capítulo 3:** traz uma revisão do estado da arte das estratégias de controle e prevenção de congestão em VANETs propostas na literatura, onde uma análise comparativa dos resultados observados é realizada.
- **Capítulo 4:** define o problema de congestão em VANETs com mais detalhes e descreve o funcionamento da proposta do protocolo *Accurate Positioning Geocast Protocol* (APGP).
- **Capítulo 5:** mostra o cenário de simulação utilizado e relata os resultados obtidos das simulações.
- **Capítulo 6:** apresenta as considerações finais sobre a proposta, contribuições e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, apresenta-se uma fundamentação teórica sobre os avanços em redes veiculares, destacando as suas principais características e aplicações. As tecnologias existentes para a implantação de uma rede veicular serão discutidas, onde é possível as categorizar em redes baseadas em *Wireless Local Area Networks* (WLAN) e baseadas em redes celulares, que também são conhecidas como *Cellular Vehicle-to-Everything* (C-V2X). Padrões comuns em redes veiculares como DSRC/WAVE e ETSI ITS-G5 também serão abordados. Por fim, uma breve discussão é feita acerca dos programas utilizados para a simulação em redes veiculares com foco em mobilidade, tráfego e rede.

2.1 Características e Modos de Operação em Redes Veiculares

À medida que as redes sem fio evoluíam, os Sistemas de Transporte Inteligente (do Inglês, *Intelligent Transportation Systems*) (ITS) [10] começaram a ser propostos para melhorar a segurança, eficiência e sustentabilidade das redes de transporte. As redes veiculares são um dos principais focos de estudo dos ITS e visam reduzir o congestionamento no tráfego, evitar acidentes e trazer conforto e entretenimento para motoristas, passageiros e pedestres. Nos Estados Unidos, as pesquisas sobre comunicações veiculares, também comumente chamadas de *Vehicle-to-Everything* (V2X), começou com as Comunicações Dedicadas de Curto Alcance (do Inglês, *Dedicated Short-Range Communications*) (DSRC) [11], que se concentrou principalmente na prevenção de acidentes e na melhoria das condições de tráfego. Posteriormente, a emenda IEEE 802.11p [1], definiu uma arquitetura para comunicações veiculares e um conjunto padronizado de serviços no ambiente veicular que ficou conhecido como *Wireless Access in Vehicular Environments* (WAVE). Na Europa,

um padrão semelhante chamado ETSI ITS-G5 [27] foi proposto e que também se baseia na emenda 802.11p [1].

As redes veiculares podem ser conhecidas como Rede Veicular Ad-Hoc (do Inglês, *Vehicular Ad-Hoc Network*) (VANET), que configura um tipo especial de Rede Móvel Ad-Hoc (do Inglês, *Mobile Ad Hoc Network*) (MANET). Em uma rede ad-hoc sem fio não existe uma infraestrutura externa que coordene as comunicações. A tarefa de transmitir mensagens entre si, fica em cargo dos nós, bem quanto a construção da topologia da rede. O mesmo vale para as VANETs, já que são formadas por veículos equipados com um dispositivo eletrônico de comunicação sem fio chamado *On-Board Unit* (OBU), que permite a transmissão e recepção de mensagens sem depender de uma infraestrutura.

As VANETs, em especial, apresentam algumas características que as tornam diferente de outros tipos de MANETs. Primeiramente, a topologia da rede é extremamente dinâmica, pois os nós estão em constate movimento, em altas velocidades e possivelmente em diferentes direções. O tempo de envio e resposta de mensagens é um fator crítico, em especial para aplicações de prevenção de acidentes. Os padrões de mobilidades dos veículos podem ser previstos considerando a pista e rota desejada. O tamanho de uma VANET pode se tornar um problema por conta das limitações do canal de transmissão em relação à disponibilidade da largura de banda.

Uma rede veicular apresenta dois modos de operação principais. A Comunicação de Veículo para Infraestrutura (do Inglês, *Vehicle-to-Infrastructure Communication*) (V2I), indica que os veículos trocam mensagens com uma infraestrutura localizada à beira da estrada conhecida como *Roadside Unit* (RSU). Essa unidade está geralmente conectada à Internet e pode prover serviços tanto para a segurança como entretenimento do usuário. A Comunicação de Veículo para Veículo (do Inglês, *Vehicle-to-Vehicle Communication*) (V2V) indica que os veículos trocam mensagens entre si para criar uma rede sem fio. Este modo é recomendado para aplicações onde o atraso na entrega de mensagens é um fator determinante, como em aplicações de prevenção de acidentes. Ainda existe outro modo de operação que visa unir as características dos dois modos apresentados e ficou conhecido como Comunicação de Veículo para Veículo para Infraestrutura (do Inglês, *Vehicle-to-Vehicle-to-Infrastructure Communication)* (V2V2I) [14]. Neste modo de operação, veículos podem utilizar o RSU para se comunicar e também a troca de mensagens com outros veículos. A Figura 2.1, apresenta um exemplo do funcionamento usual de uma VANET.

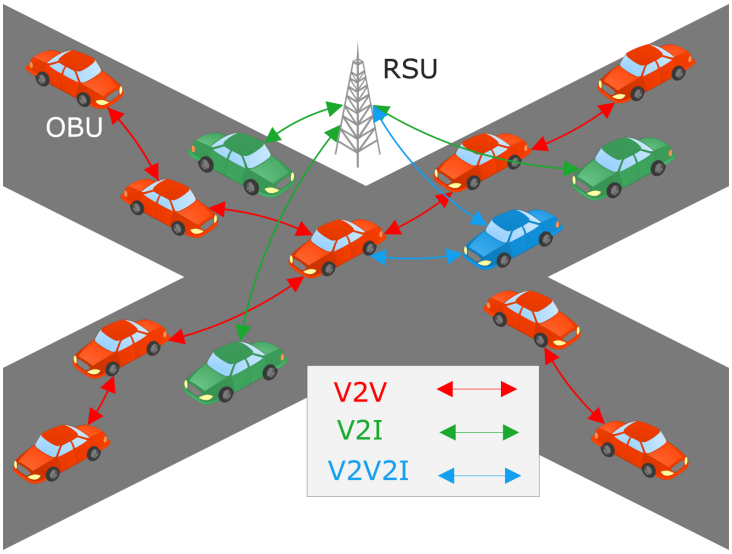

Figura 2.1: Modos de operação de uma VANET.

2.2 Tecnologias e Padrões para VANETs

Nesta seção, serão discutidas as tecnologias e padrões utilizados para implantar as VANETs, destacando-se o modo de operação V2V que é o foco principal desse trabalho. As duas tecnologias mais populares são as redes baseadas em WLAN com os padrões DSRC/WAVE [1, 28], ETSI ITS-G5 e as redes celulares, que também levam o nome de C-V2X.

2.3 VANETs Baseadas em WLAN (DSRC/WAVE)

O avanço das WLANs impulsionou os estudos do que hoje é conhecido como redes veiculares ou simplesmente VANETs. Em 1999, o governo dos Estados Unidos propôs as Comunicações Dedicadas de Curto Alcance (do Inglês, *Dedicated Short-Range Communications*) (DSRC) com principal objetivo de evitar acidentes e melhorar condições de tráfego. 75 MHz de largura de banda foram alocados próximos a uma frequência de banda de 5.9 GHz para esta tecnologia [29]. Em 2010, uma emenda ao padrão das redes sem fio usuais, o 802.11 [18], foi proposta e aprovada pelo IEEE e ficou conhecida como *Wireless Access in Vehicular Environments* (WAVE) ou 802.11p [1].

O foco principal da pilha de comunicações DSRC/WAVE, denotada na Figura 2.2, é dar suporte para operações com pouca latência e de curto alcance. Estes padrões vêm sendo atualizados constantemente nos últimos anos, com a adição de diversas emendas adicionais como a 1609.1 [30] que define um gerenciador de recursos para as VANETs, a 1609.2 [31] que define serviços de segurança para aplicações e gerenciamento de mensagens,

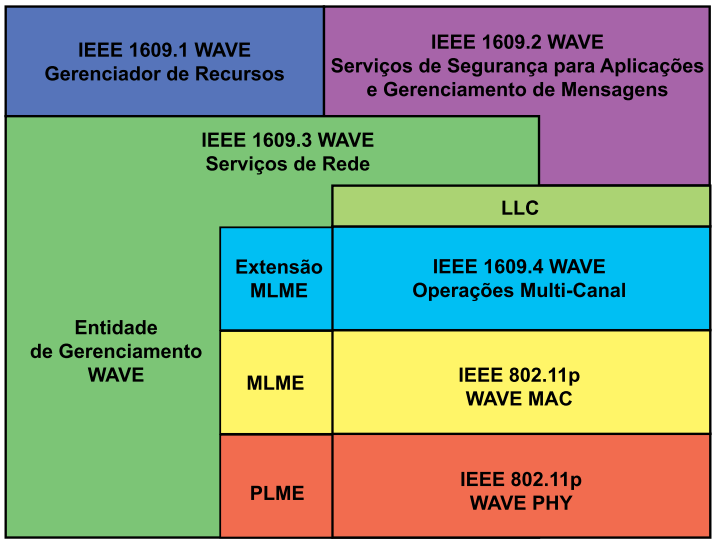

Figura 2.2: Pilha de comunicação DSRC/WAVE (adaptado de [3]).

a 1609.3 [32] que aborda serviços de rede e a 1609.4 [5] que lida com operações em multi-canais, permitindo o uso de aplicações de diferentes tipos em simultâneo.

2.3.1 Camada Física - IEEE 802.11p

A camada física de uma pilha de comunicação é responsável por transformar ondas eletromagnéticas ou sinais elétricos em unidades básicas chamadas *bits*. Portanto, esta camada se encarrega de receber e transmitir dados fisicamente em um meio de propagação. Em especial, a pilha de comunicação DSRC/WAVE define as modificações necessárias para a adaptação do padrão de redes sem fio 802.11 para o ambiente veicular através do padrão 802.11p [1]. Como mencionado na seção anterior, 75 MHz de largura de banda são alocados entre as frequências de banda $5.850 - 5.925$ GHz [29] para o uso em redes veiculares. Os sistemas DSRC suportam velocidades de até 200 km/h, com um alcance de transmissão de $300 - 1000$ m e uma taxa de dados padrão de 3 ou 6 Mbps (podendo chegar a até 27 Mbps).

Outra característica da camada física do 802.11p é a existência de sete canais de operação, conforme ilustrado na Figura 2.3. Estes canais são divididos em Canal de Controle (do inglês, *Control Channel*) (CCH) e Canal de Serviço (do Inglês, *Service Channel*) (SCH). O canal 178, do tipo CCH, é utilizado para transmissão de mensagens de controle e disseminação de mensagens de alta prioridade. Os canais 174, 176, 180 e 182, do tipo SCH, são usados para a disseminação de mensagens de baixa prioridade não relacionadas a segurança. Em especial, o canal 172 é utilizado em situações onde existe risco crítico de segurança ao usuário, enquanto o canal 184 é conhecido como *High*

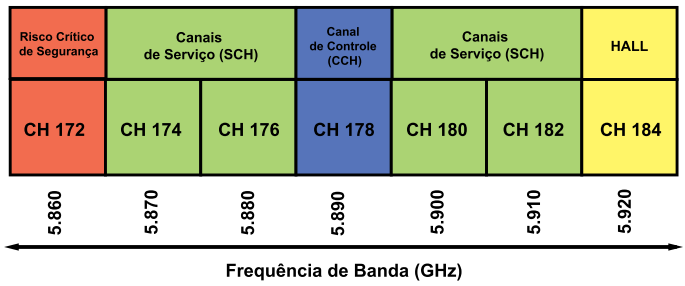

Figura 2.3: Canais de comunicação no padrão IEEE 802.11p (adaptado de [4]).

Availability Low Latency (HALL) e deve ser utilizado futuramente. Ainda na camada física, existe uma unidade conhecida como *Physical Layer Management Entity* (PLME) responsável por fornecer uma interface de serviço de gerenciamento da camada através da qual as funções gerenciamento podem ser invocadas.

2.3.2 Camada de Enlace - IEEE 802.11p e 1609.4

A camada de enlace na pilha DSRC/WAVE tem como principal objetivo prover um método de acesso confiável, justo e eficiente aos canais de transmissão. Por este motivo, ela possui a Controle de Enlace Lógico (do Inglês, *Logic Link Control*) (LLC) que é responsável por controlar a sincronização de quadros, realizar controle de fluxo e checagem de erros. Além disso, a camada de enlace também possui mecanismos para seleção do canal, roteamento de canal e também para a definição da prioridade do usuário. Uma unidade conhecida como *MAC Sublayer Management Entity* (MLME) e uma extensão também estão presentes nesta camada, juntas elas ficam responsáveis por coordenar o acesso aos canais e o mecanismo de troca entre eles.

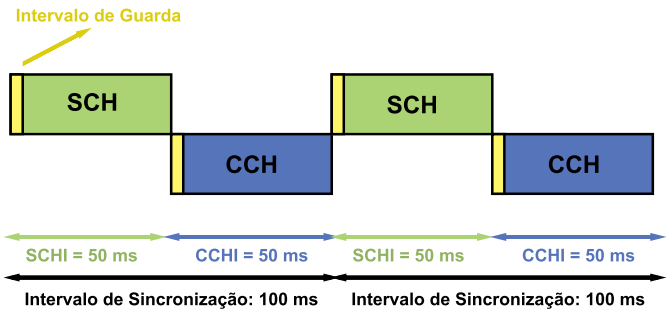

Figura 2.4: Acesso aos canais no padrão 802.11p (adaptado de [5]).

O acesso ao canal de transmissão acontece de dois modos, no acesso contínuo, o dispositivo só consegue usufruir do canal do tipo CCH, não podendo utilizar nenhum tipo de serviço que não seja relacionado a segurança. No acesso alternado, definido no padrão 1609.4 [5], um método para operar em ambos os tipos de canais foi introduzido, segundo o esquema da Figura 2.4. O tempo de sincronização de 100 ms é dividido em dois intervalos de aproximadamente 50 ms conhecidos como *Control Channel Interval* (CCHI) e *Service Channel Interval* (SCHI). Durante o CCHI, o veículo pode transmitir e receber mensagens de controle e de segurança, enquanto no SCHI, o veículo poderá receber e transmitir mensagens não relacionadas a segurança.

2.3.3 Camadas de Rede, Transporte e Aplicação - IEEE 1609.1, 1609.2 e 1609.3

As camadas de rede, transporte e aplicação da pilha DSRC/WAVE são detalhadas nas emendas IEEE 1609.1 [30], 1609.2 [31] e 1609.3 [32]. No padrão 1609.3 são definidas as características das camadas de rede e transporte onde três opções para a transmissão de mensagens são definidas, sendo elas, o *WAVE Short Message Protocol* (WSMP), UDP/IPv6 e TCP/IPv6. Entretanto, o uso de protocolos como o TCP não apresenta bom desempenho em ambientes móveis como o das VANETs [33]. Ainda no padrão 1609.3, também é definida a *WAVE Management Entity* (WME), responsável por realizar tarefas como manutenção da *Management Information Base* (MIB), processar requisições de serviço de aplicações e monitorar mensagens do tipo *WAVE Service Advertisement* (WSA), que anunciam os serviços disponíveis pelos RSUs.

Em especial, para comunicações V2V é possível destacar o uso do WSMP, criado especialmente para comunicações de curto alcance, de alta prioridade e sensíveis ao tempo. Neste protocolo, os veículos se comunicam via mensagens conhecidas como *WAVE Short Message* (WSM). Esta mensagem contém informações pertinentes tanto para a camada de rede como para a camada de transporte e seu cabeçalho é dividido em WSMP-N e WSMP-T. As informações presentes no WSMP-N indicam a versão do WSMP e o tipo de protocolo de rede usado. Além disso, um campo especial de extensão também está disponível e pode conter o número do canal, potência de transmissão, taxa de dados e carga do canal. No cabeçalho WSMP-T existe um campo chamado PSID utilizado para indicar o tipo de serviço provido por outros dispositivos e representa a forma de endereçamento na pilha DSRC/WAVE, além disso, o comprimento do WSM também está presente neste cabeçalho.

Em redes veiculares, o roteamento de informações pode ser efetuado através da comunicação *unicast*, *multicast*/*geocast* ou difusão (*broadcast*), como ilustrados na Figura

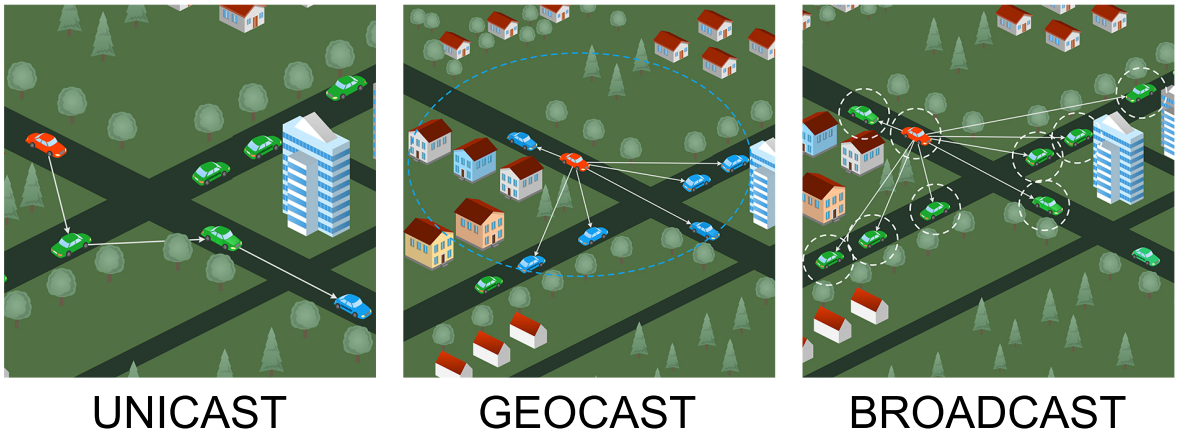

Figura 2.5: Modos de disseminação de mensagens em VANETs.

2.5. No método de disseminação de mensagens *unicast*, o principal objetivo é transmitir dados entre um nó de origem e um nó de destino através de uma comunicação sem fio multi-salto. Em VANETs é comum que veículos tenham que enviar mensagens para vários nós, portanto o *unicast* não é um método viável para a maioria das aplicações. Na comunicação *multicast* ou *geocast*, um nó de origem tem como intenção se comunicar com um grupo de nós destino. O *geocast* é uma forma especial do *multicast* onde as mensagens são enviadas de acordo com uma posição geográfica em particular em relação ao nó de origem. A distância percorrida por estas mensagens é coordenada conforme a potência de transmissão. Na comunicação por difusão ou *broadcast*, um nó de origem envia informações para todos os seus vizinhos em simultâneo, visando cobrir a maior área possível.

As aplicações para VANETs podem ser divididas em duas categorias principais, as aplicações de segurança e as não relacionadas a segurança. Outras classificações mais específicas também foram sugeridas em Cunha *et al.* [34], onde cinco categorias são definidas: aplicações de segurança, eficiência, conforto, entretenimento interativo e de sensoriamento urbano. Em aplicações de segurança, o foco principal é evitar acidentes nas pistas, por este motivo, a entrega de mensagens sem atrasos é um fator determinante. Algumas das aplicações nessa categoria incluem *Electronic Emergency Brake Light* (EEBL), *Forward Collision Warning* (FCW) e *Lane Change Advisor* (LCA).

Aplicações de eficiência, no que lhe concerne, visam melhorar as condições de mobilidade do tráfego e ainda podem ser divididas em duas sub-categorias, aplicações para controle de tráfego em cruzamentos e as para controle de congestão na pista. As aplicações de conforto visam auxiliar o motorista com informações obtidas de serviços que possam auxiliá-lo durante sua viagem como previsão do tempo, vagas de estacionamento livres e pontos turísticos. As aplicações de entretenimento interativo buscam oferecer informações

relacionadas ao entretenimento para motoristas e passageiros como acesso à Internet, jogos, compartilhamento de arquivos, entre outras. Por fim, as aplicações de sensoriamento podem ser usadas para o monitoramento das condições ambientais e atividades sociais no meio urbano.

2.4 VANETs Baseadas em LTE e 5G (C-V2X)

As redes veiculares também podem ser implantadas por comunicações celulares, essa tecnologia utilizada com a V2X recebeu o nome de *Cellular Vehicle-to-Everything* (C-V2X) e inclui sistemas V2X baseados em *Long Term Evolution* (LTE) e 5G. Pesquisas sobre os requisitos técnicos e padrões necessários para o uso dessa tecnologia foram iniciados pela *3rd Generation Partnership Project* (3GPP) [35] em 2015. Em especial, destaca-se a comunicação *Device-to-Device Communication* (D2D) para o modo de operação V2V, onde os veículos transmitem dados entre si sem depender de uma infraestrutura externa. Diferente das redes ad-hoc, a comunicação é realizada diretamente entre os dispositivos sem algoritmos de roteamento.

Alguns autores acreditam que o uso das redes celulares e do 5G virão para solucionar problemas do padrão IEEE 802.11p como falta de espectro, baixa latência e necessidade de comunicações periódicas e confiáveis em altas taxas [36]. Enquanto outros autores acreditam que ambas as tecnologias como o IEEE 802.11p e o 5G com as comunicações C-V2X devem coexistir para prover as melhores soluções em comunicações veiculares [37].

2.5 Ambiente de Simulação para Redes Veiculares

A simulação em redes veiculares é uma solução com alto custo-benefício para modelar e analisar o desempenho em um ambiente que se aproxime ao máximo do mundo real. Existem diversas ferramentas para realizar essa tarefa em VANETs, onde os simuladores podem ser divididos em duas categorias principais, simuladores de mobilidade tráfego e de rede. Ambos os tipos podem ser utilizados independentemente, mas para simular uma VANET real, geralmente deve existir um *framework* de integração entre mobilidade, tráfego e rede [38]. A Figura 2.6 serve como orientação sobre os simuladores e *frameworks* existentes, sendo que as opções mais utilizadas serão abordadas com mais detalhes.

2.5.1 Simuladores de Mobilidade e Tráfego

Os simuladores de mobilidade e tráfego visam determinar as condições da pista como seu comprimento, número de faixas, densidade de veículos, velocidade dos veículos, semáforos,

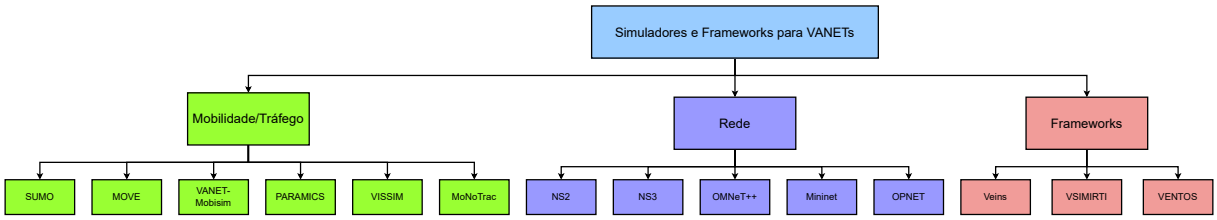

Figura 2.6: Simuladores e *frameworks* para VANETs e seus tipos.

etc. Em especial, essa simulação pode ser realizada de forma macroscópica, determinando características gerais de um fluxo de tráfego, ou microscópica, que define características individuais para cada veículo inserido na simulação. Eles podem ser utilizados individualmente para a simulação de tráfego em largas áreas geográficas. Dentre os mais populares, destacam-se:

- **SUMO [39]:** simulador de código aberto que permite que o usuário simule situações de colisão, diferentes categorias de veículos, controle de velocidade e fluxos. Com ele é possível importar e editar mapas de zonas reais e apresenta uma interface gráfica intuitiva. Também pode ser usado em conjunto com os simuladores de rede NS2, NS3 e OMNeT++.
- **MOVE [40]:** simulador construído como uma extensão do SUMO que permite criar ambientes simples através de uma interface gráfica. Com ele é possível editar mapas facilmente. Além disso, também pode ser usado com simuladores como o NS2 e GloMoSim.
- **VANET-MobiSim [41]:** simulador de mobilidade que utiliza modelos de mobilidade inteligentes que permitem a troca de faixas e controle de interseções. Além disso, é possível importar outros modelos de mobilidade. Esta ferramenta foi desenvolvida em Java, diferente das outras duas opções apresentadas (SUMO e MOVE) implementadas em C++,
- **PARAMICS [42]:** provê ferramentas que permitem visualização em 3D e gerações de tráfego em larga escala. Disponível comercialmente.
- **VISSIM [43]:** permite implementar fluxos de tráfego microscópicos, mas também dá suporte para fluxos macroscópicos. A saída pode ser exportada para plataformas de gráficos 3D como o AutoCAD.
- **MoNoTrac [44]:** outra opção implementada em Java que utiliza dados geográficos reais para criar um ambiente de simulação. Também permite a edição de características da simulação por arquivos XML, semelhantemente ao SUMO.

2.5.2 Simuladores de Rede

Os simuladores de rede são utilizados para simular eventos discretos que coordenam a interação entre os módulos em um ambiente virtual em tempo real. Para cada veículo, deve haver um módulo que implementa e executa os protocolos de comunicação. Ele deve ser usado em conjunto com o simulador de mobilidade para compor uma VANET. Em especial, deve prover as funcionalidades detalhadas nos padrões de redes veiculares observados na pilha DSRC/WAVE na Seção 2.3. Dentre as opções existentes, destacam-se os seguintes simuladores:

- **NS2 [45] e NS3 [46]:** são simuladores de código aberto, inicialmente o NS2 não dava suporte ao padrão IEEE 802.11p, mas em versões posteriores ele foi adicionado incluindo modelos de propagação do rádio e mobilidade aos nós. Nesta versão do simulador, a escalabilidade se mostra um problema. O NS3 é uma versão melhorada do NS2 que dá suporte ao processamento em paralelo e simulações distribuídas. Entretanto, ainda apresenta atrasos de transmissão, pois segundo Shabir *et al.* [6] muitos pacotes não podem ser entregues ao destino de uma vez.
- **OMNeT++ [47]:** é um simulador de código aberto implementado em C++. Ele apresenta uma interface gráfica e o sistema de simulação é definido através da integração de módulos. Ele apresenta tempo de computação e simulação rápidos, além de ser eficiente quanto ao uso de memória, também apresenta menor atraso. Também pode apresentar problemas de escalabilidade.
- **Mininet [48]:** é um simulador de código aberto que permite criar ambientes com uso de APIs Python. Ele dá suporte para o IEEE 802.11p e permite simular redes veiculares com SDN. Entretanto, não é possível analisar o desempenho da rede por conta da taxa imprevisível de encaminhamento de dados dos *switches open flow*, portanto só pode se analisar o comportamento da rede.
- **OPNET [49]:** é um simulador disponível comercialmente. Mediante uma interface gráfica ele permite simular sistemas reais com um paradigma orientado a objeto. Permite comunicação com outros tipos de protocolos e formatos de pacotes que podem ser editados. Além disso, ele é complexo e pode levar tempo para se utilizar corretamente.

2.5.3 *Frameworks* de Integração

Os *frameworks* de integração são responsáveis por integrar os simuladores de mobilidade e tráfego com os simuladores de rede para a criação de uma VANET. Dentre as opções mais conhecidas é possível citar:

- **Veins [50]:** é um framework robusto e altamente escalável que integra o gerador de mobilidade SUMO com o OMNeT++ através do protocolo TraCI, que permite comunicação em tempo real entre as aplicações veiculares e a mobilidade veicular. Ele implementa toda a pilha de comunicação DSRC/WAVE. Diferente de outras ferramentas ele também replica características reais como a interferência nas transmissões causada por prédios, veículos e outros obstáculos. Também permite implementar modelos de propagação para cálculo da perda do sinal.
- **VSIMRTI [51]:** é um *framework* flexível de integração que pode usar simuladores de rede como o NS3 e OMNeT++ e simuladores de mobilidade como SUMO e VISSIM. Permite simular aplicações ITS por meio de uma arquitetura em camadas.
- **VENTOS [52]:** utiliza o OMNeT++ e o SUMO para comunicações do tipo V2I. Permite roteamento dinâmico do tráfego e comunicação bidirecional através do protocolo SNMP. Também permite o uso de *scripts* em MATLAB para plotar diferentes aspectos do ambiente de simulação.

2.6 Discussão

Neste capítulo, abordaram-se as principais características das redes veiculares como seus modos de operação, podendo ser comunicações V2V, V2I ou ainda V2V2I. Discutiram-se as tecnologias e padrões para VANETs com foco no padrão IEEE 802.11p, DSRC/WAVE. Como este trabalho tem foco em aplicações de prevenção de acidentes, optou-se por priorizar essa tecnologia devido ao seu bom funcionamento em situações críticas com obstáculos físicos ou condições climáticas extremas em comparação ao C-V2X.

As simulações em redes veiculares também foram abordadas, onde foi possível notar a existência de simuladores de mobilidade/tráfego, rede e *frameworks* para integração. Dentre as opções apresentadas, optou-se para o uso da configuração SUMO, OMNeT++ e Veins por ser de código aberto, simples e facilmente editável, além de ser uma das configurações mais populares [53]. Na próxima seção, apresenta-se uma revisão do estado da arte no que tange ao controle e prevenção de congestão em redes veiculares ad-hoc.

Capítulo 3

Trabalhos Relacionados

Este capítulo tem como principal objetivo apresentar uma revisão do estado da arte no que tange aos protocolos e algoritmos propostos nos últimos anos para o controle de congestão em VANETs utilizando a tecnologia proposta pelo IEEE 802.11p e DSRC/WAVE. Primeiramente, apresenta-se a classificação principal das estratégias com base no problema principal que buscam solucionar, sendo eles a congestão causada pelo envio excessivo de *beacons* ou o *broadcast storm* causado pela difusão descoordenada de mensagens de alerta. Cada estratégia será detalhada destacando seus pontos positivos e negativos. No fim do capítulo, uma análise comparativa traz uma discussão sobre os trabalhos analisados.

3.1 Classificação das Estratégias para Controle de Congestão em VANETs

Nesta seção, apresenta-se uma classificação das estratégias para controle de congestão em VANETs como forma de revisão do estado da arte. As propostas são inicialmente classificadas em relação ao seu foco principal, lidar com a congestão causada pelo envio excessivo de *beacons*, Figura 3.1 ou a congestão causada pela difusão descoordenada das mensagens de alerta, Figura 3.2, problema também conhecido como *broadcast storm*.

As estratégias de controle de congestão causada pelos *beacons* vistas na Figura 3.1 podem ser categorizadas com base nos parâmetros ajustados, como a taxa de envio de mensagens e a potência de transmissão [17]. O ajuste de taxa de envio de mensagens [20, 54, 55] têm como principal objetivo diminuir o número de mensagens inseridas na rede para reduzir a ocupação do canal. O ajuste de potência de transmissão [22, 56–59] limita o alcance das mensagens quando uma congestão é detectada, podendo então liberar o canal de transmissão mais rapidamente. Alguns trabalhos utilizam uma abordagem híbrida [2, 21], alterando mais de um parâmetro de transmissão quando a congestão é

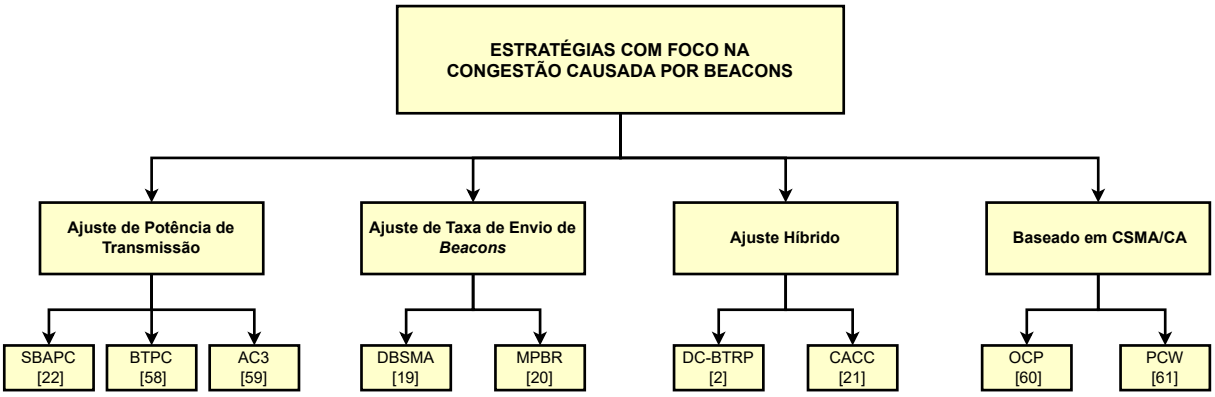

Figura 3.1: Classificação baseada nas estratégias com foco nos *beacons*.

detectada. Outros trabalhos têm foco na camada de enlace e no *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) [60, 61].

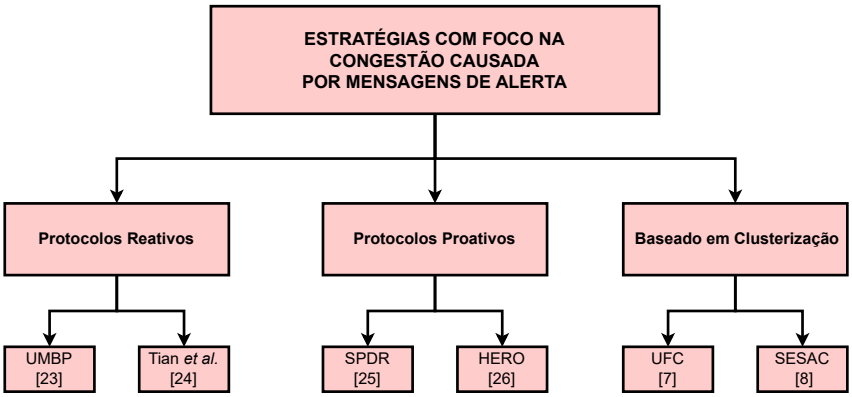

Figura 3.2: Classificação das estratégias de prevenção ou congestão em VANETs (adaptado de [6]).

Os trabalhos com foco no controle da congestão causada pelo envio de mensagens de alerta vistos na Figura 3.2 podem ser categorizados em relação à escolha dos nós encaminhadores durante a difusão [25]. Os protocolos e algoritmos reativos sugerem que a escolha do nó encaminhador deve ser feita localmente por cada veículo ao receber a mensagem de alerta [23, 24]. Na abordagem proativa, os nós encaminhadores são escolhidos com antecedência, geralmente com base em alguma métrica da distância entre os veículos [25, 26]. Além disso, também existem abordagens que propõem a criação de *clusters* para a disseminação das mensagens [7, 8].

Os trabalhos relevantes para este trabalho serão apresentados nas próximas seções conforme as categorias sugeridas nas figuras apresentadas. A ideia geral de cada abordagem é discutida em detalhes e no final de cada seção, uma discussão é feita acerca das estratégias abordadas.

3.2 Estratégias com Foco na Congestão Causada por *Beacons*

As estratégias com foco na congestão causada por *beacons* geralmente adotam algum mecanismo para medir a ocupação do canal de transmissão e tomam medidas para melhorar suas condições quando ele se torna saturado. Algoritmos como ATB [54] e DynB [55] sugerem reduzir a taxa de envio de *beacons*, enquanto D-FPAV [56] e Kloiber *et al.* [57] controlam a congestão por ajustes na potência de transmissão. Nas estratégias baseadas em CSMA/CA [60, 61], alteram-se parâmetros da camada de enlace para controlar a ocupação do canal e como as mensagens são transmitidas. Outras estratégias tomam medidas antes mesmo que uma congestão ocorra, como em Huang *et al.* [62], onde se calcula a probabilidade de enviar um *beacon* com base na predição de posição de um veículo. O algoritmo MPBR [20] segue uma estratégia similar, mas utiliza equações da cinemática com um filtro de Kalman para realizar sua predição.

3.2.1 Ajuste da Potência de Transmissão

O ajuste de potência de transmissão está relacionado a aumentar ou reduzir o número de veículos que receberão uma mensagem. A abordagem mais comum dessas estratégias é diminuir a potência de transmissão quando uma congestão é detectada, de forma que as mensagens enviadas alcancem menos nós e liberem o canal de transmissão mais rapidamente.

- ***Speed Based Adaptive Power Control* (SBAPC) [22]:** A estratégia considera a velocidade do veículo como um fator para atualizar o valor de potência de envio dos BSMs ou *beacons*. Estas mensagens são enviadas com uma potência mínima em uma duração de ciclo (c_{len}) a potência vai sendo aumentada gradativamente considerando a velocidade do veículo até retornar ao seu valor inicial novamente.

  A ideia por trás de utilizar a velocidade para controlar a potência é que veículos se movimentando rapidamente apresentarão um *Time-To-Collision* (TTC) menor, ou seja, estão mais propensos a uma colisão. Isso indica que vizinhos mais distantes devem receber informações desse veículo mais frequentemente. Eles ainda indicam que o valor de c_{len} não pode ser muito grande, para não comprometer a cobertura da vizinhança dos veículos. A potência de transmissão (TX_{power}) é então atualizada seguindo a equação abaixo:

$$TX_{power} = p_{factor} \cdot phase, \tag{3.1}$$

onde p_{factor} é um fator de potência calculado por uma função que considera a velocidade do veículo, a potência máxima e a duração do ciclo c_{len}. O fator *phase* representa quanto tempo se passou dentro do ciclo.

- ***Beacon Transmission Power Control* (BTPC) baseado em Redes Bayesianas [58]:** O algoritmo é dividido em três etapas: 1) Estimativa da carga atual do canal; 2) aprendizado dos parâmetros da rede Bayesiana e 3) ajuste da taxa de potência de transmissão baseado na predição da carga do canal.

  Para medir a carga no canal, eles utilizam o *Channel Busy Ratio* (CBR), que representa uma porcentagem do tempo que o canal foi dito ocupado durante um intervalo de tempo. Eles definem que o valor ideal deve ser igual a $0,6$. Este valor é comparado com uma previsão de CBR feita por um algoritmo chamado KS2A [63] que utiliza redes Bayesianas. Quando o valor está abaixo de $0,6$ a potência não é alterada, já que o canal ainda está livre o suficiente para transmitir mensagens. Quando for maior, a potência P_{next} é alterada seguindo o esquema:

$$P_{next} = \begin{cases} min(P_{current} + \Delta P, P_{max}) & \text{se } S = 1 \\ max(P_{current} - \Delta P, P_{min}) & \text{se } S = 2 \\ P_{min_{init}} & \text{se } S = 3, \end{cases} \tag{3.2}$$

  onde $P_{current}$ representa a potência atual, ΔP representa uma mudança do valor de potência calculada com base no valor de CBR, P_{max} e P_{min} são os valores de máximo e mínimo da potência de transmissão, enquanto $P_{min_{init}}$ é o valor inicial mínimo. S representa o tipo atual de tráfego previsto com ajuda das redes Bayesianas, onde $S = 1$ significa tráfego leve, $S = 2$ congestão leve e $S = 3$ congestão severa.

- ***Adaptive Transmit Power Cooperative Congestion Control* (AC3) [59]:** Propõe uma abordagem baseada em teoria dos jogos com o valor de Shapley, utilizado para distribuir justamente ganhos e perdas para uma coalizão [64]. A ideia principal é permitir que veículos possam alterar seu valor de potência individualmente e medir sua congestão localmente. Com isso, também se introduz a ideia de contribuição de cada nó para a congestão, de forma que um modelo justo para ajuste da potência possa ser implementado. O controle da congestão cooperativo representa o jogo, enquanto os veículos representam os jogadores. Assume-se que eles têm ciência do valor de ocupação do canal CBR de seus vizinhos através da transmissão de *beacons*.

  O algoritmo é dividido em sete etapas, que funcionam resumidamente da seguinte forma: 1) Congestão é detectada na rede por um nó que dá início a um evento

de congestão; 2) veículos que receberam a notificação deste evento ordenam sua lista de vizinhos com base na potência de transmissão; 3) cada veículo entra em uma coalizão para reduzir a potência justamente; 4) o *payoff* para cada coalizão é calculado com base em um modelo do sistema AC3; 5) dada a coalizão e o *payoff*, cada veículo calcula sua contribuição marginal para determinar o nível de redução da potência de transmissão; 6) cada veículo diminui sua potência, marginalmente equivalente à contribuição do veículo; 7) a potência é aumentada gradativamente até que o próximo evento de congestão aconteça.

Discussão

As abordagens com base em ajuste da potência de transmissão geralmente não consideram que as aplicações em VANETs definem um raio de transmissão para seu funcionamento correto. Portanto, reduzir a potência com que os *beacons* são enviados pode realmente reduzir a congestão da rede, mas também leva a perda de informações e cobertura de nós mais distantes. Além disso, estratégias como o SBAPC [22], ainda podem continuar apresentando congestão devido à alta taxa de mensagens sendo enviadas, mesmo que para um número reduzido de nós.

3.2.2 Ajuste de Taxa de Envio de Mensagens

As estratégias com base no ajuste da taxa de envio de mensagens têm como principal objetivo reduzir o número de mensagens inseridas na rede com intenção de diminuir a carga do canal. Na maioria das estratégias que implementam este método, a alteração da taxa está atrelada com o nível de congestão previsto ou medido em um intervalo de tempo definido.

- ***Dynamic Broadcast Storm Mitigation Algorithm* (DBSMA) [19]:** considera apenas a congestão causada por mensagens do tipo *Cooperative Awaraness Message* (CAM), que funcionam de maneira similar aos *beacons*. Considera o tempo de reação e ação do motorista ao se deparar com um acidente como uma forma de definir a taxa de envio de mensagens. Assumindo a existência de um veículo com defeito na pista, deve-se calcular o tempo (T_0) para chegar ao centro da posição onde o veículo defeituoso está estacionado com base em:

$$T_0 = \frac{R_M}{S_i}, \tag{3.3}$$

onde R_M representa o raio de transmissão e S_i a velocidade do veículo se direcionando ao acidente. Depois isso, o cálculo de um multiplicador α é realizado por:

$$\alpha = \left(\frac{CCH_{time}}{T_o}\right) \cdot 100, \tag{3.4}$$

onde o intervalo de tempo do canal de controle é denotado por CCH_{time}. Após isso, calcula-se o tempo de reação T_r:

$$T_r = \frac{D_m}{S_i},$$

onde D_m representa a mínima distância de segurança. E por fim, o tempo de intervalo de transmissão das mensagens:

$$T_B = T_r \cdot \alpha. \tag{3.5}$$

Dessa forma, os autores acreditam que vizinhos se aproximando do veículo defeituoso vão começar a reduzir sua velocidade. Portanto, as mensagens não precisarão ser enviadas com tanta frequência, já que o tempo de reação do motorista não será tão comprometido como quando o veículo estava em alta velocidade.

- ***Mobility Prediction Based Beacon Rate Adaptation* (MPBR) [20]:** o algoritmo de adaptação da taxa de envio de mensagens proposto neste trabalho propõe um mecanismo de predição da posição dos veículos com um filtro de Kalman e equações de cinemática. O filtro de Kalman é utilizado como um método para eliminar ruído de medidas e realizar predições com base nessa medida [65]. Estas predições são feitas para o veículo continuar ciente das posições de seus vizinhos mesmo sem o recebimento de *beacons*.

  Um erro denotado como $\xi(k)$ é calculado em todo momento k através do da distância Euclideana entre a posição da predição realizada no instante anterior com o valor obtido através do filtro de Kalman. Quando o erro $\xi(k)$ for maior que um limiar $\Phi(k)$, então um novo *beacon* deverá ser enviado. Com isso, é possível concluir que quando $\Phi(k)$ for maior, menos *beacons* serão enviados, já que um valor de erro maior também será aceito.

  A estratégia para definir o valor ideal para o limiar $\Phi(k)$ gira em torno de uma classificação das condições atuais de tráfego. Grandezas como Coeficiente de Congestão (CC), Tempo Médio de Estacionamento (PTR), ruído da aceleração (AV) e velocidade média (AV) são utilizados para determinar as condições de tráfego. Estes valores são reunidos em um conjunto de dados classificados em três categorias

(tráfego livre, moderado e limitado). Uma matriz de participação U é inicializada com valores aleatórios entre 0 e 1, depois disso, os centros dos *clusters* c_1, c_2 e c_3 são calculados com base na equação:

$$c_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m}, \tag{3.6}$$

o fator de "fuzzificação" é denotado por m, enquanto u_{ik} é o grau de participação do ponto x_k no cluster c_i, depois os *clusters* são derivados pela minimização da equação:

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \cdot ||c_i - x_k||^2, \tag{3.7}$$

$||c_i - x_k||^2$ representa a distância entre o centro do *cluster* i e o ponto x_k, onde se usa novamente a distância Euclideana. Por fim, os valores da matriz são atualizados com base na equação a seguir:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{||u_i - x_k||)}{||u_j - x_k||} \right)^{\frac{2}{m-1}}}, \tag{3.8}$$

a matriz U é atualizada desta forma até que o valor de J exceda ϵ, que indica que os valores convergiram e os dados sobre a condição de tráfego foram classificados, de forma que seja possível decidir o valor do limiar $\Phi(k)$.

Discussão

As abordagens com base em ajuste da taxa de envio de mensagens geralmente sofrem por perda de informações quando o intervalo entre mensagens não é suficiente para reduzir a congestão da rede. O DBSMA [19] não considera que veículos vizinhos já podem estar reduzindo sua velocidade ou até parados em decorrência de um acidente. Sendo assim, os veículos só começarão a ajustar seus parâmetros quando estiverem próximo do veículo com defeito e não da área impactada pelo acidente. O MPBR [20] apresenta um método de classificação do tráfego com base em clusterização *fuzzy* que necessita uma quantidade de pontos para convergir para um valor específico, ou seja, o algoritmo leva um tempo para poder ser executado, o que não é especificado.

3.2.3 Ajuste Híbrido

As estratégias híbridas buscam unir ajustes de vários parâmetros como a taxa de transmissão, a potência de transmissão ou até a taxa de dados. Assim como nas estratégias anteriores, o ajuste pode ser feito em relação aos *beacons* ou as mensagens de alerta das

aplicações de segurança. Novamente, utiliza-se algum método para prever ou detectar a congestão na rede causada pelo envio de mensagens e em seguida os parâmetros são alteradas para resolver este problema.

- ***Dynamic Control of Beacon Transmission Rate and Power* (DC-BTRP) [2]:** Propõe um algoritmo dividido em duas partes, o primeiro determinará o valor da taxa de transmissões de *beacons*, o segundo a potência de transmissão. O ajuste da taxa de transmissão é feito com base no erro de posição desejado e na velocidade do veículo como um indicativo da condição atual do tráfego. Segundo os autores, em ambientes urbanos altas velocidades geralmente indicam que o tráfego está livre, enquanto velocidades reduzidas indicam tráfego congestionado. O intervalo de transmissão de um *beacon* (I_{b_i}) para um veículo n_i é então calculado da seguinte forma:

$$I_{b_i} = \frac{2(\overline{E} - v_i t_D)}{v_i}, \tag{3.9}$$

onde v_i representa a velocidade do veículo, t_D o atraso da transmissão que é considerado igual para todos os veículos já que se considera que os *beacons* tem o mesmo tamanho (b_z) e transmitidos com a mesma taxa de dados (R_D). $\overline{E}$ representa o erro de posição médio expresso por equações cinemáticas da seguinte forma para um veículo n_i:

$$\overline{E_i} = \frac{v_i t_D + I_{b_i}\left(v_i + \frac{a_i I_{b_i}}{2}\right) + t_D(a_i I_{b_i} + v_i)}{2}, \tag{3.10}$$

onde a_i representa a aceleração do veículo.

O ajuste da potência de transmissão considera o intervalo de transmissão de *beacons* (I_b) e também a carga atual do canal. Para estimar este valor, os autores propõem o cálculo da carga relativa no canal (L_i) considerando o impacto de n_i neste valor e a carga percebida de outros veículos presentes em sua lista de vizinhos. O valor pode então ser calculado por:

$$L_i = \frac{b_z\left(R_{b_i} + \sum_{k=1}^{N_i} R_{b_k} P_{s_k} P_{R_k}\right)}{R_D}, \tag{3.11}$$

onde R_{b_i} representa a taxa de envio de *beacons* do próprio veículo n_i. A equação também considera a influência dos vizinhos na carga do canal de transmissão, onde N_i representa o número de vizinhos de um veículo n_i, tal que $k \neq i$, pois ele já considerou sua carga pessoal em R_{b_i}, similarmente, R_{b_k} representa a taxa de envio de cada vizinho k. P_{s_k} é a probabilidade de uma transmissão com sucesso do k-ésimo veículo na presença de múltiplos transmissores, P_{R_k} é a probabilidade de recepção

correta da mensagem por n_i como função da distância para o veículo n_k. Os valores de P_{R_k} e P_{s_k} são calculados por modelos analíticos encontrados em [66] e [67].

A potência de transmissão (P_{T_i}) é então ajustada pelo veículo n_i antes de toda transmissão de um *beacon* da seguinte forma:

$$P_{T_i} = P_{T_{min}} + (P_{T_{max}} - P_{T_{min}})(1 - \frac{L_i}{L_o})R_{b_i}^{-\beta}, \tag{3.12}$$

onde $P_{T_{min}}$ é a potência de transmissão requerida por n_i para gerar uma mensagem de alerta de alcance mínimo, $P_{T_{max}}$ é a potência de transmissão máxima permitida, L_o é a carga do canal crítica normalizada e β é um fator que controla o impacto da taxa de transmissão de *beacons* na potência.

- ***Channel-Aware Congestion Control Algorithm* (CACC) [21]:** propõe um ajuste híbrido da potência de transmissão e da taxa de dados. Para detectar uma congestão no canal, os autores tomam uma abordagem reativa, ou seja, os parâmetros são ajustados após a congestão ocorrer. Utiliza-se o *Receiver Signal Strength* (RSS) para distinguir a causa da perda de um pacote. O RSS é uma agregação do sinal e da interferência medidos em dBm. A ideia por trás do uso de RSS está atrelada ao fato que pacotes sofrendo colisões terão valores de RSS mais altos do que os de pacotes sofrendo atenuação do sinal para uma dada taxa de dados. Experimentos foram feitos, onde foi possível capturar aproximadamente 90% dos casos de colisão com uma taxa de falso positivo de 2%.

  Durante um intervalo de amostra (T_s), o algoritmo do CACC incrementa um contador do número de pacotes recebidos corretamente (N_s) toda vez que um pacote é decodificado com sucesso. Ainda neste intervalo, o contador do número de pacotes com erro que sofreram colisão (N_c) é incrementado sempre que o valor de RSS do sinal for maior que um valor de corte, caso for menor, incrementa-se o contador de pacotes não decodificados (N_w). Feito isso, o algoritmo calcula a taxa de colisões entre pacotes (PCR) por:

$$PCR = \frac{N_c}{(N_s + N_c)}, \tag{3.13}$$

  quando o valor de PCR for maior que um limiar μ, que indica o nível desejado para a taxa de colisões, a potência de transmissão é decrementada por um valor δ, caso contrário é incrementada também por δ. Após isso, ele calcula a taxa de recepção de pacotes (PDR) por:

$$PDR = \frac{N_s}{(N_s + N_w)}, \tag{3.14}$$

quando o valor de PCR for maior que μ, também se alterá a taxa de dados para 6 Mbps. Se $PCR < \mu$ e o $PDR > \rho$ então a taxa de dados é alterada para 3 Mbps, onde ρ indica o nível desejado de PDR. Conclui-se que quando não existem muitos pacotes sendo perdidos por colisões e a taxa de recepção estiver alta, então o algoritmo permite que o veículo transmita com menor taxa de dados e maior potência.

Discussão

As estratégias híbridas apresentam soluções interessantes para o desenvolvimento de protocolos e algoritmos para prevenção da congestão em VANETs. O algoritmo DC-BTRP [2] é uma proposta diferente das outras analisadas porque também considera o erro de acurácia de posicionamento, além de realizar ajuste da taxa de transmissão e potência. Entretanto, os autores consideram a velocidade como um fator para determinar o intervalo de transmissão de *beacons*, o que nem sempre pode representar uma congestão real na pista, assim como veículos podem estar se movimentando rapidamente em estradas com alta densidade de veículos na vizinhança.

Para o algoritmo CACC [21], como apenas a taxa de dados e a potência de transmissão são atualizadas, ainda podem existir cenários onde alterar parâmetros não será suficiente para controlar a congestão causada pelas mensagens. Além de que o método de ajuste de potência é efetuado de forma simplória considerando apenas a adição e redução de um valor fixo, não considerando os requisitos de acurácia de posicionamento das aplicações de segurança dos usuários.

3.2.4 Estratégias Baseadas em CSMA/CA

As estratégias baseadas em acesso múltiplo com verificação de portadora com prevenção de colisão, ou simplesmente CSMA/CA, procuram alterar os parâmetros de acesso ao canal na camada de enlace para diminuir a congestão causada pelo envio de mensagens. Geralmente, altera-se a janela de contenção para utilizar a largura de banda disponível eficientemente. Estes algoritmos ou protocolos podem ser divididos em três categorias conforme o meio de acesso à camada MAC, podendo ser baseados em contenção, sem contenção e híbridos [6].

- ***Optimised CSMA/CA protocol* (OCP) [60]:** indica que o valor da janela de contenção pode ser optimizado para melhorar o desempenho do canal de transmissão com base na densidade veicular. Utiliza um modelo estocástico para obter o valor máximo de CW, que representa a janela de contenção. Depois integra este

mecanismo com um protocolo de CSMA/CA para maximizar a taxa de recepção de pacotes entre veículos adjacentes em uma distância de um salto.

O primeiro passo da proposta é estabelecer uma relação entre o valor máximo de CW e a probabilidade de transmissão, definida por um modelo com base na densidade. Os autores consideram que CAMs (mensagens semelhantes aos *beacons*) são enviados em uma taxa fixa de 10 Hz, ou seja, a cada 100 ms, considerando o intervalo de CCHI. Cada pacote recebe um tempo de recuo igual ao tamanho da janela de contenção CW, esse valor é decrementado toda vez que o canal é considerado livre e congelado quando o canal está ocupado. Ao chegar em zero, o pacote é transmitido. Com isso, define-se o valor máximo da janela de contenção como:

$$W = \left\lfloor \frac{2}{b_0(\lambda)} - 1 \right\rfloor, \tag{3.15}$$

onde W é o tamanho máximo da janela de contenção $CW = [0, W-1]$ e b_0 a probabilidade que o contador do tempo de recuo é zero, ou seja, que um pacote será transmitido. Enquanto λ representa uma densidade de veículos percebida pelo veículo.

- ***Predictive Contention Window* (PCW) [61]:** propõe um mecanismo de ajuste da janela de contenção para evitar a competição pelo canal causada pelo envio excessivo de *beacons*. A estratégia de ajuste é baseada em uma classificação de estados dos veículos com base em vários atributos relacionados com a densidade de tráfego. A janela de contenção é comparada com cada conjunto de atributos através do algoritmo de aprendizado *Bayesian Personalized Ranking* (BPR) [68]. Após isso, uma predição sobre os próximos estados dos veículos é realizada por um Modelo Oculto de Markov (do Inglês, *Hidden Markov Model*) (HMM). Essa estimativa é feita para melhorar o resultado em tempo real do mecanismo, que por fim resulta em um novo tamanho da janela de contenção.

  O primeiro passo do PCW é definir os conjuntos de atributos que serão utilizados para classificar o tráfego. Ele considera o número de veículos vizinhos, velocidade e tempo de parada. Ele utiliza uma abordagem conhecida como *fuzzy nearness* para obter o grau de similaridade dos estados do veículo objeto e seus conjuntos de atributos. A janela de contenção é inicialmente calculada através da equação:

$$CW_i = \lfloor \alpha CW_{1i} + \beta CW_{2i} + \gamma CW_{3i} \rfloor, \tag{3.16}$$

  onde CW_{1i}, CW_{2i} e CW_{3i} representam o tamanho da janela de contenção para cada um dos atributos considerados (número de vizinhos, velocidade e tempo de parada),

enquanto α, β e γ são parâmetros utilizados para controlar quanto cada parâmetro deve inferir no cálculo.

Como as características do tráfego podem mudar em diferentes cenários, os valores de α, β e γ são determinados por um mecanismo de aprendizado que utiliza o algoritmo BPR. Este algoritmo é geralmente utilizado para recomendação de itens, mas nesse trabalho é proposto como um método de aprendizado genérico para resolver um problema de optimização. Eles utilizam o *Stochastic Gradient Descent* (SGD), um algoritmo onde se calcula o gradiente de todas as instâncias de um conjunto de treinamento para depois se ajustar os parâmetros, considerando o erro calculado com um algoritmo de *backpropagation* [69].

Discussão

As estratégias baseadas em CSMA/CA buscam atualizar os parâmetros de acesso ao canal como a janela de contenção para maximizar o uso do canal de transmissão. Em cenários com menor densidade de veículos, o OCP [60] e o PCW [61] demonstram bons resultados, pois evitam o desperdício da largura de banda do canal. Entretanto, em cenários com alta densidade de veículos, as estratégias de CSMA/CA não são suficientes para prevenir a congestão, pois não diminuem o número de mensagens sendo inseridas na rede, como visto em outras estratégias.

3.3 Estratégias com Foco na Congestão causada por Mensagens de Alerta

As estratégias com foco na congestão causada por mensagens de alerta têm como principal objetivo propor novas formas de rotear as mensagens em VANETs. Como mencionado na Seção 4.1.3, o *broadcast storm* é um problema comum causado pela difusão descoordenada das mensagens de alerta em uma rede. Uma das formas de resolver este problema é limitar o número de nós que vão encaminhar a mensagem. Os protocolos reativos [24,61], sugerem escolher um nó encaminhador com base em algumas condições quando a mensagem é recebida. Enquanto, os protocolos proativos [25, 26] definem métricas para a escolha de um ou mais nós encaminhadores para a transmissão das mensagens. Outras estratégias utilizam a técnica de clusterização para definir grupos controlados por um nó chamado *Cluster Head* (CH), que geralmente é responsável por transmitir as mensagens de alerta entre os membros do *cluster* [7, 8].

3.3.1 Estratégias Reativas

Nas estratégias reativas, a escolha de um nó encaminhador é feita localmente por cada veículo após receber uma mensagem de alerta. Geralmente, os veículos definem tempos de espera para encaminhar a mensagem com base em alguma métrica, como a distância e direção do encaminhamento.

- ***Urban Multi-hop Broadcast Protocol* (UMBP) [23]**: O protocolo UMBP sugere que os veículos mais distantes do veículo de origem tenham preferência ao acessar o canal de transmissão. Ele utiliza o mecanismo *black-burst* para conduzir a seleção dos nós encaminhadores em cada direção. Esse mecanismo é utilizado para fornecer atrasos de acesso garantidos ao tráfego de pacotes onde existe uma taxa limitada [70]. Quando um veículo recebe uma mensagem de alerta, o processo iterativo para seleção dos nós candidatos se inicia após um intervalo SIFS.

  Neste artigo os autores sugerem o uso de um *mini-slot*, com uma duração de $\tau = 2\delta + t_{switch}$, onde δ é o atraso máximo da propagação de sinal no raio de transmissão R e t_{switch} é o tempo que o rádio leva para mudar entre o modo de transmissão e recepção. Além disso, eles sugerem realizar a escolha de um encaminhador em ambas as direções para a difusão bi-direcional.

  A escolha do nó encaminhador na direção frontal é feita considerando iterações dos *mini-slots*. Na primeira iteração que dura dois *mini-slots*, o raio de transmissão R é divido para os veículos na *Far Area* (FA) e na *Near Area* (NA), os vizinhos que estiverem na FA enviam um *black-burst*, enquanto os vizinhos em NA escutam. Essa zona será particionada sempre que os veículos que estiveram ali não receberem o sinal de *black-burst* durante o primeiro *mini-slot*. Caso contrário, a região não será particionada. No terceiro mini-slot os veículos na FA enviam sinais de *black-burst*, enquanto os veículos em NA apenas recebem. As iterações continuam por:

  $$N \leq \left\lfloor \frac{T_{DIFS} - T_{SIFS} - \tau}{2\tau} \right\rfloor, \tag{3.17}$$

  onde na enésima iteração N que dura três *mini-slots*, os vizinhos em FA enviam um sinal *black-burst* no *mini-slot* $2N - 1$ e se tornam o encaminhador com sucesso da direção frontal.

  O encaminhador entre os vizinhos que estão atrás do veículo de origem utilizam um processo diferenciado para a seleção do encaminhador. Diferente da versão frontal, os veículos utilizam o segundo *mini-slot* para transmissão ou recepção do sinal de *black-burst*. Após N iterações, um nó candidato para o encaminhamento seleciona um *mini-slot* na janela de contenção CW e começa o processo de recuo ou *backoff*

onde:

$$CW = \left\lfloor \frac{T_{DIFS} - T_{SIFS}}{\tau} \right\rfloor, \tag{3.18}$$

quando o tempo se encerra, os veículos podem repassar a mensagem dentro daquele *mini-slot*.

- **Tian *et al.* [24]**: Propõe um protocolo para a disseminação de mensagens de alerta em apenas uma região de interesse, onde os veículos que se interessarem pela mensagem pode recebê-la e encaminhá-la independentemente. O primeiro passo do protocolo é classificar as mensagens de alerta em três níveis, o *one-hop broadcast*, *forward broadcast* e *backward multi-hops broadcast*. Para cada tipo de mensagem, uma região de interesse será calculada de acordo. O *one-hop broadcast* por exemplo é voltado para aplicações como mudança de pista e freio de emergência. Enquanto o *forward broadcast* ou difusão para frente, está relacionada com aplicações como aviso de um veículo fora de controle ou um caminhão de bombeiros. O *backward multi-hops broadcast* se relaciona com avisos de acidentes na pista e deslizamentos de terra.

  Ao receber uma mensagem de alerta, cada veículo checará o seu tipo e decidirá se ele deve se tornar o nó encaminhador. Se a mensagem for do tipo *one-hop broadcast*, isso significa que o veículo não precisa encaminhá-la. Em outros casos, a mensagem deverá ser repassada. Um veículo terá a oportunidade de repassar a mensagem quando sua distância entre si mesmo e o último encaminhador r for maior que zero e menor que R, ou o máximo alcance de transmissão. Se ele estiver na direção certa para propagar a mensagem, ele inicia um tempo de espera WT calculado por:

$$WT = -\frac{dist}{R} WT_o + m(1 + a^{-flag}) WT_o, \tag{3.19}$$

  onde $dist$ é a distância entre um veículo i e k dentro da região de interesse, a e m são constantes usadas para distinguir o tempo de espera dos veículos em direções diferentes, WT_o também é uma constante, $flag$ também é utilizada para determinar a direção do encaminhamento. Quando o tempo de espera acaba, a mensagem poderá ser encaminhada na região de interesse. Para evitar que mais mensagens sejam inseridas na rede, outros veículos cancelam seu encaminhamento, se ouvirem uma retransmissão.

Discussão

As estratégias reativas analisadas não utilizam o envio de *beacons* para realizar a transmissão das mensagens de alerta. Essa abordagem é possível, pois cada nó pode escolher se

deve se tornar um encaminhador ou não. Os autores do protocolo UMBP [23] ressaltam que há a possibilidade de mais de um veículo escolher o mesmo *mini-slot*, o que levaria a colisões, atrapalhando no repasse da mensagem. No protocolo proposto por Tian *et al.* [24], as colisões que podem ocorrer quando o tempo de espera de mais de um veículo se encerra não são consideradas. Além disso, a atenuação do sinal pode fazer com que o veículo mais distante não receba a mensagem, aumentando o atraso de entrega.

3.3.2 Estratégias Proativas

As estratégias proativas visam realizar a escolha do nó encaminhador da mensagem de alerta antes mesmo dela ser recebida pelos veículos vizinhos. Com isso, espera-se reduzir o número de mensagens inseridas na rede, que poderia acontecer em estratégias reativas quando mais de um veículo tenta repassar a mensagem. Além disso, as estratégias proativas podem diminuir o atraso de entrega, visto que os encaminhadores são escolhidos previamente.

- ***Heuristic Routing for Vehicular Networks* (HERO)** [26]: O HERO é um protocolo de roteamento que seleciona segmentos de pista e veículos encaminhadores com o uso de funções heurísticas. A seleção de segmentos de pista é feita com base em duas funções, a primeira calcula a distância mínima utilizando o produto escalar do ângulo, a distância perpendicular e o comprimento do segmento. A segunda função heurística considera a conectividade do seguimento considerando o alcance de transmissão, número de faixas, comprimento do segmento e a contagem média de veículos. Com isso, o protocolo tenta encontrar a melhor rota para rotear uma mensagem de alerta.

  A escolha do veículo encaminhador é feita conforme a direção de movimento do veículo, a diferença de velocidade, perda de sinal e o tamanho do *buffer*. Com esses valores, ele então calcula a prioridade de cada veículo por:

$$\tilde{Q}_{i,j} = \frac{\tilde{D}_{i,j}(\tilde{S}_{i,j} + \tilde{F}_{i,j} + \tilde{M}_{i,j})}{\sum_{\forall n_x \in \mathbb{N}_i} \tilde{D}_{i,x}(\tilde{S}_{i,x} + \tilde{F}_{i,x} + \tilde{M}_{i,x})} \forall n_j \in \mathbb{N}_i, \tag{3.20}$$

  onde $\tilde{S}_{i,j}$ representa a distribuição de diferença de velocidade, $\tilde{D}_{i,j}$ a distribuição da direção do veículo, $\tilde{M}_{i,j}$ a distribuição do tamanho disponível do *buffer*, $\tilde{F}_{i,j}$ a distribuição da perda de sinal. O conjunto de veículos vizinhos de um veículo n_i é representado por $\mathbb{N}_i$, sendo que n_j representa um vizinho nesta lista. Após o cálculo ser realizado, o veículo de origem escolhe o vizinho com o maior valor de $\tilde{Q_{i,j}}$ para ser o encaminhador da mensagem.

- ***Speed and Position aware Dynamic Routing* (SPDR)** [25]: O SPDR é um protocolo de roteamento que utiliza um roteamento dinâmico e guloso, com uma estratégia de encaminhamento colaborativa. Ele propõe dividir os veículos vizinhos dentro de sua *Routing Decision Area* (RDA) como *Forward Neighbors* (FNs) e *Backward Neighbors* (BNs). A decisão do tamanho da RDA considera a direção dos veículos e a zona de destino que a mensagem de alerta deve chegar. Após isso, o protocolo calcula qual veículo deve rotear a mensagem no momento t por:

$$b_{opt}(t) = arg \max_{\{b \in N_{RDA}\}} [D(a, d, t) - D(b, d, t)], \tag{3.21}$$

onde $D(a, d, t)$ representa uma função da distância entre o transmissor a e o destino d no tempo t, já $D(b, d, t)$ representa a distância entre o próximo encaminhador b e o destino d no momento t. Com isso, o protocolo conseguirá escolher o seu veículo vizinho mais próximo do destino.

Discussão

As estratégias proativas apresentam uma solução para o problema de *broadcast storm*, pois evitam a disseminação de mensagens por vários nós simultaneamente. Entretanto, os protocolos HERO e SPDR utilizam do envio de *beacons*, mas não consideram o impacto que a congestão causada por eles pode ocasionar na rede. No caso do SPDR, os *beacons* são enviados em longos intervalos de tempo, o que reduzirá a congestão, mas as informações presentes na lista de vizinhos dos veículos estarão desatualizadas, o que pode prejudicar na seleção dos nós encaminhadores. Mostrando também, que essa estratégia não atenderia aos requisitos das aplicações de segurança, como informado na Seção 4.1.2.

3.3.3 Clusterização

As estratégias baseadas em clusterização visam aumentar a confiabilidade e escalabilidade da rede. Os veículos são organizados em grupos ou *clusters* que podem transmitir mensagens de forma distribuída entre eles. Geralmente um veículo é denominado como *Cluster Head* (CH), e é responsável por coordenar as comunicações dos membros presentes dentro do *cluster*. A coordenação dos

- ***Unified Framework of Clustering* (UFC) [7]:** Propõe uma estratégia baseada em clusterização como forma para resolver o problema da escalabilidade em redes veiculares chamada de UFC. O *framework* proposto é dividido em três etapas: 1) Amostragem de vizinhos; 2) seleção de cabeça do *cluster* baseada em tempo de

recuo; e 3) manutenção do *cluster*. No UFC, os veículos operam em um dos quatro estados mostrados na Figura 3.3.

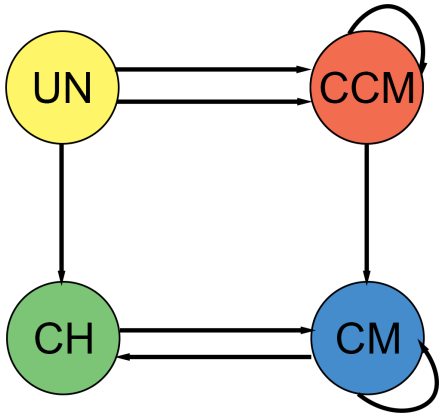

Figura 3.3: Transição dos estados do *framework* UFC (adaptado de [7]).

O estado UN (*Undecided Node*) representa o estado inicial dos nós, onde o veículo não pertence a nenhum *cluster*, CCM (*Candidate Cluster Member*) representa o veículo que entrará em um *cluster*, mas ainda não recebeu uma mensagem de confirmação. CM (*Cluster Member*) representa um veículo que pode estar ligado a um CH (*Cluster Head*) que representa o líder do *cluster*, que deverá se comunicar com os outros membros do *cluster*.

Na fase de amostragem dos vizinhos, o veículo analisa os *beacons* recebidos recentemente e filtra os vizinhos mais ideais em um conjunto chamado SN com base nas condições de mobilidade (direção e velocidade semelhantes). A seleção do *cluster* pode ser feita utilizando métricas como o tempo médio de vida da conexão denotado como $\overline{LLT_i}$, distância média relativa $\overline{\Delta D_i}$ e velocidade média relativa $\overline{\Delta v_i}$.

A outra forma de seleção do *cluster* é feita com uso de tempo de recuo, que determina um tempo máximo para o veículo esperar receber uma mensagem de anúncio de um CH, denotada como CHA, caso o tempo for ultrapassado e o veículo não tiver recebido nenhuma mensagem, ele pode se tornar o CH (*Cluster Head*).

Se a capacidade de um *cluster* não tiver sido excedida, o CH envia uma mensagem $ACKJoin$ para os vizinhos que enviaram uma requisição de entrada $ReqJoin$, se o veículo não receber a confirmação então ele parte para um mecanismo de seleção de *cluster* de *backup*. O veículo cria uma lista com seus nós com uma lista ordenada das conexões com maior estabilidade e envia mensagens $ReqJoin$ e inicia um novo tempo de recuo.

A manutenção do *cluster* feita por um CH considera o tempo desde que ele recebeu um *beacon* de um membro, se este tempo for excedido ele remove aquele nó do seu *cluster*. O mesmo é feito para um nó que não recebeu um *beacon* ou mensagem de seu CH em um determinado tempo, onde ele assume que perdeu conexão com aquele *cluster*.

- ***SDN-Enabled Social-Aware Clustering* (SESAC) [8]:** Propõe uma estratégia baseada em redes SDNs e clusterização. Primeiramente, utiliza um modelo para prever o padrão de mobilidade de um veículo, que consiste em sua rota futura e o tempo de permanência correspondente em cada segmento de estrada. Baseado neste modelo de predição, o algoritmo SESAC utiliza a visão global de um controlador SDN para seleção de um cabeça do *cluster* denotado como CH que fica responsável por transmitir mensagens entre os membros do *cluster*.

  A movimentação de um veículo é descrita por um modelo semi-Markov homogêneo de tempo discreto. Um modelo semi-Markov é uma extensão do modelo Markov que prevê a direção de transição do estado futuro conforme o estado atual e também a um estado de período de transição [8]. Nesse trabalho, a entrada é considerada o histórico de informações, incluindo a probabilidade de transição de estado e a distribuição de probabilidade de tempo de permanência. A equação resumida para o estado de mobilidade de um veículo é denotada por:

$$G_{ij}^{m}(t) = P_{ij}^{m} \cdot W_{ij}^{m}(t), \tag{3.22}$$

  onde $G_{ij}^{m}(t)$ é a probabilidade que a transição de estado de um veículo v_m do segmento de pista r_i para r_j seja completo no período de 0 a t. P_{ij}^{m} representa a probabilidade do veículo mudar de segmento de pista e $W_{ij}^{m}(t)$ a distribuição de probabilidade do tempo de permanência para o veículo v_m.

  As redes SDNs são utilizadas segundo a Figura 3.4. No plano de aplicação, funções de rede para diferentes cenários são implementadas em módulos relacionados a três aspectos, segurança, eficiência e entretenimento. A interface *Northbound* é utilizada para a comunicação entre o plano de aplicação e o de controle com uma visão global da rede e é responsável por monitorar e prever a localização de veículos com base nas informações históricas. Depois as regras de fluxo são distribuídas para as *Base Stations* (BS) através da interface *Southbond* por conexões de fibra e depois para os veículos por comunicações celulares utilizando 5G. Por fim, o plano de dados permite que os veículos se comuniquem entre si por comunicações V2V após o *cluster* ter sido informado aos veículos pela conexão V2I com as BSs.

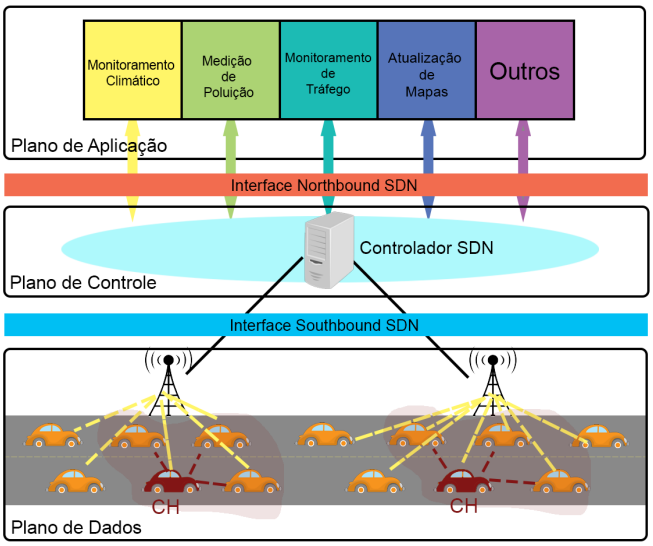

Figura 3.4: Arquitetura das VANETs baseadas em SDNs (adaptado de [8]).

Discussão

As estratégias baseadas em clusterização apresentam uma solução eficiente para o problema de congestão e de escalabilidade em VANETs. Entretanto, *frameworks* como o UFC [7] adicionam uma sobrecarga adicional com as ações de criação, entrada, saída e manutenção do *cluster*, além da dependência ao CH (*Cluster Head*) que em caso de falha comprometerá todos os membros do *cluster*.

As estratégias baseadas em SDNs oferecem uma forma promissora para o funcionamento das VANETs. Entretanto, podem apresentar altos custos de manutenção e gerenciamento ao se considerar a arquitetura mostrada na Figura 3.4 do SESAC [8], onde há a necessidade de existência de infraestruturas a beira da pista conectadas a um servidor por cabos de fibra óptica.

3.4 Análise Comparativa

Nessa seção, apresenta-se um sumário e uma análise comparativa das estratégias para controle de congestão em VANETs analisadas. A Tabela 3.1 traz um resumo indicando o nome, ano, foco do controle de congestão, categoria e os simuladores utilizados por cada trabalho. No total foram analisados 15 trabalhos, onde 9 deles tiveram como foco principal o controle da congestão causada por *beacons* e os outros 6 a congestão causada pela difusão de mensagens de alerta.

Tabela 3.1: Sumário dos trabalhos analisados para controle de congestão em VANETs.

Nome	**Ano**	**Foco do Controle de Congestão**	Categoria	**Simuladores**
SBAPC [22]	2018	*Beacons*	Ajuste de Potência de Transmissão	SUMO/OMNeT++/Veins
BTPC [58]	2020	*Beacons*	Ajuste de Potência de Transmissão	SUMO/NS-3
AC3 [59]	2018	*Beacons*	Ajuste de Potência de Transmissão	SUMO/OMNeT++/Veins
DBSMA [19]	2020	*Beacons*	Ajuste de Taxa de Envio de Mensagens	MATLAB
MPBR [20]	2018	*Beacons*	Ajuste de Taxa de Envio de Mensagens	SUMO/NS-3
DC-BTRP [2]	2018	*Beacons*	Ajuste Híbrido	SUMO/OMNeT++/Veins
CACC [21]	2020	*Beacons*	Ajuste Híbrido	SUMO/OMNeT++/Veins
OCP [60]	2017	*Beacons*	Baseado em CSMA/CA	Não informado.
PCW [61]	2016	*Beacons*	Baseado em CSMA/CA	Não informado.
UMBP [23]	2016	Mensagens de Alerta	Protocolo Reativo	NS-2
Tian *et al.* [24]	2018	Mensagens de Alerta	Protocolo Reativo	Não informado
SPDR [25]	2021	Mensagens de Alerta	Protocolo Proativo	VanetMobiSim/NS-2
HERO [26]	2022	Mensagens de Alerta	Protocolo Proativo	VSIM
UFC [7]	2018	Mensagens de Alerta	Baseado em Clusterização	SUMO/NS-2
SESAC [8]	2018	Mensagens de Alerta	Baseado em Clusterização	Não informado
APGP	2023	*Beacons* e Mensagens de Alerta	Ajuste Híbrido	SUMO/OMNeT++/Veins

Em relação ao uso dos simuladores, pode-se notar uma preferência para a configuração SUMO, OMNeT++ e Veins nos trabalhos CACC [21], SBAPC [22], AC3 [59] e DC-BTRP [2]. O uso do SUMO com os simuladores de rede NS-2 ou NS-3 foi escolhido em trabalhos como BTPC [58], MPBR [20] e UFC [7], no caso do SPDR [25], optou-se pelo VanetMobiSim com o NS-2. Apenas o DBSMA [19] utilizou o MATLAB para as simulações, enquanto HERO [26] foi o único a utilizar o VSIM. Os trabalhos PCW [61], OCP [60], Tian *et al.* [24] e SESAC [8] não informaram a configuração utilizada, prejudicando a reprodutibilidade dos experimentos. A proposta abordada nesse trabalho (APGP), lida com *beacons* e mensagens de alerta através de um ajuste híbrido. A configuração de simuladores utilizada foi o SUMO, OMNeT++ e Veins.

3.4.1 Critérios para a Análise Comparativa

Antes de comparar os resultados de cada trabalho, é necessário destacar quais são as características de uma estratégia eficiente para garantir que os requisitos das aplicações de prevenção de acidentes sejam atendidos. Para isso, o protocolo ou algoritmo deve controlar a congestão causada pelo envio de *beacons* e mensagens de alerta, enquanto também garante a acurácia de posicionamento. As métricas utilizadas para analisar os resultados dos trabalhos estão representadas na Tabela 3.2.

A análise comparativa dos fatores apresentados é feita em duas etapas, primeiramente se consideram as métricas que influenciam na congestão do canal como a taxa de recepção de pacotes e a ocupação do canal; posteriormente, consideram-se as métricas que influenciam nos requisitos das aplicações de segurança como o atraso de entrega e acurácia de posicionamento.

Os trabalhos devem apresentar alguma medida da taxa de recepção de pacotes para demonstrar que os *beacons* e mensagens de alerta estão sendo transmitidos corretamente.

Tabela 3.2: Métricas abordadas ou discutidas em cada trabalho analisado.

Nome	Atraso	Taxa de Recepção de Pacotes	Ocupação do Canal	Acurácia de Posicionamento	Colisões
SBAPC [22]	☑	☐	☑	☐	☐
BTPC [58]	☑	☑	☑	☐	☐
AC3 [59]	☐	☑	☑	☐	☐
DBSMA [19]	☐	☑	☑	☐	☐
MPBR [20]	☐	☑	☐	☑	☐
DC-BTRP [2]	☐	☑	☐	☑	☑
CACC [21]	☐	☑	☑	☐	☑
OCP [60]	☑	☑	☐	☐	☐
PCW [61]	☑	☐	☐	☐	☑
UMBP [23]	☑	☑	☐	☐	☐
Tian *et al.* [24]	☑	☑	☐	☐	☐
SPDR [25]	☑	☑	☐	☐	☐
HERO [26]	☑	☑	☐	☐	☐
UFC [7]	☑	☑	☐	☐	☐
SESAC [8]	☑	☐	☐	☐	☐
APGP	☑	☑	☑	☑	☐

Resultados como a taxa de pacotes perdidos, seja por colisões ou erros, não são necessários, visto que podem ser inferidos da taxa de recepção. Entretanto, eles podem ser ideais para identificar os pontos negativos de uma proposta.

A taxa de ocupação do canal deve ser mantida a um nível próximo a 60% mesmo em cenários propensos a saturação do canal, como discutido na Seção 4.1.1. O atraso para a entrega das mensagens de alerta deve ser o menor possível considerando as condições do tráfego e de roteamento das mensagens. A acurácia de posicionamento dos veículos deve corresponder com os requisitos estabelecidos pelas aplicações de segurança, definidos com mais detalhes na Seção 4.1.2.

3.4.2 Análise das Métricas Relacionadas ao Controle da Congestão

As métricas relacionadas ao controle da congestão são as que influenciam diretamente no canal de transmissão. Em aplicações para prevenção de acidentes, a perda de um pacote pode ter sérias consequências. Garantir uma taxa alta recepção é o objetivo principal de grande parte dos trabalhos analisados [2, 19, 20, 24–26, 58, 59]. A ocupação do canal é um indicativo das condições do canal de transmissão e pode mostrar se a largura de banda disponível está sendo utilizada corretamente. O APGP, proposta desse trabalho que será apresentada no capítulo seguinte, lida com todas as métricas, exceto as colisões, por serem inferidas da taxa de recepção de pacotes. Seus resultados serão discutidos com mais detalhes no Capítulo 5.

Taxa de Recepção de Pacotes, Colisões

A taxa de recepção de pacotes, também conhecida como *Packet Delivery Ratio* (PDR) é um indicativo que demonstra se as mensagens estão sendo transmitidas corretamente. Em relação aos protocolos e algoritmos com foco na congestão causada por *beacons*, notou-se no BTPC [58] um valor de PDR de aproximadamente 92% em um cenário com 100 veículos, enquanto o AC3 [59] fica abaixo de 60% para apenas 50 veículos. Em um cenário com 200 veículos, o MPBR [20] alcança aproximadamente 95%, porém também diminui a acurácia da posição consideravelmente.

Para os protocolos voltados a congestão causada pelas mensagens de alerta, foi possível observar no UMBP [23] um valor de PDR acima de 95%, mesmo no cenário com maior densidade de veículos. O protocolo proposto por Tian *et al.* [24] apresenta uma recepção de aproximadamente 96%, que fica acima do UMBP, mas abaixo de outras propostas comparadas no trabalho proposto pelos autores. A justificativa dada é que nessas outras propostas, veículos fora da região de interesse também receberam as mensagens de alerta.

O SPDR [25] apresenta PDR abaixo de 70% em um cenário com menor densidade de veículos, mas gradualmente aumenta para aproximadamente 80% quanto mais veículos estão na simulação. Esse comportamento é explicado pelos autores pelo fato do número de veículos para repassar as mensagens aumentarem. Entretanto, eles não consideram que essa perda de pacotes pode estar sendo causada pelo envio de *beacons*, que também faz parte da estratégia. O HERO [26] também faz o envio periódico de *beacons* e não considera a congestão causada por eles, mas o PDR ainda se mantêm acima de 90% nos cenários apresentados.

As colisões entre pacotes também são abordadas brevemente em trabalhos como o CACC [21] e PCW [61]. No PCW foi possível observar uma taxa de colisões acima de 50%, enquanto no CACC este valor não ultrapassou 10%.

Ocupação do Canal

A ocupação do canal ou *Channel Busy Ratio* (CBR) indica o tempo em que o canal foi percebido como ocupado pelos nós de uma rede. Em um cenário com 100 veículos, o BTPC [58] conseguiu manter o valor de CBR em aproximadamente 60%, no AC3 [59] os autores compararam diferentes valores de uma constante para medir o CBR sendo o maior valor 45% e o menor próximo a 30%. Para o CACC [21], o valor de CBR se aproxima de 60% para diferentes parâmetros analisados em um cenário que considera uma topologia com 800 veículos gerados, entretanto, não se informa quantos estão presentes na simulação simultaneamente.

3.4.3 Análise das Métricas Relacionadas aos Requisitos das Aplicações de Segurança

As métricas relacionadas aos requisitos das aplicações de segurança envolvem o atraso e a acurácia da posição. Em ambientes veiculares, a entrega de mensagens é um fator sensível ao tempo, já que as informações de posicionamento podem se tornar obsoletas rapidamente. O atraso é considerado em vários trabalhos [22, 58, 60, 61], entretanto não é mencionado se a acurácia de posicionamento é mantida aos níveis desejados pela aplicação. Apenas os algoritmos MPBR [20] e DC-BTRP [2] tratam da acurácia de posicionamento dos seus vizinhos.

Atraso de Entrega

O atraso de entrega se refere ao tempo que uma mensagem leva até chegar ao seu destinatário. O SBAPC [22] mede em torno de 50 ms para o *Inter-Packet Delay* (IPD), que indica o tempo médio do recebimento de pacotes sucessivos de um mesmo vizinho. O BTPC [58] aproximadamente 35 ms. Em um cenário semelhante observado no PCW [61], o atraso chega em aproximadamente 320 ms, enquanto no OCP [60] a mesma métrica fica em aproximadamente 100 ms para apenas 45 veículos, mostrando a ineficácia de estratégias baseadas em CSMA/CA conforme a densidade aumenta.

O protocolo UMBP [23], apresenta um atraso de aproximadamente 3 ms para a entrega de mensagens de alerta a um salto de distância nos cenários mais densos. No pior caso, onde a transmissão deve ser feita para ambas as direções, o protocolo fica acima de 5 ms e tem desempenho pior do que outra estratégia analisada. O protocolo proposto por Tian *et al.* [24] apresenta o atraso medido por diferentes receptores, o que não necessariamente indica o desempenho geral da rede. Os resultados mostram que a estratégia proposta demora aproximadamente 27 ms para transmitir uma mensagem.

O protocolo SPDR [25] apresenta um atraso médio de aproximadamente 30 ms para realizar a transmissão de uma mensagem em um cenário com uma densidade de 150 veículos. Esse resultado melhora quanto mais veículos estão presentes na simulação, devido à capacidade de melhoria na escolha de um nó encaminhador. O protocolo HERO [26] considera o atraso total que um veículo leva para transmitir uma mensagem até um veículo de destino. Em um cenário com 200 veículos, o protocolo manteve o atraso médio em aproximadamente 5 segundos. Assim como no SPDR [25], esse valor diminui para aproximadamente 4 segundos em cenários com mais veículos.

Acurácia de Posicionamento

A acurácia de posicionamento se refere principalmente ao envio de *beacons* e o quão atual as informações deles são. Quanto mais *beacons* são enviados, mais acurada a informação da posição será. O MPBR [20] apresenta um gráfico com o número de *beacons* e acurácia da posição, onde para 6000 *beacons*, o erro médio da posição real do veículo se aproxima de 2.2 m, para 13000 *beacons* este número cai para 0.6 m. No DC-BTRP [2], uma contagem é conduzida toda vez que o erro fica acima de 1 m, considerada pelos autores como perigoso, onde o algoritmo consegue se sobressair em relação a outras propostas em 50%.

Clusterização

As estratégias baseadas em *clusters* como o UFC [7] e o SESAC [8] apresentam métricas relacionadas à eficiência do *cluster* como sua duração, número de *clusters*, participação dos veículos nos *clusters*, taxa de desconexão, atraso para entrar em um *cluster*. No UFC, os autores consideram um cenário de 200 veículos onde foi possível notar uma duração de *cluster* de em média 180 s, enquanto a desconexão aconteceu ao *cluster* em média em 3 vezes.

3.5 Discussão

Este capítulo apresentou uma breve revisão do estado da arte sobre os trabalhos para controle da congestão em VANETs. Dentre as propostas com foco no controle da congestão de *beacons*, analisaram-se estratégias com ajuste de parâmetros como potência de transmissão, taxa de envio, ou ajuste híbrido, bem quanto as estratégias baseadas em CSMA/CA. As propostas com foco no controle da congestão causada por envio de mensagens de alerta foram categorizadas entre proativas, reativas ou baseadas em clusterização.

Após isso, uma análise comparativa foi feita em relação aos trabalhos citados, considerando as diferentes estratégias, simuladores, métricas e resultados obtidos. Foi possível notar que existe uma lacuna sobre os requisitos de acurácia de posicionamento definidos pelas aplicações de prevenção de acidentes. Sendo que a maioria dos trabalhos também não considera as operações multicanal introduzidas no padrão IEEE 1609.4 [5], que limitam a capacidade do canal de transmissão. Partindo dessa revisão, no próximo capítulo será apresentada uma nova proposta para resolver o problema da congestão considerando os requisitos das aplicações de segurança.

Capítulo 4

Proposta de Protocolo para Controle de Congestão e Acurácia de Posicionamento em VANETs

Este capítulo apresenta a proposta *Accurate Positioning Geocast Protocol* (APGP) para lidar com o problema de congestão em VANETs que utilizam a tecnologia do padrão WAVE. A capacidade do APGP de atender os requisitos de acurácia de posicionamento das aplicações de prevenção de acidentes é um dos principais fatores que o diferencia de outros trabalhos previamente analisados. Primeiramente, realiza-se uma descrição detalhada do problema, considerando as limitações do canal de transmissão no padrão WAVE, bem quanto as restrições de posicionamento estabelecidas pelas aplicações. Após isso, apresenta-se uma visão geral do APGP, considerando sua estratégia de envio de mensagens para grupos *geocast* e a arquitetura de três componentes do protocolo. Cada componente é detalhado e por fim, uma discussão é feita acerca do comportamento esperado da proposta.

4.1 Descrição do Problema

Primeiramente, discute-se a capacidade teórica do canal de transmissão no padrão WAVE [1], com principal intuito de demonstrar o número limitado de nós que podem se comunicar em um intervalo de sincronização sem causar uma congestão. Posteriormente, os requisitos de posicionamento das aplicações serão evidenciados, destacando-se o envio frequente de *beacons*, que pode causar congestão na rede em ambientes com uma alta densidade de tráfego. Além disso, trata-se do problema de *broadcast storm*, um dos fatores que também agrava o funcionamento correto das aplicações de prevenção de acidentes.

4.1.1 Capacidade Teórica do Canal de Transmissão

Em comunicações de veículo para veículo ou V2V, não existe uma infraestrutura externa para controlar e coordenar a rede. Por este motivo, aplicações de prevenção de acidentes requerem que veículos enviem mensagens periódicas para seus vizinhos contendo suas informações, como posição geográfica, velocidade e aceleração. Estas mensagens também são conhecidas como *beacons*, BSMs [71] ou CAMs [72]. Essas aplicações definem que suas mensagens sejam transmitidas em intervalos de 100 ms (10 por segundo) até 20 ms (50 por segundo) [16]. Ao receber *beacons*, veículos podem construir a topologia da rede e definir rotas para transmissão de mensagens de alerta.

Considerando as informações apresentadas nas Seções 2.3.1 e 2.3.2, que descrevem a camada física e de enlace do padrão WAVE, pode-se afirmar que as mensagens em uma rede veicular podem ser enviadas por dois tipos de canal de transmissão. O Canal de Controle (do inglês, *Control Channel*) (CCH) é alocado para mensagens de segurança com alta prioridade, como os *beacons* e mensagens de alerta, enquanto o Canal de Serviço (do Inglês, *Service Channel*) (SCH) é alocado para mensagens não relacionadas à segurança. Cada tipo de canal divide um intervalo de sincronização de 100 ms para realizar suas transmissões, ou seja, 50 ms para o *Control Channel Interval* (CCHI) e 50 ms para o *Service Channel Interval* (SCHI).

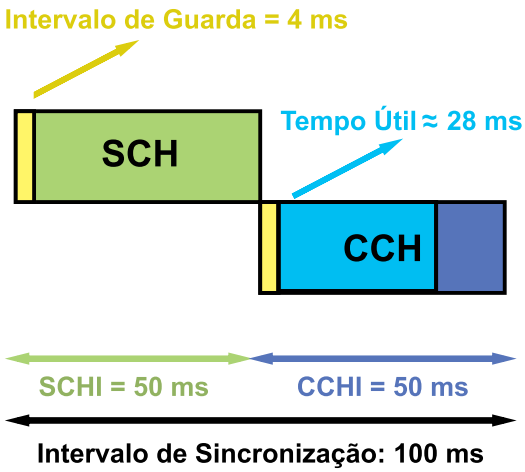

Figura 4.1: Tempo útil para transmissão de *beacons* e mensagens de alerta em um sistema WAVE.

Os *beacons* são enviados através do CCH durante um intervalo CCHI menos um Intervalo de Guarda (IG), utilizado para evitar que transmissões distintas não interfiram umas nas outras ou causem sobreposição. Assume-se também que a ocupação do canal não deve ultrapassar 60% [55], valor obtido considerando o padrão IEEE 802.11 para re-

des sem fio [18] e as restrições de tempo estabelecidas pelos padrões de redes veiculares no IEEE 1609.4 [5]. A Figura 4.1 demonstra então o intervalo útil para transmissões de mensagem considerando os fatores citados.

Com base nas informações apresentadas, estima-se a capacidade teórica do canal de transmissão com o tempo útil para transmissão de mensagens (Ω), que pode ser obtido da seguinte forma:

$$\begin{aligned} \Omega &= (CCHI - IG) \cdot 60\%, \\ \Omega &= (50 \text{ ms} - 4 \text{ ms}) \cdot 60\%, \\ \Omega &\approx 28 \text{ ms}, \end{aligned} \tag{4.1}$$

onde $CCHI$ representa o *Control Channel Interval* de 50 ms, e IG o Intervalo de Guarda de 4 ms conforme o padrão WAVE [5]. Considerando os valores apresentados, o tempo restante para transmissões durante um CCHI seria igual a apenas 28 ms, já que a ocupação do canal não deve ultrapassar 60%, como discutido anteriormente.

O tempo para a transmissão de um *beacon* foi estimado como 1 ms para um *beacon* de 256 bytes em um sistema WAVE [73]. Se o intervalo entre o envio de *beacons* for igual a 100 ms, pode-se concluir que o número máximo de nós para realizar transmissões simultâneas seria de aproximadamente 28 nessas condições. Esse resultado empírico demonstra a capacidade limitada do canal de transmissão em VANETs e como ele pode ser facilmente saturado em cenários com uma alta densidade de veículos, como em engarrafamentos ou interseções.

4.1.2 Requisitos de Acurácia de Posicionamento das Aplicações de Prevenção de Acidentes

Por conta da natureza dinâmica do ambiente veicular, aplicações voltadas para a prevenção de acidentes exigem que veículos mantenham a posição de seus vizinhos com um certo grau de acurácia. Estes requisitos foram definidos por um projeto chamado *Vehicular Safety Communications - Applications* (VSCA) criado pelo *National Highway Traffic Safety Administration* (NHTSA) [9]. Nele, três níveis de acurácia de posicionamento foram definidos como *which-road* (5,0 m), *which-lane* (1,5 m) e *where-in-lane* (abaixo de 1,0 m), que representam o erro máximo para definir se um veículo está em uma pista, em qual faixa e onde naquela faixa, respectivamente.

Aplicações veiculares de segurança como *Electronic Emergency Brake Light* (EEBL), *Forward Collision Warning* (FCW) e *Lane Change Advisor* (LCA) exigem uma acurácia de posicionamento no nível *where-in-lane* [13], ou seja, veículos devem estar cientes das

posições de seus vizinhos com um erro de menos de 1 m. Para satisfazer esse requisito, as aplicações definem altas taxas de envio de *beacons*, que podem variar entre 10 a até 50 mensagens por segundo [16], como visto na subseção anterior.

Entretanto, em cenários com alta densidade de tráfego, esse comportamento pode ocasionar em uma congestão na rede devido ao número de mensagens inseridas no canal simultaneamente. Considerando o cenário explorado na subseção anterior e as restrições de tamanho da mensagem e frequência de envio de *beacons*, quando mais de 28 nós tentarem transmitir em um mesmo intervalo de sincronização, mensagens poderão ser perdidas por erros de pacote e colisões, causando perda de acurácia de posicionamento.

4.1.3 Repasse de Mensagens de Alerta e o *Broadcast Storm*

Quando uma aplicação de segurança detecta um evento de emergência, a aplicação correspondente será acionada e o veículo envia uma mensagem de alerta para seus vizinhos mais próximos, que poderão repassá-la, se necessário, para veículos mais distantes com objetivo de evitar possíveis acidentes. Estas mensagens são disseminadas via difusão ou *broadcast*. Assim como os *beacons*, as mensagens de alerta são mensagens de segurança que devem ser transmitidas através do Canal de Controle (do inglês, *Control Channel*) (CCH), logo também podem causar a saturação do canal de transmissão.

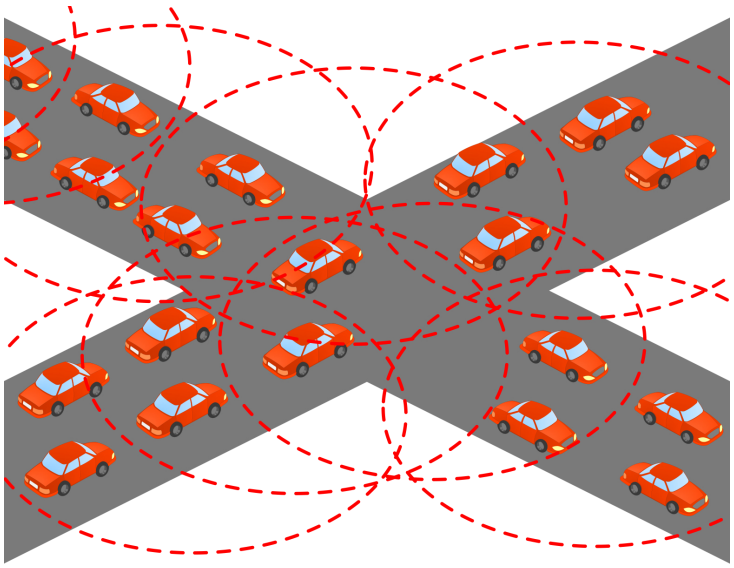

Figura 4.2: Cenário de engarrafamento propenso ao *broadcast storm*.

Em ambientes veiculares com tráfego denso, como indicado na Figura 4.2, vários veículos tentam repassar a mesma mensagem de alerta, causando a ocorrência de um *broadcast storm* [74]. Durante este fenômeno, a contenção frequente gerada pelos *beacons* e as mensagens de alerta gera erros e colisões entre pacotes. Como a entrega das mensagens de

alerta dessas aplicações é sensível ao tempo, protocolos de roteamento eficientes são essenciais para evitar a ocorrência do *broadcast storm* e da congestão do canal de transmissão.

4.2 Visão Geral do Protocolo APGP

O *Accurate Positioning Geocast Protocol* (APGP) é uma nova proposta para o envio de *beacons* e de mensagens de alerta em VANETs. O foco principal do APGP é auxiliar aplicações de prevenção de acidentes, garantindo seus requisitos de acurácia de posicionamento, enquanto também reduz a possibilidade de congestão para o envio de mensagens. Quanto ao envio dos *beacons*, este protocolo pode ser classificado como de ajuste híbrido, pois a taxa de envio e a potência de transmissão serão ajustadas. A estratégia de envio de mensagens de alerta pode ser classificada como proativa, pois a escolha dos nós encaminhadores será realizada previamente com base em fatores como a distância e estabilidade da conexão.

4.2.1 Arquitetura do Protocolo APGP

O funcionamento do protocolo APGP pode ser descrito conforme a arquitetura de três componentes vista na Figura 4.3.

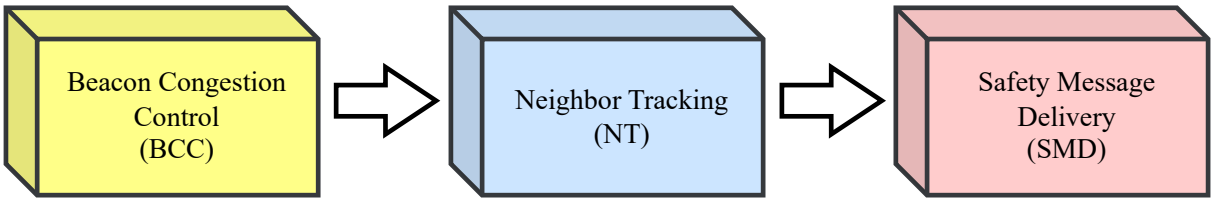

Figura 4.3: Arquitetura de três componentes do protocolo *Accurate Positioning Geocast Protocol* (APGP).

Cada nó deve ser capaz de enviar e receber *beacons* frequentemente com o auxílio do componente *Beacon Congestion Control* (BCC). O componente *Neighbor Tracking* (NT) é responsável pela manutenção dos nós vizinhos, utilizando uma lista construída através das informações presentes nos *beacons* recebidos do componente BCC. Por fim, o componente *Safety Message Delivery* (SMD) é utilizado quando a aplicação necessita enviar uma mensagem de alerta, de forma que o roteamento é realizado com base na lista de vizinhos mantida pelo componente NT. As atividades de cada componente serão descritas com mais detalhes nas seções subsequentes.

4.3 Controle de Congestão de *Beacons*

O componente *Beacon Congestion Control* (BCC) é responsável por reduzir a possibilidade de congestão na rede causada pelo envio frequente de *beacons*. Ele utiliza uma estratégia de envio de *beacons* com base em grupos *geocast*, que serão definidos durante a inicialização do protocolo. Além disso, também é responsável por informar o componente *Neighbor Tracking* (NT) sobre as informações de *beacons* recebidos para a construção da lista de vizinhos. Sendo assim, o componente é dividido em duas etapas para lidar com os requisitos RQ1 e RQ2 definidos no Capítulo 1. Na próxima subseção, explica-se a estratégia de grupos *geocast* para redução da contenção causada por *beacons* e em seguida, as tarefas do componente serão detalhadas.

4.3.1 Estratégia de Envio de *Beacons* para Grupos *Geocast*

Em redes veiculares, cada pacote pode ser enviado com um nível de potência específico, permitindo que a aplicação controle a distância que uma mensagem deve alcançar [1]. Alguns dos trabalhos analisados no Capítulo 3 utilizam essa estratégia de ajuste de potência para reduzir o número de veículos que irão receber uma mensagem, liberando o canal de transmissão rapidamente para novas transmissões. Entretanto, o ajuste de potência é geralmente feito sem considerar que veículos mais distantes do veículo de origem deixarão de receber mensagens por longos períodos, como visto nos trabalhos de Cho *et al.* [21] e Joseph *et al.* [22]. Sendo assim, o objetivo principal do componente é lidar com os seguintes requisitos:

RQ1: Diminuir a taxa de ocupação do canal com uma estratégia de ajuste de potência consciente, ou seja, que garante que todos os veículos dentro do raio de transmissão estabelecido recebam mensagens.

RQ2: Diminuir a taxa de ocupação do canal com uma estratégia de ajuste de potência consciente, ou seja, que garante que todos os veículos dentro do raio de transmissão estabelecido recebam mensagens.

A estratégia proposta neste trabalho, sugere que cada nó particione seus vizinhos em grupos distintos baseados em sua posição geográfica, também conhecidos como grupos *geocast*, um tipo de grupo *multicast*. Segundo Medhi [75], o *multicasting* pode ser definido como a entrega de pacotes gerados por uma única fonte para múltiplos nós. Diferente da difusão ou *broadcast*, o pacote não será transmitido para todos os vizinhos, mas sim para um conjunto deles que recebe o nome de grupo *multicast*.

Em redes veiculares, o *multicast* pode ser realizado com base na posição dos veículos em relação ao nó de origem, por este motivo, também é chamado de *geocast*. Um grupo *geocast* é então composto pelos nós localizados numa determinada região geográfica. Dois

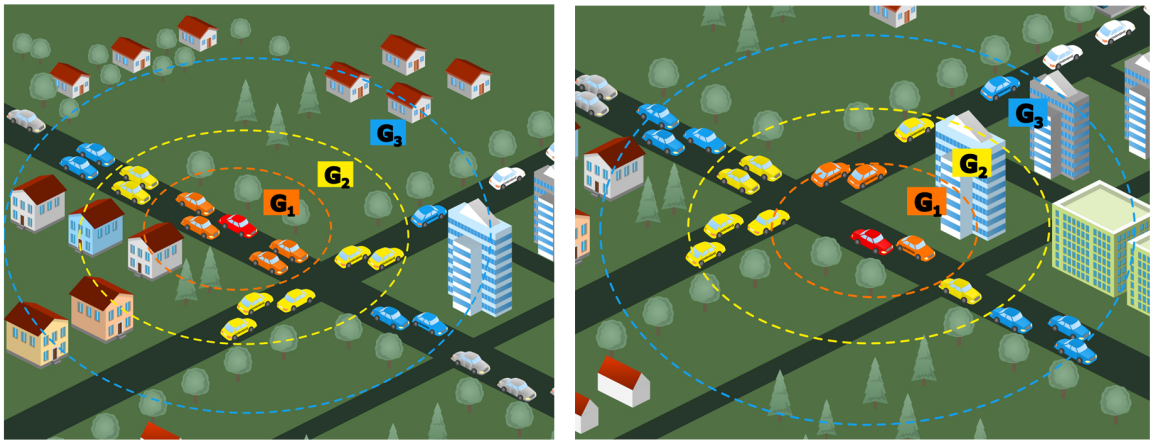

Figura 4.4: Diferentes nós de origem com Grupos de *Geocast* indicados por G_1, G_2 e G_3.

cenários com grupos *geocast* em um ambiente veicular são ilustrados na Figura 4.4, onde é possível notar que o nó de origem pode definir os membros de um grupo conforme sua distância.

A ideia principal do protocolo APGP é utilizar o ajuste de taxa de envio e o ajuste de potência, de forma que mais *beacons* serão enviados para os grupos *geocast* que estejam mais próximos do veículo de origem. Assim, conforme a Figura 4.4, espera-se que um número reduzido de veículos localizados em G_1 sempre recebam mais atualizações quando comparados aos veículos localizados nos grupos G_2 e G_3, compostos por veículos mais distantes.

Esta abordagem permite que um veículo envie informações mais frequentes para seus vizinhos mais próximos, que podem impactar no desempenho das aplicações de prevenção de acidentes diretamente como a *Electronic Emergency Brake Light* (EEBL), *Forward Collision Warning* (FCW) e *Lane Change Advisor* (LCA) [13]. Dependendo da densidade de tráfego de veículos, abordagens comuns não conseguirão manter os requisitos de acurácia de posicionamento para todos os nós devido à congestão inserida na rede pelo constante envio de *beacons*. Sendo assim, o APGP visa garantir que diferentes requisitos possam ser atendidos para diferentes grupos de nós sem aumentar a ocupação do canal consideravelmente.

4.3.2 Definição de Grupos *Geocast*

A inicialização do protocolo APGP envolve a definição das informações relevantes dos grupos *geocast* para permitir o envio de *beacons*. Para isso, denota-se um grupo *geocast* por $G = \{G_1, G_2, ..., G_i\}$, onde $i = 1 \ldots n$ e n representa o número de grupos. Cada grupo *geocast* então pode ser representado pela tupla $G_i = (\delta_i, \Delta_i, \Psi_i)$, onde δ_i é um Erro

Máximo de Posicionamento, Δ_i é a Distância de Borda e Ψ_i a Potência Máxima para um grupo de índice i. Com essas informações, será possível definir quando um *beacon* deve ser enviado e para qual grupo de vizinhos, bem quanto a potência de transmissão necessária para alcançar a distância estabelecida.

Conforme descrito na Seção 4.1, as aplicações veiculares requerem acurácia de posicionamento que pode ser classificada em *Which-Road* (5,0 m), *Which-Lane* (1,5 m) e *Where-In-Lane* (abaixo de 1,0 m), que representam respectivamente o maior erro de posicionamento aceitável para determinar em qual pista, qual faixa e aonde na faixa o veículo está localizado [9]. Neste trabalho, esses valores serão utilizados para definir o conjunto de Erro Máximo de Posicionamento como $\delta = \{1\text{ m}, 1,5\text{ m}, 5\text{ m}\}$.

Assim, torna-se possível definir os valores do Conjunto de Distância de Borda $\Delta = \{\Delta_1, \Delta_2, \ldots \Delta_i\}$, onde $i = 1 \ldots n$, com base na equação:

$$\Delta_i = \delta_i \cdot \epsilon, \tag{4.2}$$

onde δ_i representa o Erro Máximo de Posicionamento para o grupo G_i, enquanto ϵ é denominado nesse trabalho como o Fator de Escala de Distância. Pode-se afirmar que quanto maior for o valor de ϵ, maior será a distância máxima de borda de cada grupo. Consequentemente, cada grupo poderá abranger mais veículos, aumentando a contenção causada por *beacons*. A Tabela 4.1, por exemplo, considera os valores de δ informados anteriormente para os níveis de acurácia estabelecidos pelo projeto NHTSA [9], enquanto um valor de $\epsilon = 100$ é utilizado para o cálculo do conjunto de Distância de Borda Δ com a Eq. 4.2.

Tabela 4.1: Definição do conjunto inicial de grupos *geocast* considerando os valores definidos em [9] para δ.

G	**Descrição**	δ **(m)**	Δ **(m)**	Ψ **(dBm)**
G_1	*Where-In-Lane*	1	100	[-41, -28]
G_2	*Which-Lane*	1,5	150	(-28, -26]
G_3	*Which-Road*	5	500	(-26, -21]

Cada *beacon* será transmitido com uma potência de transmissão diferente conforme o grupo com o qual deseja se comunicar. Esse Conjunto de Potência Máxima é denominado como Ψ e cada valor pode ser calculado da seguinte forma:

$$P_t = \frac{\left(\frac{\lambda}{4\pi}\right)^2 \left| \frac{1}{d_{LOS}} + \frac{\mu_\perp \cdot e^{j\Delta\phi}}{d_{ref}} \right|}{P_r}, \tag{4.3}$$

onde P_t representa a potência de transmissão e o comprimento de onda por λ. O comprimento do caminho de propagação da linha de visada direta é denotado por d_{LOS}, e o

comprimento da linha de visada indireta via reflexão do solo por d_{ref}. O coeficiente de reflexão é descrito como $\mu_{\perp}$. Essa equação é utilizada no modelo de propagação *Two-Ray Interference* descrito em [76]. Ele foi escolhido, pois, considera as interferências construtivas e destrutivas na propagação do sinal, comuns no ambiente veicular. Entretanto, outros modelos de propagação também poderiam ser escolhidos, sendo necessária apenas a alteração do cálculo de P_t feito na Eq. 4.3.

A definição de um Conjunto de Grupos *Geocast*, G, pode então ser feita seguindo os passos:

1. Definir os valores do Conjunto de Distâncias de Borda (Δ) utilizando a Eq. 4.2 com o Conjunto de Erro Máximo de Posicionamento (δ) e o Fator de Escala de Distância (ϵ) como entrada.
2. Definir os valores do Conjunto de Potência Máxima (Ψ) para cada valor de Δ utilizando a Eq. 4.3.
3. Definir o Conjunto de Grupos *Geocast*: $G = \{G_1, G_2, ..., G_i\}$, onde $i = 1 \ldots n$ e $G_i = (\delta_i, \Delta_i, \Psi_i)$.

Após a definição do Conjunto de Grupos *Geocast* (G), torna-se possível enviar mensagens para apenas um conjunto de nós localizados em uma posição geográfica. Ao enviar um *beacon* para um grupo específico, basta utilizar o valor de potência calculado previamente. O APGP consegue então propor um mecanismo de ajuste de potência consciente como descrito no **RQ1**. Na subseção seguinte, apresenta-se o mecanismo para a tomada de decisão sobre o envio de *beacons*.

4.3.3 Envio de *Beacons* Utilizando o Erro de Predição e Grupos *Geocast*

Além do ajuste de potência mencionado na subseção anterior, outra abordagem para diminuir a congestão da rede em VANETs é o ajuste da taxa de envio de mensagens. Entretanto, poucos trabalhos consideram a perda de acurácia da posição causada pela diminuição na frequência de envio de *beacons*, como visto no Capítulo 3. Sendo assim, o componente *Beacon Congestion Control* (BCC) também deve ser responsável por prover um mecanismo que permita o envio consciente de *beacons*. Nesta subseção, define-se como o intervalo entre transmissões de *beacon* é controlado para cada grupo *geocast*, permitindo que o protocolo decida para quem e quando enviar novas atualizações.

Visão Geral do Mecanismo de Envio de *Beacons*

Depois que os grupos *geocast* são inicializados conforme os passos descritos na subseção anterior, o mecanismo de envio de *beacons* introduz um Conjunto de Erros de Predição $e = \{e_1, e_2, ..., e_i\}$, onde $i = 1 \ldots n$ e n representa o número de grupos *geocast*. Portanto, uma tupla de um grupo *geocast* será agora representada por $G_i = (\delta_i, \Delta_i, \Psi_i, e_i)$, onde $e_i = 0$ inicialmente. O valor do erro de predição e_i será constantemente atualizado para definir quando um *beacon* deve ser enviado para o grupo G_i.

No protocolo APGP, cada veículo é responsável por manter a sua posição atual obtida do GPS, $X_t = (x_t, y_t)$ e sua posição prevista para o próximo intervalo de sincronização, denotado como $X_{t+1} = (x_{t+1}, y_{t+1})$, que pode ser obtido através das leis da cinemática conforme as equações:

$$\begin{aligned} x_{t+\Delta t} &= x_t + s_t \Delta t, \\ y_{t+\Delta t} &= y_t + s_t \Delta t, \end{aligned} \tag{4.4}$$

onde (x_t, y_t) representa a posição do veículo no momento t, Δ_t o intervalo de tempo desde a última atualização e s_t a velocidade média do veículo no momento t.

Após todo intervalo de sincronização, cada veículo estima o Erro de Predição Local, denotado como ξ, entre sua posição atual X_t e a sua posição prevista X_{t+1} no intervalo anterior por:

$$\xi = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{4.5}$$

onde se assume $\Delta t = 1$ por simplicidade. Este valor será adicionado a cada entrada do Conjunto de Erros de Predição e da seguinte forma: $e_i = e_i + \xi$, de forma que seja possível acumular o erro de predição para cada grupo. Toda vez que este valor estiver próximo ou ultrapassar o Erro Máximo de Posicionamento δ_i, ou seja, $e_i > \delta_i$, um novo *beacon* será enviado para o grupo G_i. A Figura 4.5 apresenta dois cenários onde um veículo V_1 toma a decisão de enviar *beacons*.

No primeiro caso, o Erro de Predição $e_1 = 1.3$ m ultrapassa o limite estabelecido por $\delta_1 = 1$ m, portanto um *beacon* deve ser gerado para todos os vizinhos presentes no grupo G_1, após isso, o valor de e_1 é zerado, pois um *beacon* acaba de ser transmitido. No segundo cenário, o valor Erro de Predição $e_2 = 2.7$ m ultrapassa o limite estabelecido por $\delta_2 = 2.5$ m, definindo que uma transmissão deverá ser efetuada para o grupo G_2, e que também engloba o grupo G_1. Em casos, onde mais de um dos erros ultrapassa o limite máximo, prioriza-se o maior índice, já que seu *beacon* também englobará grupos com índices menores.

Após um *beacon* ser enviado, a posição no próximo momento X_{t+1} é atualizada com a posição atual (X_t) através da Equação 4.4. Em casos onde nenhum dos valores de Erro de

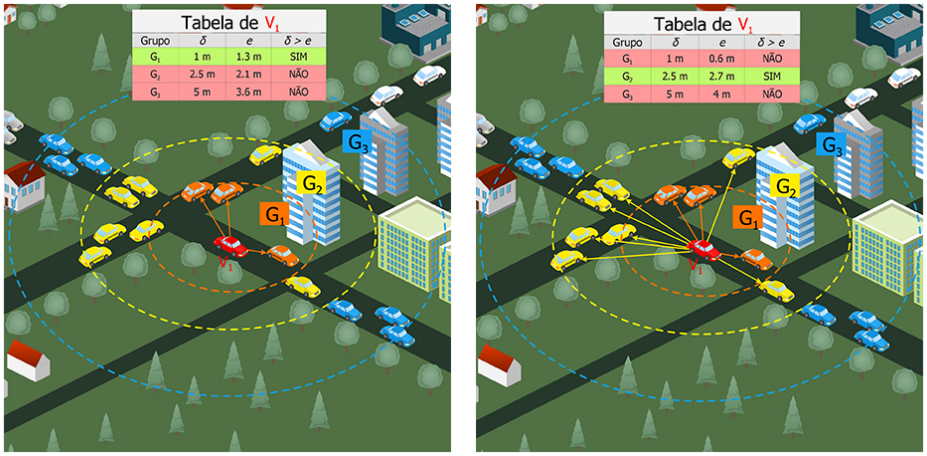

Figura 4.5: Cenários onde o veículo V_1 envia *beacons* para os grupos *geocast* G_1 e G_2.

Predição (e) ultrapassa o Erro Máximo (δ), não é necessário enviar um *beacon* e a posição de X_{t+1} é atualizada utilizando a predição realizada no intervalo anterior.

Essa abordagem permite que veículos transmitam *beacons* apenas quando for necessário aumentar a conscientização de um conjunto de vizinhos em relação a sua própria posição. Fica então claro supor que os veículos mais próximos (G_1) receberão atualizações mais frequentes, pois seu Erro Máximo de Posicionamento (δ) é menor, enquanto os veículos em outros grupos receberão atualizações menos frequentes quanto maior for seu valor de (δ).

Algoritmo de Envio de *Beacons*

O Algoritmo 1 mostra os passos necessários para realizar a tomada de decisão de envio de *beacons*, enquanto a Tabela 4.2 traz os símbolos necessários para descrever o algoritmo.

Primeiramente, um conjunto de grupos *geocast* (G) deve ser recebido como entrada dos passos indicados na Subseção 4.3.2. Então um conjunto de Erros de Predição (e) é criado para os n grupos *geocast* na Linha 2, onde cada valor é inicializado com zero. Feito isso, na Linha 3, o conjunto de Dados dos grupos *geocast* (G) recebe as informações do conjunto de Erros de Predição (e).

Na linha 5, depois de todo intervalo de sincronização, cada veículo estima o Erro de Predição Local (ξ) entre sua posição prevista (X_{t+1}) e a posição atual (X_{cur}) utilizando a Eq. 4.5. Na Linha 6, inicializa-se o índice i com zero, pois nenhum grupo *geocast* foi escolhido até o momento.

Na linha 7, itera-se por cada Erro de Predição (e_i) e se adiciona o Erro de Predição Local (ξ) a ele. Se o novo valor do Erro de Predição (e_i) estiver acima do Erro Máximo

Tabela 4.2: Símbolos usados no algoritmo 1

Símbolo	**Descrição**
G	Conjunto de Dados de Grupos *Geocast*.
G_i	Grupo *geocast* $i = (\delta_i, \Delta_i, \Psi_i, e_i)$.
n	Número de Grupos *Geocast*.
δ	Conjunto de Erro Máximo de Posicionamento.
Ψ	Conjunto de Potência Máxima.
e	Conjunto de Erros de Predição.
e_i	Erro de Predição acumulado para um grupo G_i.
X_t	Posição atual do veículo no momento t.
X_{t+1}	Posição prevista do veículo no momento $t+1$.
ξ	Erro de Predição Local da posição do veículo.
i	Índice que representa o grupo que receberá um novo *beacon*.
P_t	Potência de transmissão que será usada para transmitir um *beacon*.

de Posicionamento (δ_i), um *beacon* será transmitido para o grupo G_i, portanto $i \leftarrow j$ na Linha 10. É importante notar que apenas um *beacon* é transmitido por vez, portanto i sempre terá o valor do grupo com maior índice, já que os grupos com índices menores também serão alcançados pelo *beacon*.

Na Linha 13, se um grupo foi selecionado, então o índice i sempre será maior que zero. Se $i = 2$, por exemplo, a potência de transmissão atual P_t será igual a Ψ_2, obtida de G_2. Quando o *beacon* for transmitido, ele alcançará todos os vizinhos dos grupos G_1 e consequentemente G_2, semelhante ao segundo cenário apresentado na Figura 4.5.

No entanto, na Linha 14 o Erro de Predição (e_i) só será reinicializado como zero para aquele grupo específico, ou seja, $e_2 = 0$ no exemplo mencionado. A posição da predição (X_{t+1}). O beacon é finalmente enviado na Linha 15 para o grupo indicado. Por fim, a posição no momento seguinte pode ser atualizada com a posição atual (X_t) na Linha 16. Se nenhum grupo *geocast* for escolhido, ele será atualizado com a última posição prevista X_{t+1}. Na Linha 18, o novo valor X_{t+1} é calculado usando as leis da cinemática indicadas na Eq. 4.4. Esse algoritmo permite realizar o ajuste da taxa de envio de *beacons*, considerando a acurácia de posicionamento necessária para cada grupo *geocast*, sendo esse o requisito RQ2 definido no Capítulo 1.

Recebimento de *Beacons*

Um veículo poderá receber *beacons* de seus vizinhos dependendo de sua distância até eles. Um *beacon* pode ser definido como:

$$b_k = \{v_k, (x_k, y_k, z_k), s_k, d_k\}, \tag{4.6}$$

Algorithm 1 Mecanismo de envio de *beacons*.

Require: G, n
1: $X_{t+1} \leftarrow (0, 0)$
2: $e \leftarrow \{e_1, e_2, ..., e_i\}$, onde $i = 1 \ldots n$ e $e_i \leftarrow 0$
3: $G \leftarrow G \cup e$
4: **Depois de todo** intervalo de sincronização **do**:
5: Calcule ξ com a Eq. 4.5 com X_{t+1} e X_t
6: $i \leftarrow 0$
7: **for** $j \leftarrow 1$ até n **do**
8: $e_i \leftarrow e_i + \xi$
9: **if** $e_i > \delta_i$ **then**
10: $i \leftarrow j$ ▷ Grupo G_i receberá um *beacon*.
11: **end if**
12: **end for**
13: **if** $i > 0$ **then**
14: $P_t \leftarrow \Psi_i$; $e_i \leftarrow 0$
15: Envie novo *beacon* com potência de transmissão P_t para o Grupo G_i
16: $X_{t+1} \leftarrow X_t$ ▷ Atualiza a posição
17: **end if**
18: Atualize o valor de X_{t+1} com a Eq. 4.4 para o próximo intervalo

onde v_k representa o ID de um veículo, (x_k, y_k, z_k) sua posição geográfica medida por GPS, s_k sua velocidade e d_k sua direção. Quando um veículo recebe um novo *beacon*, sua única função é enviar as informações dele para o componente *Neighbor Tracking* (NT), que tomará as atitudes necessárias caso seja a primeira vez que recebeu o *beacon* ou não. Na seção seguinte, apresenta-se com mais detalhes o componente responsável por monitorar os vizinhos.

4.4 Monitoramento de Vizinhos

Cada veículo que estiver utilizando o protocolo APGP deve manter uma lista atualizada de seus vizinhos denotada como L. Essa lista deve conter informações dos veículos a partir dos *beacons* recebidos do componente BCC, além de fornecer informações para o componente SMD para a escolha de nós encaminhadores. Além disso, a lista deve ser constantemente atualizada, removendo vizinhos que não enviaram *beacons* em longos intervalos de tempo ou que estão fora do alcance. Esse componente tem então como foco principal atender ao requisito RQ3 definido no Capítulo 1.

RQ3: Manter uma lista de candidatos (vizinhos) para auxiliar no envio de mensagens de alerta das aplicações sobre possíveis eventos, como acidentes e problemas na pista.

4.4.1 Estrutura da Lista de Vizinhos

A lista de vizinhos L tem como principal objetivo manter as informações necessárias sobre os *beacon* recebidos do componente BCC. Neste trabalho, cada entrada nesta lista tem a seguinte estrutura:

$$l_k = \{v_k, X_k, s_k, t_k, \sigma_k, G_k\}, \tag{4.7}$$

onde v_k representa o ID de um veículo, $X_k = (x_k, y_k, z_k)$ sua posição geográfica medida por GPS, s_k sua velocidade, t_k o tempo desde que o último *beacon* foi recebido, σ_k seu Grau de Estabilidade e G_k o grupo *geocast* daquele vizinho. Com exceção de σ_k e G_k, todos os outros valores podem ser extraídos ou inferidos dos *beacons*. Um exemplo de uma lista de vizinhos L de um veículo v_1 pode então ser visto na Tabela 4.3

Tabela 4.3: Exemplo da lista de vizinhos de um veículo v_1.

ID	**Posição** (m)	**Velocidade** (m/s)	**Tempo** (s)	**Estabilidade**	**Grupo *Geocast***
v_2	(322,4, 321,53, 1,2)	14	0.3	17	G_1
v_3	(352,4, 351,53, 1,3)	15	0.6	12	G_2
v_4	(472,4, 481,53, 1,2)	16	1.4	2	G_3

O Grau de Estabilidade (σ_k) será utilizado posteriormente para definir os melhores vizinho para rotear uma mensagem de alerta. Este valor será inicializado com 1 ao receber um *beacon* de um vizinho pela primeira vez. Com o passar do tempo, este valor poderá aumentar quanto mais *beacons* forem recebidos do mesmo vizinho. Assim, quanto maior for σ maior será a estabilidade da conexão com aquele veículo.

4.4.2 Adição e Atualização de Vizinhos

Quando o componente *Neighbor Tracking* (NT) recebe um *beacon*, sua primeira tarefa é checar se o ID daquele veículo já existe na lista, no passo $v_i \in L$, caso ele não exista, o veículo deverá criar uma entrada para aquele vizinho com a mesma estrutura apresentada na Eq. 4.7. Como já mencionado, o Grau de Estabilidade (σ_k) será inicializado com 1 ao receber o primeiro *beacon*. Para decidir o grupo *geocast* do novo vizinho (v_i), o veículo de origem deve calcular sua distância até ele com a Eq. 4.5. Este valor é então comparado com as Distâncias de Borda (Δ) de cada grupo.

Tabela 4.4: Definição do grupo *geocast* de novos *beacons* recebidos por v_1.

ID	**Posição** (m)	**Distância** (m)	**Grupo *Geocast***
v_2	(322,4, 321,53, 1,3)	$72,54$	G_1
v_3	(380,5, 381,9, 1,3)	$114,84$	G_2
v_4	(472,1, 481,66, 1,3)	$250,23$	G_3

A Tabela 4.4 ilustra um exemplo onde um veículo v_1 localizado em (300 m, 300 m, 1.3 m) recebeu novos *beacons* de v_2, v_3 e v_4. Considerando o Conjunto de Distâncias de Borda $\Delta = \{100\text{m}, 150\text{m}, 500\text{m}\}$ da Tabela 4.1, torna-se simples checar que o veículo v_2 está na zona de cobertura do G_1, pois está abaixo da distância de borda $\Delta_1 = 100$ m. O mesmo raciocínio pode ser aplicado para definir os grupos dos outros vizinhos.

Quando um veículo recebe um *beacon* de um vizinho v_i que já existe em L, basta atualizar sua posição com a informação advinda do *beacon*. O tempo desde que recebeu um *beacon* (t_i) será reinicializado para zero e o contador estabilidade de conexão (σ_i) incrementado.

4.4.3 Manutenção da Lista de Vizinhos

Depois de todo intervalo de sincronização, o componente NT deve checar quais vizinhos estão fora do alcance de transmissão e quais não enviam *beacons* há muito tempo. Para isso, primeiro realiza-se uma atualização da posição de cada vizinho com base nas leis da cinemática com as Eq. 4.4. Aqueles vizinhos que estiveram mais distantes que a Distância de Borda Δ do grupo com maior índice serão removidos. A lista de vizinhos L também mantém uma constante t_{max}, que determina o tempo máximo de permanência de um vizinho. Na Tabela 4.3, por exemplo, o veículo v_4 seria removido se $t_{max} = 1$ s, já que o valor de t_4 excede o limite máximo de permanência.

4.5 Envio de Mensagens sobre Eventos Críticos

Quando um evento crítico é detectado na pista, as aplicações de segurança são responsáveis por transmitir mensagens de alerta para veículos vizinhos sobre o ocorrido. Como citado na Seção 4.1, essas mensagens são transmitidas por difusão ou *broadcast*, onde os vizinhos podem repassar a mensagem novamente para conscientizar outros veículos sobre o evento. Esse modo de transmissão pode ocasionar em um *broadcast storm* em cenários com tráfego denso, pois erros e colisões entre pacotes se tornarão frequentes quando muitas mensagens estiverem sendo transmitidas simultaneamente. Isso também impedirá os veículos de transmitirem seus *beacons* e mensagens de alerta corretamente.

Para auxiliar as aplicações de segurança, o protocolo APGP também possui um componente chamado *Safety Message Delivery* (SMD). A estratégia utilizada permite que veículos transmitam mensagens de alerta numa Zona de Interesse (ZdI) através do *geocast*. Este modo de disseminação diminui a congestão da rede, pois as mensagens serão enviadas apenas para nós restritos a uma posição geográfica, que podem então repassar essa informação para outras regiões, conforme a necessidade da aplicação. O fluxograma da Figura 4.6 apresenta o esquema de funcionamento do componente SMD, que será

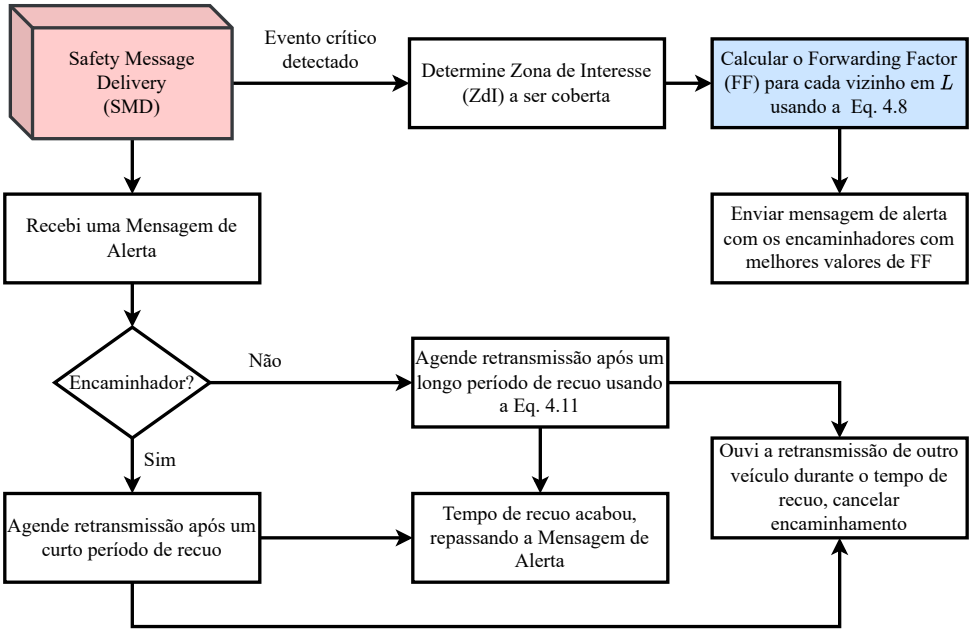

Figura 4.6: Fluxograma de funcionamento do componente *Safety Message Delivery* (SMD).

apresentado com mais detalhes nas subseções seguintes. Ele tem como principal objetivo atender o requisito RQ4 definido no Capítulo 1.

RQ4: Apresentar um mecanismo confiável e com baixo atraso para a transmissão das mensagens de alerta sobre eventos críticos.

4.5.1 Envio de Mensagens de Alerta

Quando um veículo detecta um evento crítico ele deve determinar uma Zona de Interesse (ZdI). Isso significa que a mensagem de alerta conterá posições geográficas indicando a área que a mensagem deve cobrir, por exemplo, 1 km para trás da posição atual do veículo. Dependendo do valor da ZdI, o veículo que originou a mensagem não conseguirá cobri-la, por este motivo, escolhem-se vizinhos para repassarem a mensagem adiante até que a zona seja coberta da melhor forma possível.

Para decidir qual veículo deve repassar a mensagem de alerta, utiliza-se a lista de vizinhos (L) para calcular o *Forwarding Factor* (FF) com:

$$FF = \alpha \cdot d_{factor} + \beta \cdot \sigma_{factor}, \tag{4.8}$$

onde d_{factor} representa o Fator de Distância e σ_{factor} o Fator de Estabilidade, enquanto α e β são constantes que definem quanto cada fator deve contribuir no cálculo.

O Fator de Distância (d_{factor}) representa quanto a distância do vizinho deve influenciar na decisão do repasse da mensagem, esse cálculo é feito considerando a seguinte equação:

$$d_{factor} = \frac{d_k}{d_{max}}, \tag{4.9}$$

onde d_k representa a distância do veículo de origem até o vizinho v_k e d_{max} o máximo alcance de transmissão do veículo de origem. Veículos que estão mais distantes serão priorizados, já que o valor de d_{factor} será mais alto, enquanto veículos mais próximos terão valores menores em comparação.

O Fator de Estabilidade (σ_{factor}) indica o grau de estabilidade da conexão entre o veículo de origem e seu vizinho, esse valor pode ser calculado como:

$$\sigma_{factor} = \frac{\sigma_k}{\sigma_{max}}, \tag{4.10}$$

onde σ_k representa o Grau de Estabilidade do vizinho e σ_{max} o Grau de Estabilidade Máximo presente na lista de vizinhos do veículo. Assim como no Fator de Distância, os veículos com maior estabilidade apresentarão valores de σ_{factor} maiores, pois o veículo de origem recebeu mais *beacons* corretamente desse vizinho. Essa abordagem para o cálculo do *Forwarding Factor* (FF) permite considerar tanto a distância, quanto a estabilidade da conexão entre os veículos.

Após iterar sobre a lista de vizinhos L e calcular o valor de FF para todos os veículos presentes nela, o veículo de origem escolhe aqueles com os melhores valores de FF. Além de informações básicas como a posição geográfica, velocidade, direção, a mensagem de alerta também terá uma lista de IDs dos vizinhos que deverão encaminhar a mensagem e uma Zona de Interesse a ser coberta.

4.5.2 Recebimento e Encaminhamento de Mensagens de Alerta

Seguindo o esquema apresentado na Figura 4.6, quando um veículo recebe uma mensagem de alerta, ele primeiro deve checar se o seu ID está na lista de encaminhadores da mensagem. Se sim, uma retransmissão daquela mensagem é agendada após um pequeno período de recuo ou *backoff*. Caso contrário, a retransmissão é agendada após um período de recuo mais longo, que prioriza veículos mais distantes. A janela de contenção CW é calculada segundo Palazzi *et al.* [77] através da equação:

$$CW = [\frac{d_{max} - d_{factor}}{d_{max}}(CW_{max} - CW_{min})] + CW_{min}, \tag{4.11}$$

onde d_{factor} é o fator de distância calculado no passo anterior, CW_{max} e CW_{min} representam os valores de máximo e mínimo para a janela de contenção no padrão IEEE 802.11p [1]. Com isso, o veículo poderá determinar o seu tempo de recuo na janela calculada.

Essa estratégia de reenvio quando o veículo não é o encaminhador funciona como uma alternativa para os casos de falha quando nenhum dos nós encaminhadores consegue receber a mensagem corretamente. Esse cenário pode ser comum, já que o veículo de origem não consegue prever quando seus vizinhos poderão estar no meio de seu intervalo de uso do Canal de Serviço (do Inglês, *Service Channel*) (SCH), ao invés do canal de controle CCH, voltado para mensagens de segurança. Além de que colisões e erros de pacote podem ocorrer em cenários com tráfego denso.

Quando um veículo dentro do seu período de recuo receber outra cópia de uma mensagem que ele já recebeu, ele poderá cancelar sua retransmissão. A decisão de retransmitir a mensagem poderá ser feita novamente, caso o veículo perceba que a ZdI ainda não foi coberta. Nesse caso, ele poderá seguir os mesmos passos mencionados anteriormente para escolher o melhor encaminhador e por fim, repassar a mensagem. Quando a zona for coberta, a mensagem de alerta é simplesmente transmitida sem encaminhadores, indicando o fim da transmissão.

4.6 Discussão

Neste capítulo, apresentou-se o protocolo *Accurate Positioning Geocast Protocol* (APGP) como uma nova solução para o problema discutido na Seção 4.1. Espera-se que o APGP consiga controlar a congestão no canal de transmissão através do ajuste da taxa de envio e da potência de transmissão dos *beacons* com os grupos *geocast*. Também se espera que seja possível evitar a ocorrência de *broadcast storm* durante a transmissão de mensagens de alerta com o componente SMD. O protocolo deve assegurar que os requisitos de acurácia de posicionamento das aplicações sejam mantidos através do mecanismo de predição local descrito no Algoritmo 1. No próximo capítulo, apresenta-se o cenário de simulação construído para colher dados e mostrar os resultados experimentais com as métricas relevantes para avaliar o desempenho do protocolo APGP.

Capítulo 5

Resultados Experimentais

Este capítulo tem como principal objetivo analisar os resultados experimentais do protocolo APGP descrito no Capítulo 4. Inicialmente, apresenta-se a metodologia utilizada descrevendo o ambiente de simulação, bem quanto as configurações, parâmetros e o cenário escolhido. As métricas e os protocolos utilizados para a comparação também serão descritos. Em seguida, os resultados obtidos das simulações são apresentados por gráficos e tabelas para medir o desempenho do protocolo APGP. Por fim, uma discussão é feita acerca do desempenho do protocolo quando comparado às outras propostas.

5.1 Metodologia

A metodologia utilizada para a avaliação do desempenho do protocolo APGP é apresentada com mais detalhes nessa seção. O ambiente virtual será especificado considerando a configuração de simuladores, os parâmetros de simulação e o cenário escolhido. As métricas para a avaliação do protocolo também serão detalhadas. Os protocolos relacionados que serão usados para comparação também são descritos. Como informado anteriormente, o foco desse trabalho é na tecnologia do padrão WAVE.

5.1.1 Ambiente de Simulação

A configuração escolhida para este trabalho é baseada no *framework* Veins [50], no simulador de mobilidade SUMO [39] e no simulador de rede OMNeT++ [47]. Como discutido na Seção 2.5 e em trabalhos como [6,78], a configuração SUMO/OMNeT++/Veins é uma das mais utilizadas para simulação em VANETs. O Veins é um *framework* que implementa toda pilha IEEE 802.11p/WAVE [1] e permite replicar a interferência nas transmissões causadas por obstáculos. Além disso, também oferece a implementação de modelos de propagação para calcular o desvanecimento do sinal.

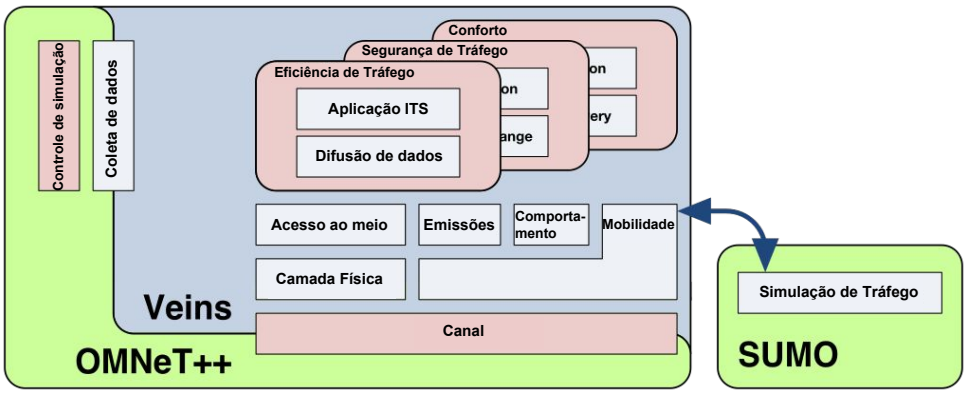

Figura 5.1: Visão geral do cenário de simulação.

A configuração dos simuladores pode ser vista na Figura 5.1, onde é possível notar que o SUMO é responsável pela simulação do tráfego, permitindo a modelagem de pistas e rodovias, bem quanto definir o comportamento dos veículos e das rotas que eles devem utilizar. O SUMO se comunica com o Veins através do simulador de rede OMNeT++ responsável por lidar com o controle da simulação, a coleta de dados, bem quanto prover serviços para a transmissão de mensagens. O Veins opera junto ao OMNeT++ como um *framework* responsável por implementar as últimas adições dos padrões de redes veiculares. Ele permite analisar a mobilidade e comportamento dos veículos, bem quanto realizar a difusão de dados e o uso das aplicações ITS [50].

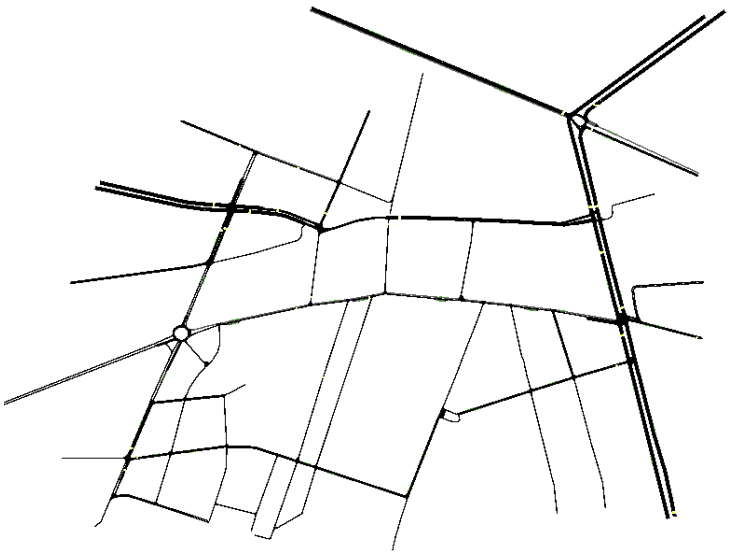

Figura 5.2: Quadrante localizado na cidade de Bologna, Itália.

O quadrante indicado na Figura 5.2 representa uma área obtidas através da ferramenta OpenStreetMap [79] e convertida para um formato compatível com o simulador de

mobilidade SUMO. A região representa uma parte na cidade de Bologna na Itália. Este cenário é disponibilizado para uso público pelo DLR-TS [80]. Os veículos podem trafegar com diferentes velocidades por qualquer uma das pistas, onde também é possível notar a presença de interseções e semáforos.

Os experimentos foram realizados em uma máquina virtual com o sistema operacional Linux, em específico a distribuição do Ubuntu 64 bits na versão 16.04 com 8GB de RAM. A versão utilizada do OMNeT++ foi a 5.1.1, do SUMO 0.25 e a versão do framework Veins 5.1. Outros parâmetros de simulação utilizados podem ser vistos na Tabela 5.1.

Tabela 5.1: Parâmetros de simulação.

Parâmetro	**Valor**
Tempo de Simulação (S_{time})	25, 50, 75, 100 s
Densidade Veicular	40, 50, 70, 80 veh/km^2
Tempo de Aquecimento	100 s
Raio de Transmissão	500 m
Modelo de Propagação	*Two-Ray Interference* (TRI)
Altura da Antena	1.85 m
Potência de Transmissão Inicial	20 mW
Sensibilidade do Rádio	-89 dBm
Taxa de Dados	3 Mb/s
Tamanho do *Beacons*	256 bytes
Tamanho da Mensagem de Alerta	500 bytes

Os veículos trafegam no cenário indicado na Figura 5.2 durante os intervalos de tempo de simulação indicados na Tabela 5.1. Considera-se também um tempo de aquecimento de 100 s para os veículos ingressarem na simulação. O raio de transmissão máximo é de 500 m.

O modelo de propagação *Two-Ray Interference* (TRI) foi escolhido para calcular o desvanecimento do sinal, pois também considera interferências construtivas e destrutivas [76]. A potência inicial de transmissão é de 20 dBm, enquanto a sensibilidade de rádio do receptor é igual a -89 dBm e a taxa de dados 3 Mb/s, valores que são frequentemente utilizados durante a simulação de protocolos em redes veiculares [50]. O tamanho de cada mensagem de *beacon* é igual a 256 *bytes*, enquanto a mensagem de alerta tem tamanho igual a 500 bytes. Foram realizadas pelo menos 10 simulações em cada cenário, onde os resultados são representados com um intervalo de confiança de 95%.

Os parâmetros utilizados para a simulação do protocolo APGP são descritos com mais detalhes na Tabela 5.2. O número de grupos *geocast* (n) é igual a 3, e a definição dos parâmetros δ, Δ e Ψ são escolhidos conforme a descrição da Seção 4.3.2, em especial da Tabela 4.1. A Constante do Fator de Distância (α) foi escolhida como $0,7$ com intenção de dar prioridade para veículos mais distantes e evitar retransmissões durante o repasse

Tabela 5.2: Parâmetros de simulação do protocolo APGP.

Parâmetro	**Valor**
Número de Grupos *Geocast* (n)	3
Conjunto de Erro Máximo de Posicionamento (δ)	{1,0 m, 1,5 m, 5,0 m}
Fator de Escala de Distância (ϵ)	100 m
Conjunto de Distâncias de Borda (Δ)	{100 m, 150 m, 500 m}
Conjunto de Potência Máxima (Ψ)	{0,2 dBm, 1 dBm, 5 dBm}
Constante do Fator de Distância (α)	0,7
Constante do Fato de Estabilidade (β)	0,3

da mensagem de alerta. A Constante do Fator de Estabilidade (β) é igual a $0,3$, para que também seja considerada a estabilidade da conexão entre os veículos.

Outros testes foram feitos com outros valores para α e β, onde foi possível notar que quanto maior fosse a Constante do Fator de Distância (α), veículos cada vez mais distantes seriam priorizados, mas que nem sempre apresentavam boa estabilidade de conexão. Entretanto, valores elevados para o Fator de Estabilidade (β) faziam com que veículos priorizassem veículos mais próximos, aumentando o número de saltos que a mensagem deveria percorrer. Considerando o cenário apresentado, os valores de $\alpha = 0,7$ e $\beta = 0,3$ apresentaram os melhores resultados para balancear a distância e estabilidade de conexão.

5.1.2 Métricas de Avaliação

Como discutido na análise comparativa dos trabalhos analisados no Capítulo 3, algumas métricas são essenciais para mensurar o desempenho de um protocolo ou algoritmo em VANETs. Neste trabalho, considerou-se a taxa de *beacons* gerados, a taxa de recepção de pacotes, a taxa de ocupação do canal, a taxa de atraso de entrega e o erro de posicionamento médio. Cada uma dessas métricas será explicitada a seguir, bem quanto o modo como elas foram calculadas no ambiente de simulação proposto.

Taxa de *Beacons* Gerados

A Taxa de *Beacons* Gerados (T_{BG}) é uma métrica simples utilizada para se obter uma média das mensagens deste tipo que foram transmitidas durante a simulação. Cada veículo deve manter um contador, inicializado como zero. Toda vez que ele transmite um *beacon*, esse valor será incrementado. No final da simulação, basta realizar a soma dos valores medidos para cada veículo e dividir pelo número total de veículos da seguinte forma:

$$T_{BG} = \frac{1}{k} \sum_{i=0}^{n} B_i, \tag{5.1}$$

onde BG representa a Taxa de *Beacons* Gerados, k o número total de veículos naquela simulação e B é o contador de *beacons* transmitidos por um veículo v_i tal que $i = 1 \ldots k$. Gerar mais ou menos *beacons* nem sempre indica um bom desempenho da estratégia adotada, por este motivo, outras métricas também serão analisadas.

Taxa de Recepção de Pacotes

A Taxa de Recepção de Pacotes (T_{RP}) é uma métrica fundamental para determinar o bom desempenho de qualquer rede, onde é possível inferir se um protocolo consegue evitar colisões e erros de pacotes, bem quanto inferir se o canal está sendo utilizado da melhor forma possível. Durante a simulação, cada veículo também mantém contadores dos pacotes recebidos corretamente e dos pacotes perdidos por colisões, erros ou por estar transmitindo enquanto recebia. Essa métrica pode então ser calculada como:

$$T_{RP} = \frac{P_{REC}}{P_{REC} + P_{COL} + P_{RXTX} + P_{ERROR}}, \tag{5.2}$$

onde TRP é a Taxa de Recepção de Pacotes, que pode incluir tanto *beacons* como mensagens de alerta. P_{REC} é a soma de todos os pacotes recebidos corretamente, P_{COL} é a soma dos pacotes perdidos por colisões, P_{RXTX} a soma dos pacotes perdidos por estar transmitindo enquanto recebia e P_{ERROR} a soma dos pacotes perdidos por erros durante a decodificação do pacote. Esses valores podem ser obtidos através do simulador.

Taxa de Ocupação do Canal

A Taxa de Ocupação do Canal (T_{OC}) é outra métrica fundamental para avaliar o funcionamento de um protocolo em redes. Valores altos desta grandeza podem indicar que o canal se tornou saturado e propenso a colisões e erros de pacotes. Enquanto valores muito baixos, podem indicar que o canal está ocioso por grande parte do tempo. Cada veículo mantém uma variável com o tempo que o canal foi considerado ocupado durante a simulação. Toda vez que ele transmite ou recebe um pacote, este valor é atualizado de acordo. Esta métrica pode então ser calculada por:

$$T_{OC} = \frac{1}{S_{time}} \sum_{i=0}^{k} OC_i, \tag{5.3}$$

onde T_{OC} é a Taxa de Ocupação do Canal, S_{time} é o tempo útil para transmissão durante a simulação e OC_i é a variável responsável por guardar o tempo em que o canal foi considerado ocupado por cada veículo v_i, de forma que $i = 1 \ldots k$ e k é o total de veículos durante a simulação.

Taxa de Atraso de Entrega

A Taxa de Atraso de Entrega (T_{AE}) é utilizada para determinar o tempo em que uma mensagem de alerta demorou para chegar até cada veículo durante a simulação. Ela é utilizada neste trabalho para medir o desempenho do componente de envio de mensagens de alerta, descrito na Seção 4.5. Quanto menor o atraso, melhor será o desempenho da aplicação. Quando uma mensagem de alerta é gerada, o veículo que originou a mensagem deve incluir o tempo atual em seu cabeçalho além da Zona de Interesse (ZdI), que indica a distância que a mensagem deve percorrer. Ao receber esta mensagem, um veículo vizinho pode então medir o tempo que ela levou para chegar até ele. A taxa de atraso de entrega pode então ser calculada como:

$$T_{AE} = \frac{1}{k_{ZdI}} \sum_{i=0}^{n} (m_{r_i} - m_{t_i}), \tag{5.4}$$

onde T_{AE} é a Taxa de Atraso de Entrega, m_{r_i} o tempo que o veículo v_i recebeu a mensagem de alerta e m_{t_i} o tempo que a mensagem foi transmitida. Desta vez, k_{ZdI} representa o número total de veículos na ZdI indicada na mensagem. Com este cálculo, torna-se possível ter uma média do tempo em que a mensagem levou para cobrir os veículos pertinentes na simulação.

Erro de Posicionamento Médio

O Erro de Posicionamento Médio (EPM) é uma medida utilizada para determinar a acurácia da posição dos veículos vizinhos armazenados em uma lista de vizinhos L. Com esta métrica, torna-se possível estimar se os requisitos estabelecidos pelas aplicações de segurança estão sendo atendidos. Para medir este erro é necessário obter a posição real dos veículos na simulação e comparar com o valor guardado na lista de vizinhos do veículo. Os passos detalhados para calcular o EPM são descritos no Algoritmo 2, enquanto a Tabela 5.3 mostra os símbolos utilizados.

Tabela 5.3: Símbolos usados no algoritmo 2

Símbolo	**Descrição**
t	Contador de vezes que o erro de posicionamento foi calculado.
ω	Erro de posicionamento acumulado.
v_k	ID de um veículo presente em L.
L	Lista de vizinhos do veículo.
X_k	Posição geográfica do veículo v_k.
X_{Lk}	Posição geográfica estimada do veículo v_k na lista de vizinhos L.
ξ_k	Erro de posicionamento entre X_k e X_{Lk}.
ξ	Erro de posicionamento acumulado dos vizinhos.

Algorithm 2 Cálculo do erro de posicionamento acumulado de um veículo v.

1: $t = 0; \omega = 0$
2: **Depois de todo** intervalo de sincronização **do**:
3: **for each** $v_k \in L$ **do**
4: Obter a posição real do vizinho v_k denominada X_k através do simulador
5: Obter a posição prevista do vizinho v_k de L, denominada X_{Lk}
6: Calcular o erro ξ_k entre X_k e X_{Lk} através da Eq. 4.5
7: $\xi = \xi + \xi_k$
8: **end for**
9: $\omega = \omega + \xi/|L|$
10: $t = t + 1$
11: **end do**
12: No final da simulação calcule $\omega = \omega/t$

O Algoritmo 2 inicia um contador j com zero, que será utilizado para medir o número de vezes que o veículo calculou o erro de posicionamento médio durante a simulação. No Passo 2, depois de todo intervalo de sincronização de geralmente 100 ms, um veículo poderá iterar sob sua lista de vizinhos no Passo 3 e medir o erro observado para a posição de cada vizinho. Durante a simulação é possível obter a posição real de qualquer veículo vizinho v_k denominada X_k. Além disso, também é possível acessar a posição estimada para aquele vizinho presente em L e denominada X_{Lk}. No Passo 6, calcula-se o erro de posicionamento ξ_k com a Eq. 4.5, que representa a distância Euclideana entre as duas posições. O valor do erro de posicionamento é acumulado no Passo 7.

Após iterar sob todos os vizinhos presentes na lista L, no Passo 9 a média do erro acumulado $\xi/|L|$ é adicionada ao erro total acumulado ω. O contador de vezes que esse valor foi atualizado é incrementado no Passo 10. No final da simulação, o valor do erro total acumulado ω é atualizado considerando o número de vezes que esse valor foi atualizado. Com isso, torna-se possível calcular o Erro de Posicionamento Médio (EPM) por:

$$EPM = \frac{1}{n} \sum_{i=0}^{n} \omega_i \tag{5.5}$$

onde ω_i é uma média do erro de posicionamento acumulado observado pelo veículo v_i, tal que $i = 1 \ldots n$ e n é o número total de veículos na simulação.

5.1.3 Protocolos Utilizados para Comparação

A metodologia usada para avaliar o desempenho do protocolo APGP também considera a comparação dos resultados com outros protocolos relevantes para a comunicação em VANETs. Escolheu-se um protocolo de comunicação baseado no padrão IEEE 802.11p

[1] e o protocolo DC-BTRP [2]. Ambos foram implementados no mesmo ambiente de simulação que o APGP e serão descritos com mais detalhes a seguir.

Protocolo Baseado no IEEE 802.11p

A comparação com o protocolo baseado no IEEE 802.11p [1] tem como principal objetivo mostrar como o padrão atual não considera a congestão da rede. Nesse protocolo, *beacons* são enviados a uma taxa de 10 por segundo, aumentando consideravelmente a ocupação do canal. As mensagens de alerta são enviadas por difusão ou *broadcast*, sem nenhum mecanismo para impedir a ocorrência do *broadcast storm*. Espera-se que os resultados desse protocolo apontem uma alta taxa de ocupação do canal de transmissão, bem quanto dificuldades para manter a taxa de recepção de pacotes em um nível aceitável para as aplicações de prevenção de acidentes.

DC-BTRP

O protocolo DC-BTRP [2] foi escolhido para comparação por se aproximar da estratégia proposta no APGP. Além de realizar o ajuste da taxa de envio de *beacons* o DC-BTRP também ajusta a potência de transmissão com base na congestão estimada no canal. Além disso, o protocolo também possui um mecanismo para garantir a acurácia de posicionamento, que será comparado com o do protocolo APGP. O funcionamento mais detalhado desse protocolo específico pode ser visto na Seção 3.2.3.

5.2 Resultados Experimentais

A estratégia de envio com grupos *geocast* do protocolo APGP será analisada, com principal objetivo de demonstrar seu funcionamento nos cenários propostos em relação ao número de *beacons* gerados, bem quanto o número de vizinhos de cada grupo. Em seguida, uma análise do desempenho do controle da congestão e acurácia de posicionamento será realizada utilizando as métricas apresentadas previamente. O protocolo APGP será comparado com outras estratégias, destacando seus pontos positivos e negativos.

5.2.1 Beacons Gerados para Grupos Geocast do Protocolo APGP

A Figura 5.3 apresenta o número de *beacons* gerados no protocolo APGP para cada grupo *geocast* durante o tempo de simulação. Segundo o componente de controle de congestão de *beacons* do protocolo APGP, Seção 4.3, um *beacon* sempre será enviado para um grupo *geocast* G_i, quando o seu Erro de Predição Acumulado e_i ultrapassar o Erro de Posicionamento Máximo δ. Como os valores de δ_1 e δ_2 são iguais a $1,0$ m e $1,5$ m, como

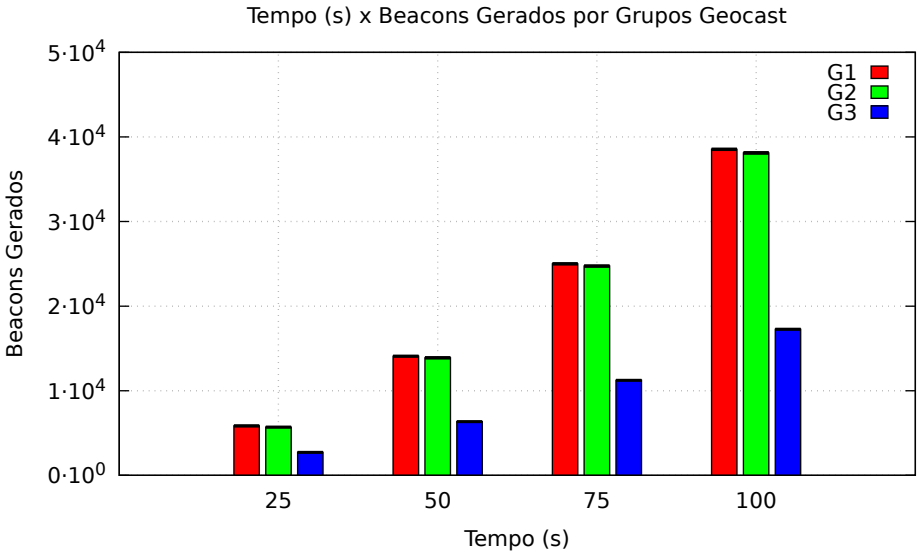

Figura 5.3: *Beacons* foram gerados mais frequência para os grupos *geocast* G1 e G2 em comparação ao grupo G3 em todos os testes.

indicado na Tabela 5.2, o erro de posicionamento oscila de forma que o quantitativo de *beacons* dentre os grupos G_1 e G_2 se torna quase igual como indicado no gráfico.

No instante onde o tempo é igual a 100 s, notou-se que os veículos enviaram cerca de 40 mil *beacons* para os grupos G_1 e G_2, e aproximadamente 18 mil *beacons* para o grupo G_3. Sendo assim, pode-se concluir que os grupos G_1 e G_2 receberão atualizações com mais frequência, pois são formados pelos vizinhos que estão mais próximos do veículo de origem. Os veículos vizinhos em G_3 estão mais distantes e receberão atualizações com menos frequência.

5.2.2 Média de Vizinhos nos Grupos Geocast do Protocolo APGP

Na lista de vizinhos L de cada veículo é possível inferir o número de membros em cada grupo *geocast*, como indicado no componente de monitoramento de vizinhos da Seção 4.4. Esse valor é medido periodicamente e uma média é realizada no final da simulação. O número de vizinhos em cada grupo *geocast* é influenciado pelo Fator de Escala de Distância ϵ e as Distâncias de Borda Δ calculadas previamente e demonstradas na Tabela 5.2.

Segundo a Figura 5.4, aos 100 s de simulação, os vizinhos presentes no grupo G_1 e G_2 somam em torno de 28 veículos, sendo aqueles que receberão atualizações com mais frequência. O grupo G_3 chega a aproximadamente 30 veículos. A Distância de Borda dos veículos em G_1 é de 100 m, indicando que todos os veículos nesse alcance receberão *beacons* com mais frequência. Os veículos em G_2 estão entre 100 m e 150 m de distância, portanto tem uma área menor a ser coberta, o que pode ser visto no gráfico. Como a

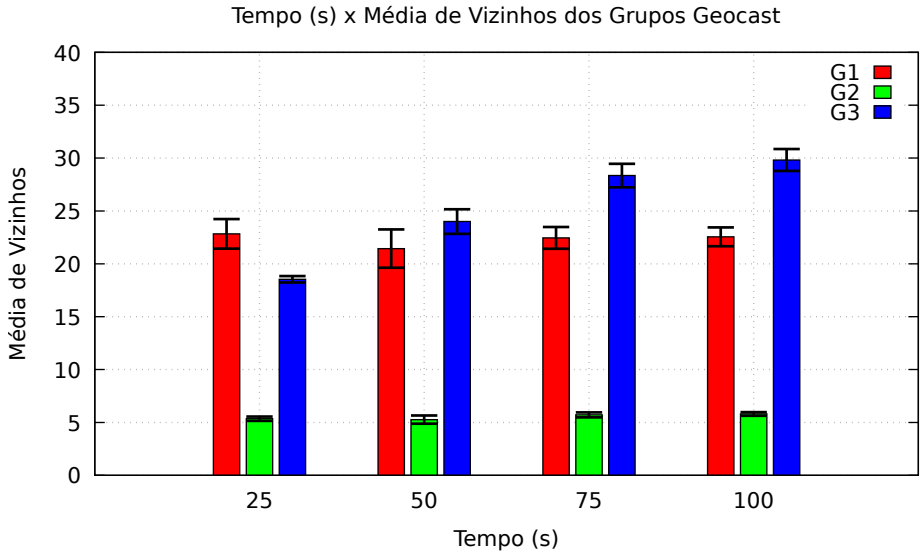

Figura 5.4: Os grupos G_1 e G_2 apresentam menos vizinhos, enquanto os vizinhos em G_3 aumentam conforme o tempo.

distância do grupo G_3 engloba veículos entre 150 m até 500 m de distância, espera-se que quanto maior for o tempo de simulação e mais veículos entram na simulação, maior seja o seu número de vizinhos.

O número reduzido de veículos em G_1 e G_2 é essencial para o bom funcionamento do protocolo APGP, já que mais *beacons* serão enviados para esses grupos, como visto na Figura 5.3. Com isso, torna-se possível garantir que o número de *beacons* sejam enviados para em torno de 28 veículos, o que se assemelha aos limites máximos discutidos no Capítulo 4, que trata do problema e das limitações do canal.

5.2.3 Beacons Gerados

A Figura 5.5 apresenta os *beacons* gerados durante a simulação pelos protocolos APGP, DC-BTRP e o protocolo baseado no 802.11p. Como esperado, o protocolo baseado no 802.11p é o que gera mais *beacons*, pois ele indica que essas mensagens devem ser enviadas constantemente a uma taxa fixa de 10 Hz. Conforme o tempo, mais veículos ingressam na simulação, o que é capturado pelo comportamento do gráfico, já que mais *beacons* são gerados pelo 802.11p.

Os protocolos APGP e DC-BTRP incluem mecanismos de ajuste de potência, permitindo que o veículo ajuste o raio de transmissão que as mensagens vão alcançar. Portanto, mesmo o número de *beacons* aumentando, não necessariamente significará que o canal se tornará congestionado. Quando se compara a proposta APGP ao DC-BTRP e 802.11p, pode-se observar que aos 100 s de simulação foram gerados aproximadamente 26.3% e

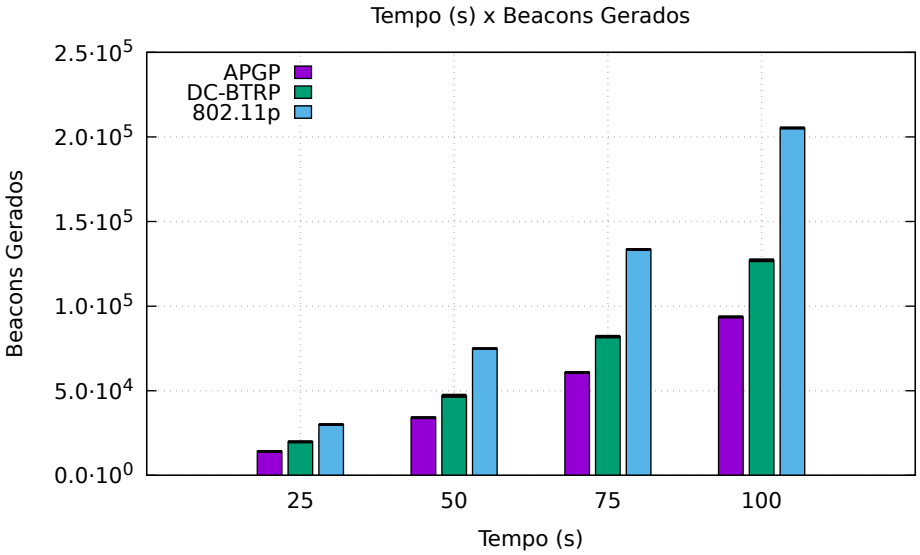

Figura 5.5: Quantitativo de *beacons* gerados durante a simulação dos três protocolos usados para comparação.

54.4% menos *beacons* respectivamente. Entretanto, esse resultado não é suficiente para provar a eficácia do protocolo, pois também é necessário checar se a acurácia de posicionamento é mantida.

5.2.4 Taxa de Recepção de Pacotes

A Figura 5.6 apresenta a Taxa de Recepção de Pacotes ou T_{RP}, que pode ser medida conforme a Eq. 5.2 considerando os quantitativos de pacotes recebidos e perdidos por colisões e erros durante a simulação.

O valor de T_{RP} é essencial para demonstrar o bom funcionamento de um protocolo, pois indica se os pacotes enviados estão sendo recebidos corretamente e sem existência de colisões e erros. O alto número de *beacons* gerados pelo protocolo baseado no 802.11p, ocasiona também em um alto número de pacotes perdidos, como pode ser visto na Figura 5.6, onde este valor oscila de aproximadamente 53% até 38.4% aos 100 s de simulação. O mesmo comportamento pode ser observado no protocolo DC-BTRP e no protocolo APGP, porém a diminuição da recepção de pacotes é de 76.6% para 58% no DC-BTRP e de aproximadamente 96.8% para 93.4% no protocolo APGP.

Considerando ainda os resultados apresentados na Figura 5.6, o protocolo APGP consegue garantir que o valor de T_{RP} se mantenha acima de 90% mesmo durante os 100 s de simulação, onde existem mais veículos enviando *beacons*. Esse comportamento é explicado pela estratégia de envio para os grupos *geocast*, onde um número reduzido de veículos receberá *beacons* com mais frequência, quando comparado ao grupo *geocast* com

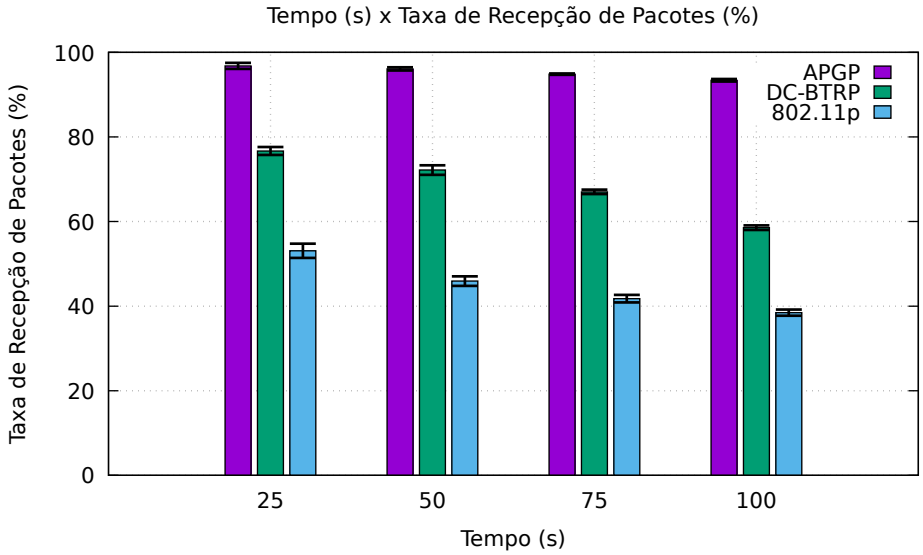

Figura 5.6: A taxa de recepção de pacotes do protocolo APGP apresenta a menor queda durante o tempo quando comparada aos protocolos DC-BTRP e 802.11p.

mais veículos. No protocolo baseado em 802.11p esta distinção não é efetuada, enquanto no DC-BTRP ela é realizada considerando apenas a carga observada no canal.

5.2.5 Taxa de Ocupação do Canal

A Figura 5.7 apresenta a Taxa de Ocupação do Canal ou T_{OC}, que pode ser medida conforme a Eq. 5.3 considerando o tempo de ocupação observado por cada veículo. O resultado demonstra a eficácia do mecanismo de ajuste de potência do APGP e como ele lida com o requisito RQ1.

RQ1: Diminuir a taxa de ocupação do canal com uma estratégia de ajuste de potência consciente, ou seja, que garante que todos os veículos dentro do raio de transmissão estabelecido recebam mensagens.

Como discutido no Capítulo 4, em VANETs, o canal de transmissão apresenta limitações pelo curto período de sincronização que é divido entre os canais de controle CCH e os canais de serviço SCH. Os *beacons* e mensagens de alerta são enviados pelo canal de controle CCH durante um curto intervalo de sincronização de 50 ms. Como pode ser visto na Figura 5.7, o alto número de *beacons* e mensagens gerados pelo protocolo baseado em 802.11p resulta em uma alta ocupação do canal, podendo chegar a aproximadamente 80% aos 100 s de simulação.

O protocolo DC-BTRP oscila de aproximadamente 20% até pouco mais de 40%, como indicado na Figura 5.7, enquanto o protocolo APGP chega a cerca de 8% aos 100 s de simulação. Conforme a descrição do problema no Capítulo 4, o canal pode ser considerado

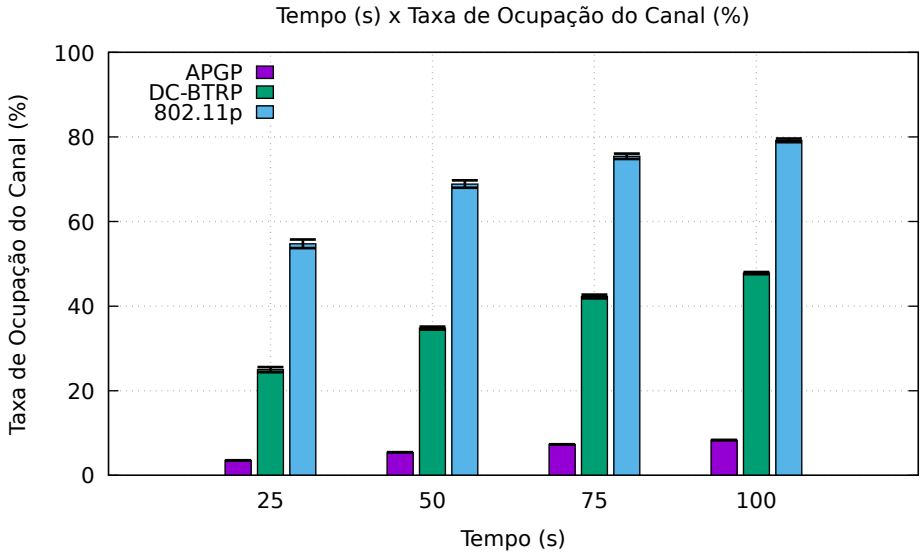

Figura 5.7: O mecanismo de ajuste de potência permite controlar a taxa de ocupação do canal observada pelos veículos nos protocolos APGP e DC-BTRP.

saturado quando este valor ultrapassa 60%, o que é observado no 802.11p após os 50 s de simulação. O APGP e DC-BTRP conseguem permanecer abaixo desse limiar, enquanto o APGP apresenta melhorias de aproximadamente 83% em relação ao DC-BTRP.

Esse resultado pode ser implicado pelo fato do protocolo utilizar o ajuste de potência com os valores indicados na Tabela 4.1. Enquanto o protocolo DC-BTRP também ajusta a potência, mas mantém esse valor em níveis mais altos considerando que a carga do canal não é alta o suficiente para ser necessário reduzir a potência. O APGP define valores fixos de potência, que sempre serão usados para enviar seus *beacons* para os grupos *geocast*. No caso dos protocolos DC-BTRP e 802.11p não existe uma distinção dos vizinhos que deverão receber um *beacon*, explicando o valor mais alto de ocupação do canal.

5.2.6 Taxa de Atraso de Entrega

A Figura 5.8 apresenta a Taxa de Atraso de Entrega ou T_{AE}, que foi observada pelos veículos durante as simulações. Este valor pode ser medido conforme a Eq. 5.4. O resultado demonstra como o protocolo consegue atender ao requisito RQ4 definido no Capítulo 1.

RQ4: Apresentar um mecanismo confiável e com baixo atraso para a transmissão das mensagens de alerta sobre eventos críticos.

Durante a simulação, um veículo gera uma mensagem alerta que deve ser repassada para todos os veículos em uma Zona de Interesse (ZdI). O protocolo APGP utiliza o componente de envio de mensagens de alerta descrito na Seção 4.5 para realizar essa

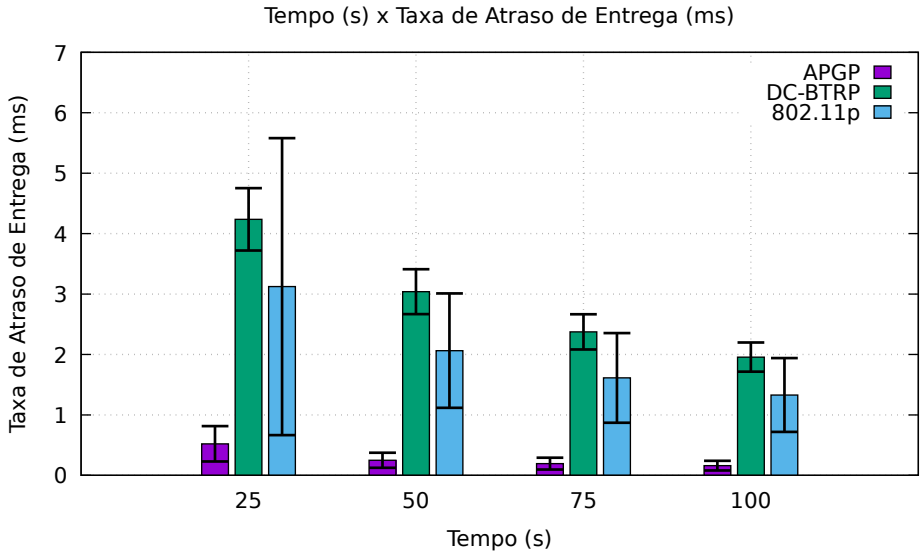

Figura 5.8: A taxa de atrasdo de entrega foi reduzida consideravelmente para o protocolo APGP, enquanto o DC-BTRP apresentou resultados abaixo do 802.11p.

tarefa, enquanto o protocolo DC-BTRP e 802.11p utilizam outras estratégias para realizar o repasse. A taxa de atraso de entrega é uma média do tempo que uma mensagem de alerta leva para chegar a até um veículo dentro da ZdI.

Assim como visto nos trabalhos relacionados analisados no Capítulo 3, quanto mais veículos na simulação, mais rápido o repasse da mensagem acontece. Esse resultado também pode ser visto nos resultados da Figura 5.8, onde o atraso inicialmente pode chegar a até aproximadamente 6 ms para o 802.11p no intervalo de confiança e oscila para em torno de 2 ms. O protocolo APGP reduz drasticamente o atraso, pois realiza a escolha de nós encaminhadores para cobrir a ZdI, enquanto também apresenta um mecanismo de retransmissão em caso de erro. Os veículos em média conseguem receber as mensagens de alerta 87% e 83% que o DC-BTRP e o 802.11p respectivamente. O fato do protocolo DC-BTRP ser menos efetivo que o 802.11p pode ser explicado pelo mecanismo de ajuste de potência utilizado, enquanto o 802.11p utiliza a potência máxima para realizar o envio da mensagem de alerta.

5.2.7 Erro de Posicionamento Médio

O erro de posicionamento médio é uma medida da posição real de um veículo vizinho e a posição estimada presente em sua lista de vizinhos L, ele pode ser calculado segundo o Algoritmo 2. A tabela 5.4 apresenta o erro de posicionamento para cada grupo de *geocast* no protocolo APGP e no protocolo DC-BTRP, pois apenas essas duas abordagens

usam uma lista de vizinhos. Resultado que demonstra a eficácia do APGP ao atender o requisito RQ2 definido no Capítulo 1.

RQ2: Propor mecanismos que possibilitem alterar a taxa de envio de *beacons*, sem perda da acurácia de posicionamento, considerando uma predição de posição do veículo.

Tabela 5.4: Erro de posicionamento médio dos grupos *geocast* do protocolo APGP e o protocolo DC-BTRP.

Tempo (s)	APGP - G_1 (m)	APGP - G_2 (m)	APGP - G_3 (m)	DC-BTRP (m)
25	$0,91 \pm 0,008$	$1,38 \pm 0,005$	$4,81 \pm 0,026$	$2,67 \pm 0,030$
50	$0,90 \pm 0,002$	$1,39 \pm 0,001$	$4,83 \pm 0,048$	$2,74 \pm 0,024$
75	$0,89 \pm 0,001$	$1,38 \pm 0,003$	$4,82 \pm 0,015$	$2,88 \pm 0,022$
100	$0,89 \pm 0,002$	$1,38 \pm 0,002$	$4,84 \pm 0,005$	$3,06 \pm 0,023$

Para o grupo geocast G_1, o principal objetivo era manter o erro de posicionamento abaixo do valor máximo de $\delta_1 = 1$ m, conforme definido na Tabela 4.1, onde os veículos estão localizados a não mais de 100 m do veículo de origem. O resultado esperado é confirmado na Tabela 5.4, onde pode-se notar que para todos os cenários, o erro de posicionamento se mantém próximo a $0,89$ m. O mesmo comportamento era esperado dos grupos G_2 e G_3, onde pode-se observar que os veículos conseguem manter erro de posicionamento menor que $\delta_2 = 1,5$ e $\delta_3 = 5$ m, para veículos localizados numa distância de até 150 m e até 500 m. Com isso, pode-se concluir que o protocolo APGP consegue satisfazer localmente os requisitos de posicionamento das aplicações de segurança descritos na Seção 4.1.2.

Em média, o protocolo DC-BTRP apresenta um erro de posicionamento médio por volta de $2,67$ m aos 25 s de simulação, oscilando para em torno de $3,06$ s posteriormente. Esse resultado pode ser explicado pelo fato do protocolo não utilizar um mecanismo de predição em casos onde um *beacon* não é enviado, tornando a lista de vizinhos L desatualizada. Entretanto, quando comparado aos resultados para veículos mais distantes, o DC-BTRP se sobressai, pois a posição dos *beacons* é mais acurada do que as dos veículos dentro do grupo G_3.

5.3 Discussão

Neste capítulo, apresentou-se a metodologia utilizada para avaliar o funcionamento deste trabalho. Citou-se o ambiente de simulação utilizado, as métricas de avaliação, bem quanto os protocolos utilizados para comparação. Os resultados experimentais foram apresentados por meio de gráficos e tabelas. Considerando a análise feita nesse capítulo, pode-se afirmar que o protocolo APGP conseguiu, com êxito, garantir que os requisitos de acurácia de posicionamento das aplicações de segurança sejam atendidos localmente para

grupos de vizinhos chamados de grupos *geocast*. Além disso, o protocolo APGP conseguiu aumentar consideravelmente a taxa de recepção de pacotes e diminuir a taxa de ocupação do canal quando comparado com outras estratégias.

O protocolo é ideal para as aplicações de prevenção de acidentes que dependem da boa comunicação entre veículos mais próximos. Ele foi testado em um ambiente urbano, com objetivo de se aproximar ao máximo de um cenário veicular real. O APGP pode ser utilizado em outras situações, mas como visto nos resultados de acurácia de posicionamento, ele não consegue manter a acurácia para veículos muito distantes, como o protocolo DC-BTRP utilizado para comparação.

No próximo capítulo, apresenta-se uma síntese desse trabalho, onde serão destacadas as principais contribuições, conclusões e possíveis trabalhos futuros.

Capítulo 6

Considerações Finais

Esse capítulo apresenta uma síntese da pesquisa realizada nesse trabalho, definindo as principais contribuições, bem quanto as conclusões obtidas e perspectivas para trabalhos futuros.

6.1 Contribuições

Os estudos sobre Rede Veicular Ad-Hoc (do Inglês, *Vehicular Ad-Hoc Network*) (VANET) são um dos pontos principais dos Sistemas de Transporte Inteligente (do Inglês, *Intelligent Transportation Systems*) (ITS) e apresentam formas de melhorar a segurança de motoristas, passageiros e pedestres no ambiente veicular. Protocolos e algoritmos têm sido propostos na literatura para resolver o problema de congestão em VANETs. Enquanto a maioria dos trabalhos se preocupa com garantir melhores condições para o canal de transmissão, poucos discutem como a acurácia de posicionamento afeta o funcionamento das aplicações de prevenção de acidentes.

Por conseguinte, esse trabalho objetivou propor o protocolo *Accurate Positioning Geocast Protocol* (APGP) para atender os requisitos de posicionamento das aplicações, evitando também a congestão no canal de transmissão. Nesse sentido, adotou-se uma arquitetura de três componentes para lidar com a congestão causada pelo envio de *beacons*, o monitoramento de vizinhos e o envio de mensagens de alerta para prevenção de acidentes.

O componente de controle de congestão de *beacons* teve como principal função controlar para quais grupos de vizinhos, chamados de grupos *geocast*, os *beacons* deveriam ser enviados. Para isso, cada veículo calculava periodicamente seu erro de predição acumulado localmente. Este valor era comparado com um erro máximo de posicionamento definido pelas aplicações. Toda vez que a predição falhava, um novo *beacon* era enviado para o grupo *geocast* correspondente ao erro excedido. A estratégia prioriza que veículos

mais próximos recebam atualizações mais frequentes e evita o envio de informações desnecessárias para veículos mais distantes. Podendo assim, diminuir a ocupação do canal.

As informações obtidas dos *beacons* foram utilizadas para a criação de uma lista de vizinhos, que monitorava o ID, posição geográfica, velocidade, grupo *geocast*, tempo desde o último envio e fator de estabilidade de cada veículo. Toda vez que um novo *beacon* era recebido de um vizinho, sua posição era atualizada, bem quanto o contador de estabilidade da conexão. Com isso, o veículo conseguia ter acesso aos seus vizinhos que seguiam uma rota semelhante. Um mecanismo para a manutenção da lista também foi proposto, onde veículos são removidos quando não se comunicam há muito tempo ou que estão fora do alcance.

Além disso, também foi proposto um componente para envio de mensagens de alerta com escolha de nós encaminhadores com base em fatores de distância e estabilidade da conexão. Os nós encaminhadores, ao receberem a mensagem, podem repassá-la até que ela cobrisse uma Zona de Interesse (ZdI), conscientizando mais veículos sobre um possível evento. Um mecanismo de retransmissão também foi proposto nos casos em que o repasse por um nó encaminhador falhou por conta de uma colisão ou erro durante a transmissão.

O protocolo apresentado neste trabalho evoluiu de outros trabalhos aceitos em conferências e revistas previamente. O protocolo DCAP [81] estudou os impactos da densidade de tráfego na congestão causada pelo envio de *beacons*. O protocolo COVANET [82] analisou os efeitos causados pelos modelos de propagação, bem quanto propôs melhorias na escolha de um nó encaminhador durante a disseminação de mensagens de alerta. O protocolo NCAP [83] deu continuidade aos estudos sobre o envio eficiente de *beacons* em conjunto com o envio de mensagens de alerta.

6.2 Conclusões

Um ambiente de simulação foi criado, considerando as tecnologias mais utilizadas por outros trabalhos na literatura. Simulações foram realizadas em um cenário urbano para a avaliação da implementação do protocolo APGP. Além disso, implementaram-se outros protocolos relacionados utilizados posteriormente como forma de comparação durante a análise dos resultados experimentais. Primeiramente, definiu-se um tempo de aquecimento para que veículos pudessem ingressar na simulação, passado esse tempo, os veículos podiam transmitir seus *beacons* e mensagens de alerta conforme a necessidade.

Os resultados experimentais iniciais mostraram que o protocolo APGP consegue efetivamente implementar a estratégia de grupos *geocast*. As Seções 5.2.1 e 5.2.2, trouxeram o número de *beacons* gerados para cada grupo, bem quanto uma média de vizinhos percebida pelos veículos durante a simulação. Foi possível notar que os veículos enviam mais

beacons para os grupos G_1 e G_2 devido aos valores de Erro Máximo de Posicionamento (δ) menores, como indicado na Tabela 4.1.

O desempenho do protocolo APGP está intrinsecamente relacionado com o Fator de Escala de Distância ϵ, que determina a abrangência de cada grupo *geocast*. Valores mais altos de ϵ aumentarão o número de veículos que receberão *beacons* com mais frequência, enquanto valores mais baixos reduzirão esse mesmo quantitativo. Procurou-se definir este valor com base nas limitações do canal de transmissão estudados no Capítulo 4, onde uma estimativa do número ideal de veículos foi realizada para transmissão. Os resultados apresentados na Seção 5.2.2 corroboram com as expectativas realizadas.

Após a definição dos parâmetros do protocolo APGP, ele foi comparado com outras estratégias. Sendo que uma delas se assemelha ao APGP por também lidar com a acurácia de posicionamento dos veículos. Métricas como o número de *beacons* gerados, taxa de recepção de pacotes e taxa de ocupação do canal foram analisadas, onde foi possível notar uma melhora considerável da proposta em relação aos outros protocolos.

Analisou-se também o comportamento do protocolo durante o envio de mensagens de alerta através da métrica de taxa de atraso de entrega, onde também foi possível notar que o APGP se sobressaiu no cenário analisado. Pode-se afirmar que a proposta é ideal para aplicações de prevenção de acidentes, quando comparada aos outros trabalhos. Entretanto, mais testes devem ser feitos para concluir que o protocolo proposto tenha o mesmo desempenho em outros cenários.

O foco principal do trabalho era garantir a acurácia de posicionamento dos veículos, que foi analisada com mais detalhes na Seção 5.2.7, onde se mediu o erro de posicionamento médio observado pelos veículos durante a simulação. Enquanto o protocolo APGP conseguiu manter a acurácia nos níveis desejados para veículos em diferentes distâncias, o protocolo DC-BTRP apresentou o mesmo valor em todos os casos. Isso demonstra a efetividade do APGP em cenários com veículos mais próximos, colocando em detrimento a acurácia observada pelos veículos mais distantes. Sendo a troca realizada neste trabalho para evitar a congestão.

6.3 Trabalhos Futuros

O protocolo APGP foi proposto principalmente para atender aos requisitos de funcionamento de aplicações de prevenção de acidentes em um ambiente urbano. Estudos futuros sobre a abordagem com uso de grupos *geocast* podem ser de interesse para evitar a congestão de redes, não limitadas apenas ao âmbito de redes veiculares.

Em relação à criação e manutenção dos grupos *geocast*, os parâmetros utilizados no protocolo APGP foram estáticos, considerando um estudo realizado sobre a limitação do

canal de transmissão em VANETs. Mesmo obtendo resultados favoráveis em um cenário urbano, não há garantia se esses parâmetros seriam ideias para todos outros cenários. Sendo assim, seria de interesse empregar uma técnica de ajuste dinâmico dos parâmetros dos grupos conforme as condições da pista, como número de faixas e velocidade máxima permitida.

Os componentes do protocolo APGP são independentes o suficiente, de forma que podem ser facilmente integrados a outros protocolos. Adicionar os componentes de congestão de *beacons* e monitoramento de vizinhos a outras estratégias de envio de mensagens de alerta já existentes pode trazer melhorias durante o roteamento. A abordagem com foco na acurácia de posicionamento dos componentes do APGP também tornaria essas estratégias ideias para as aplicações de prevenção de acidentes.

Por fim, melhorias no modo como as mensagens de alerta são enviadas e repassadas no protocolo APGP podem ser realizadas. O componente responsável por essa tarefa define a escolha de encaminhadores com base em fatores como a estabilidade da conexão e a distância. Entretanto, fatores como a atenuação do sinal não são considerados, o que pode ser explorado em trabalhos futuros para melhorias durante o processo de encaminhamento da mensagem.

Referências

[1] *IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*. https://doi.org/10.1109/ieeestd.2005.97890. vi, vii, 1, 2, 6, 7, 8, 9, 41, 46, 58, 59, 66

[2] Bolufe, Sandy, Samuel Montejo-Sanchez, Cesar A. Azurdia-Meza, Sandra Cespedes, Richard Demo Souza e Evelio M. G. Fernandez: *Dynamic control of beacon transmission rate and power with position error constraint in cooperative vehicular networks*. Em *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM, abril 2018. https://doi.org/10.1145/3167132.3167356. vi, vii, 17, 24, 26, 36, 37, 39, 40, 66

[3] Marroquin, Alberto, Marco Antonio To, Cesar A. Azurdia-Meza e Sandy Bolufe: *A General Overview of Vehicle-to-X (V2X) Beacon-Based Cooperative Vehicular Networks*. Em *2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)*. IEEE, nov 2019. https://doi.org/10.1109%2Fconcapanxxxix47272.2019.8977034. xi, 9

[4] Guerrero-Ibanez, Antonio, Carlos Flores-Cortes, Pedro Damian-Reyes, M. Andrade-Arechiga e J. R. G. Pulido: *Emerging Technologies for Urban Traffic Management*. Em *Urban Development*. InTech, mar 2012. https://doi.org/10.5772%2F37760. xi, 10

[5] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-Channel Operation*. https://doi.org/10.1109/ieeestd.2016.7435228. xi, 3, 9, 10, 11, 40, 43

[6] Shabir, Balawal, Muazzam A. Khan, Anis U. Rahman, Asad W. Malik e Abdul Wahid: *Congestion Avoidance in Vehicular Networks: A Contemporary Survey*. IEEE Access, 7:173196–173215, 2019. https://doi.org/10.1109/access.2019.2955142. xi, 15, 18, 26, 59

[7] Ren, Mengying, Jun Zhang, Lyes Khoukhi, Houda Labiod e Veronique Veque: *A Unified Framework of Clustering Approach in Vehicular Ad Hoc Networks*. IEEE Transactions on Intelligent Transportation Systems, 19(5):1401–1414, maio 2018. https://doi.org/10.1109/tits.2017.2727226. xi, 18, 28, 32, 33, 35, 36, 37, 40

[8] Qi, Weijing, Qingyang Song, Xiaojie Wang, Lei Guo e Zhaolong Ning: *SDN-Enabled Social-Aware Clustering in 5G-VANET Systems*. IEEE Access, 6:28213–28224, 2018. https://doi.org/10.1109/access.2018.2837870. xi, 18, 28, 34, 35, 36, 37, 40

[9] *Vehicle Safety Communications – Applications (VSC-A) - Final Report*, 2011. https://www.nhtsa.gov/sites/nhtsa.gov/files/811492a.pdf, Acessado em 18/08/2023. xiii, 43, 48

[10] Hasan, Syed: *Intelligent transportation systems : 802.11-based vehicular communications*. Springer, Cham, Switzerland, 2018, ISBN 978-3-319-64057-0. 1, 6

[11] Kenney, John B.: *Dedicated Short-Range Communications (DSRC) Standards in the United States*. Proceedings of the IEEE, 99(7):1162–1182, julho 2011. https://doi.org/10.1109/jproc.2011.2132790. 1, 6

[12] Lu, Zhaojun, Gang Qu e Zhenglin Liu: *A Survey on Recent Advances in Vehicular Network Security, Trust, and Privacy*. IEEE Transactions on Intelligent Transportation Systems, 20(2):760–776, fevereiro 2019. https://doi.org/10.1109/tits.2018.2818888. 1

[13] Pinto Neto, João B., Lucas C. Gomes, Fernando M. Ortiz, Thales T. Almeida, Miguel Elias M. Campista, Luís Henrique M.K. Costa e Nathalie Mitton: *An accurate cooperative positioning system for vehicular safety applications*. Computers & Electrical Engineering, 83:106591, 2020, ISSN 0045-7906. https://www.sciencedirect.com/science/article/pii/S0045790618326776. 1, 43, 47

[14] Miller, Jeffrey: *Vehicle-to-vehicle-to-infrastructure (V2V2I) intelligent transportation system architecture*. Em *2008 IEEE Intelligent Vehicles Symposium*. IEEE, jun 2008. https://doi.org/10.1109%2Fivs.2008.4621301. 2, 7

[15] *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, março 2016. https://www.sae.org/standards/content/j2735_201603/. 2

[16] Lyu, Feng, Nan Cheng, Haibo Zhou, Wenchao Xu, Weisen Shi, Jiayin Chen e Minglu Li: *DBCC: Leveraging Link Perception for Distributed Beacon Congestion Control in VANETs*. IEEE Internet of Things Journal, 5(6):4237–4249, dezembro 2018. https://doi.org/10.1109/jiot.2018.2844826. 2, 42, 44

[17] Zemouri, Sofiane, Soufiene Djahel e John Murphy: *An Altruistic Prediction-Based Congestion Control for Strict Beaconing Requirements in Urban VANETs*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, páginas 1–16, 2018. https://doi.org/10.1109/tsmc.2017.2759341. 3, 17

[18] *IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. https://doi.org/10.1109/ieeestd.2016.7786995. 3, 8, 43

[19] Feukeu, E.A. e T. Zuva: *Dynamic Broadcast Storm Mitigation Approach for VANETs*. Future Generation Computer Systems, 107:1097–1104, junho 2020. https://doi.org/10.1016/j.future.2017.12.049. 3, 21, 23, 36, 37

[20] Li, Fei e Chuanhe Huang: *A Mobility Prediction Based Beacon Rate Adaptation Scheme in VANETs*. Em *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, junho 2018. https://doi.org/10.1109/iscc.2018.8538734. 3, 17, 19, 22, 23, 36, 37, 38, 39, 40

[21] Cho, Byeong Moon, Min Seong Jang e Kyung Joon Park: *Channel-Aware Congestion Control in Vehicular Cyber-Physical Systems*. IEEE Access, 8:73193–73203, 2020. https://doi.org/10.1109/access.2020.2987416. 3, 17, 25, 26, 36, 37, 38, 46

[22] Joseph, Maan, Xiaofeng Liu e Arunita Jaekel: *An Adaptive Power Level Control Algorithm for DSRC Congestion Control*. Em *Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications - DIVANet'18*. ACM Press, 2018. https://doi.org/10.1145/3272036.3272041. 3, 17, 19, 21, 36, 37, 39, 46

[23] Bi, Yuanguo, Hangguan Shan, Xuemin Sherman Shen, Ning Wang e Hai Zhao: *A Multi-Hop Broadcast Protocol for Emergency Message Dissemination in Urban Vehicular Ad Hoc Networks*. IEEE Transactions on Intelligent Transportation Systems, 17(3):736–750, março 2016. https://doi.org/10.1109/tits.2015.2481486. 3, 18, 29, 31, 36, 37, 38, 39

[24] Tian, Daxin, Chao Liu, Xuting Duan, Zhengguo Sheng, Qiang Ni, Min Chen e Victor C. M. Leung: *A Distributed Position-Based Protocol for Emergency Messages Broadcasting in Vehicular Ad Hoc Networks*. IEEE Internet of Things Journal, 5(2):1218–1227, abril 2018. https://doi.org/10.1109/jiot.2018.2791627. 3, 18, 28, 30, 31, 36, 37, 38, 39

[25] Hawbani, Ammar, Xingfu Wang, Ahmed Al-Dubai, Liang Zhao, Omar Busaileh, Ping Liu e Mohammed A. A. Al-Qaness: *A Novel Heuristic Data Routing for Urban Vehicular Ad Hoc Networks*. IEEE Internet of Things Journal, 8(11):8976–8989, junho 2021. https://doi.org/10.1109/jiot.2021.3055504. 3, 18, 28, 32, 36, 37, 38, 39

[26] Han, Ruiyan, Jinglun Shi, Quansheng Guan, Farhad Banoori e Weiqiang Shen: *Speed and Position Aware Dynamic Routing for Emergency Message Dissemination in VANETs*. IEEE Access, 10:1376–1385, 2022. https://doi.org/10.1109/access.2021.3138960. 3, 18, 28, 31, 36, 37, 38, 39

[27] V1.3.1, ETSI EN 302 663: *Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems Operating in the 5 GHz Frequency Band*, 2019. https://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.03.01_30/en_302663v010301v.pdf, Acessado em 18/08/2023. 7

[28] *IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture*. https://doi.org/10.1109/ieeestd.2014.6755433. 8

[29] Jiang, Daniel e Luca Delgrossi: *IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments*. Em *VTC Spring 2008 - IEEE Vehicular Technology Conference*. IEEE, maio 2008. https://doi.org/10.1109/vetecs.2008.458. 8, 9

[30] *Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Resource Manager*. https://doi.org/10.1109/ieeestd.2006.246485. 8, 11

[31] *IEEE Standard for Wireless Access in Vehicular Environments–Security Services for Applications and Management Messages - Amendment 1*. https://doi.org/10.1109/ieeestd.2017.8065169. 8, 11

[32] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Services*. https://doi.org/10.1109/ieeestd.2016.7458115. 9, 11

[33] Fu, Zhenghua, Xiaoqiao Meng e Songwu Lu: *How bad TCP can perform in mobile ad hoc networks*. Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications, páginas 298–303, 2002. 11

[34] Cunha, Felipe e Leandro Villas: *Data communication in VANETs: Protocols, applications and challenges*. Ad Hoc Networks, 44:90–103, 2016, ISSN 1570-8705. https://www.sciencedirect.com/science/article/pii/S1570870516300580. 12

[35] Yang, Yang e Kun Hua: *Emerging Technologies for 5G-Enabled Vehicular Networks*. IEEE Access, 7:181117–181141, 2019. https://doi.org/10.1109%2Faccess.2019.2954466. 13

[36] Haykin, S.: *Cognitive radio: brain-empowered wireless communications*. IEEE Journal on Selected Areas in Communications, 23(2):201–220, feb 2005. https://doi.org/10.1109%2Fjsac.2004.839380. 13

[37] Filippi, Alessio, Kees Moerman, Vincent Martinez e Onn Haran: *IEEE 802.11p ahead of LTE-V2V for safety applications*, 2017. https://www.semanticscholar.org/paper/IEEE802.11p-ahead-of-LTE-V2V-for-safety-Filippi-Moerman/1183f80c50fd2986615bea2e07a1f0d342a31abc, Acessado em 18/08/2023. 13

[38] Martinez, Francisco, Chai Toh, Juan Carlos Cano, Carlos Calafate e Pietro Manzoni: *A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)*. Wireless Communications and Mobile Computing, 11:813 – 828, julho 2011. 13

[39] Krajzewicz, Daniel, Jakob Erdmann, Michael Behrisch e Laura Bieker: *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5(3&4):128–138, dezembro 2012. https://elib.dlr.de/80483/. 14, 59

[40] Lan, Kun Chan: *MOVE*. Em *Telematics Communication Technologies and Vehicular Networks*, páginas 355–368. IGI Global, 2010. https://doi.org/10.4018/978-1-60566-840-6.ch021. 14

[41] Härri, Jérôme, Marco Fiore, Fethi Filali e Christian Bonnet: *Vehicular mobility simulation with VanetMobiSim*. SIMULATION, 87(4):275–300, setembro 2009. https://doi.org/10.1177/0037549709345997. 14

[42] Cameron, Gordon D. B. e Gordon I. D. Duncan: *PARAMICS Parallel microscopic simulation of road traffic*. The Journal of Supercomputing, 10(1):25–53, 1996. https://doi.org/10.1007/bf00128098. 14

[43] *PTV Vissim new — ptvgroup.com*. https://www.ptvgroup.com/en/solutions/products/ptv-vissim/. Acessado em 18/08/2023. 14

[44] Gunes, Mesut, Felix Juraschek, Bastian Blywis e Christian Graff: *MoNoTrac A mobility trace generator based on OpenStreetMap geo-data*. Em *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*. IEEE, novembro 2010. https://doi.org/10.1109/mass.2010.5663869. 14

[45] *NS-2 Network Simulator*. http://nsnam.sourceforge.net/wiki/index.php/Main_Page, Acessado em 18/08/2023. 15

[46] *NS-3 Network Simulator*. https://www.nsnam.org/, Acessado em 18/08/2023. 15

[47] *OMNeT++ Discrete Event Simulator*. https://omnetpp.org. Acessado em 18/08/2023. 15, 59

[48] Ahmed, Bilal, Asad Waqar Malik, Taimur Hafeez e Nadeem Ahmed: *Services and simulation frameworks for vehicular cloud computing: a contemporary survey*. EURASIP Journal on Wireless Communications and Networking, 2019(1), janeiro 2019. https://doi.org/10.1186/s13638-018-1315-y. 15

[49] *OPNET Network Simulator*, Apr 2020. https://opnetprojects.com/opnet-network-simulator/, Acessado em 18/08/2023. 15

[50] Sommer, C, R German e F Dressler: *Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis*. IEEE Transactions on Mobile Computing, 10(1):3–15, janeiro 2011. https://doi.org/10.1109/tmc.2010.133. 16, 59, 60, 61

[51] Amoozadeh, Mani, Hui Deng, Chen Nee Chuah, H. Michael Zhang e Dipak Ghosal: *Platoon management with cooperative adaptive cruise control enabled by VANET*. Vehicular Communications, 2(2):110–123, abril 2015. https://doi.org/10.1016/j.vehcom.2015.03.004. 16

[52] Zarrad, Anis e Izzat Alsmadi: *Evaluating network test scenarios for network simulators systems*. International Journal of Distributed Sensor Networks, 13(10):155014771773821, outubro 2017. https://doi.org/10.1177/1550147717738216. 16

[53] Mussa, Sofian Ali Ben, Mazani Manaf, Kayhan Zrar Ghafoor e Zouina Doukha: *Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives*. Em *2015 International Conference on Wireless Networks and Mobile Communications (WINCOM)*. IEEE, outubro 2015. https://doi.org/10.1109/wincom.2015.7381319. 16

[54] Sommer, Christoph, Ozan Tonguz e Falko Dressler: *Traffic information systems: efficient message dissemination via adaptive beaconing*. IEEE Communications Magazine, 49(5):173–179, maio 2011. https://doi.org/10.1109/mcom.2011.5762815. 17, 19

[55] Sommer, Christoph, Stefan Joerer, Michele Segata, Ozan K. Tonguz, Renato Lo Cigno e Falko Dressler: *How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help*. IEEE Transactions on Mobile Computing, 14(7):1411–1421, julho 2015. https://doi.org/10.1109/tmc.2014.2362752. 17, 19, 42

[56] Torrent-Moreno, M., J. Mittag, P. Santi e H. Hartenstein: *Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information*. IEEE Transactions on Vehicular Technology, 58(7):3684–3703, setembro 2009. https://doi.org/10.1109/tvt.2009.2017545. 17, 19

[57] Kloiber, Bernhard, Jerome Harri e Thomas Strang: *Dice the TX power — Improving Awareness Quality in VANETs by random transmit power*. Em *2012 IEEE Vehicular Networking Conference (VNC)*. IEEE, novembro 2012. https://doi.org/10.1109/vnc.2012.6407445. 17, 19

[58] Qiao, Yu, Xiaohui Hu e Le Cao: *Transmission power adaptive congestion control algorithm based on Bayesian network*. Em *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence*. ACM, outubro 2020. https://doi.org/10.1145/3438872.3439067. 17, 20, 36, 37, 38, 39

[59] Shah, Syed Adeel Ali, Ejaz Ahmed, Joel J. P. C. Rodrigues, Ihsan Ali e Rafidah Md Noor: *Shapely Value Perspective on Adapting Transmit Power for Periodic Vehicular Communications*. IEEE Transactions on Intelligent Transportation Systems, 19(3):977–986, março 2018. https://doi.org/10.1109/tits.2017.2775965. 17, 20, 36, 37, 38

[60] Rossi, Giorgia V. e Kin K. Leung: *Optimised CSMA/CA protocol for safety messages in vehicular ad-hoc networks*. Em *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, julho 2017. https://doi.org/10.1109/iscc.2017.8024608. 18, 19, 26, 28, 36, 37, 39

[61] Lu, Yanfei, Jianmin Ren, Jin Qian, Meng Han, Yan Huo e Tao Jing: *Predictive Contention Window-Based Broadcast Collision Mitigation Strategy for VANET*. Em *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*. IEEE, outubro 2016. https://doi.org/10.1109/bdcloud-socialcom-sustaincom.2016.41. 18, 19, 27, 28, 36, 37, 38, 39

[62] Huang, Ching Ling, Yaser Fallah, Raja Sengupta e Hariharan Krishnan: *Adaptive intervehicle communication control for cooperative safety systems*. IEEE Network, 24(1):6–13, janeiro 2010. https://doi.org/10.1109/mnet.2010.5395777. 19

[63] Jin, Y., Y. Hu, J. Zhang e J. Huang: *Bayesian network structure learning combining K2 with simulated annealing*. Dongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Southeast University (Natural Science Edition), 42:82–86, setembro 2012. 20

[64] Lipovetsky, Stan: *Handbook of the Shapley Value*. Technometrics, 62(2):1–280, abril 2020. https://doi.org/10.1080/00401706.2020.1744904. 20

[65] Gibbs, Bruce P: *Advanced Kalman filtering, least-squares and modeling*. John Wiley & Sons, Nashville, TN, março 2011. 22

[66] Killat, Moritz, Felix Schmidt-Eisenlohr, Hannes Hartenstein, Christian Rössel, Peter Vortisch, Silja Assenmacher e Fritz Busch: *Enabling efficient and accurate large-scale simulations of VANETs for vehicular traffic management*. Em *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks - VANET '07*. ACM Press, 2007. https://doi.org/10.1145/1287748.1287754. 25

[67] Eenennaam, Martijn van, Wouter Klein Wolterink, Georgios Karagiannis e Geert Heijenk: *Exploring the solution space of beaconing in VANETs*. Em *2009 IEEE Vehicular Networking Conference (VNC)*. IEEE, outubro 2009. https://doi.org/10.1109/vnc.2009.5416370. 25

[68] Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner e Lars Schmidt-Thieme: *BPR: Bayesian Personalized Ranking from Implicit Feedback*, 2012. https://arxiv.org/abs/1205.2618. 27

[69] Huh, Gio: *Enhanced Stochastic Gradient Descent with Backward Queried Data for Online Learning*. Em *2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*. IEEE, dezembro 2020. https://doi.org/10.1109/icmlant50963.2020.9355978. 28

[70] Korkmaz, GÖkhan, Eylem Ekici e FÜsun Ozguner: *Black-Burst-Based Multihop Broadcast Protocols for Vehicular Networks*. IEEE Transactions on Vehicular Technology, 56(5):3159–3167, setembro 2007. https://doi.org/10.1109/tvt.2007.900493. 29

[71] IEEE: *IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture*. IEEE Std 1609.0-2013, páginas 1–78, March 2014. 42

[72] Lyamin, Nikita, Alexey Vinel, Magnus Jonsson e Boris Bellalta: *Cooperative Awareness in VANETs: On ETSI EN 302 637-2 Performance*. IEEE Transactions on Vehicular Technology, 67(1):17–28, janeiro 2018. https://doi.org/10.1109/tvt.2017.2754584. 42

[73] Park, Yongtae e Hyogon Kim: *Application-Level Frequency Control of Periodic Safety Messages in the IEEE WAVE*. IEEE Transactions on Vehicular Technology, 61(4):1854–1862, 2012. 43

[74] Virdaus, Irvanda Kurniadi, Moonsoo Kang, Soekjoo Shin e Goo Rak Kwon: *A simulation study: Is the broadcast storming really harmful for emergency delivery in VANETs?* Em *2015 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, outubro 2015. https://doi.org/10.1109/atc.2015.7388415. 44

[75] Medhi, Deep e Karthik Ramasamy: *Chapter 8 - Multicast Routing*. Em Medhi, Deep e Karthik Ramasamy (editores): *Network Routing (Second Edition)*, The Morgan Kaufmann Series in Networking, páginas 260–285. Morgan Kaufmann, Boston, second edition edição, 2018, ISBN 978-0-12-800737-2. https://www.sciencedirect.com/science/article/pii/B9780128007372000107. 46

[76] Sommer, Christoph, Stefan Joerer e Falko Dressler: *On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation*. Em *4th IEEE Vehicular Networking Conference (VNC 2012)*, páginas 64–69, Seoul, Korea, November 2012. IEEE. 49, 61

[77] Palazzi, C. E., M. Roccetti e S. Ferretti: *An Intervehicular Communication Architecture for Safety and Entertainment*. IEEE Transactions on Intelligent Transportation Systems, 11(1):90–99, March 2010, ISSN 1524-9050. 57

[78] Ben Mussa, S. A., M. Manaf, K. Z. Ghafoor e Z. Doukha: *Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives*. Em *2015 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, páginas 1–8, Oct 2015. 59

[79] *OpenStreetMap*. https://www.openstreetmap.org/, Acessado em 18/08/2023. 60

[80] DLR-TS: *Publicly available traffic networks and corresponding demands for usage with Eclipse SUMO*. https://github.com/DLR-TS/sumo-scenarios, 2014. Acessado em 18/08/2023. 61

[81] Farias, Paulo Victor Goncalves, Jacir Luiz Bordim e Marcos Fagundes Caetano: *A Density-Based Congestion Avoidance Protocol for Strict Beaconing Requirements in VANETs*. Em *2019 Seventh International Symposium on Computing and Networking (CANDAR)*. IEEE, novembro 2019. https://doi.org/10.1109/candar.2019.00023. 76

[82] Lima, Tulio A., Paulo V. G. Farias e Jacir L. Bordim: *Mitigating the Effects of Multipath Propagation in Vehicular Ad Hoc Networks*. Em *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, junho 2020. https://doi.org/10.1109/iwcmc48107.2020.9148297. 76

[83] Farias, Paulo V. G., Tulio A. Lima e Jacir L. Bordim: *Mitigating message dissemination issues in safety applications for vehicular ad hoc networks*. Concurrency and Computation: Practice and Experience, 35(11), setembro 2020. https://doi.org/10.1002/cpe.6034. 76