



**PAULO HENRIQUE DE ALMEIDA**

**RODOGE: PROTOCOLO DE DISSEMINAÇÃO DE  
MENSAGENS DE ALERTA DE ACIDENTES COM  
CONTROLES E REGRAS DE REENVIO DE MENSAGENS**

**LAVRAS – MG**

**2018**

**PAULO HENRIQUE DE ALMEIDA**

**RODOGE: PROTOCOLO DE DISSEMINAÇÃO DE MENSAGENS DE ALERTA DE  
ACIDENTES COM CONTROLES E REGRAS DE REENVIO DE MENSAGENS**

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

Prof. PhD Luiz Henrique Andrade Correia  
Orientador

**LAVRAS – MG**  
**2018**

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca  
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Almeida, Paulo Henrique de.

RODOGE: Protocolo de Disseminação de Mensagens de Alerta  
de Acidentes com Controles e Regras de Reenvio de Mensagens /  
Paulo Henrique de Almeida. - 2018.

93 p. : il.

Orientador(a): Luiz Henrique Andrade Correia.

.  
Dissertação (mestrado acadêmico) - Universidade Federal de  
Lavras, 2018.

Bibliografia.

1. Rede Veicular. 2. Disseminação de Mensagens. 3. Broadcast  
Storm. I. Correia, Luiz Henrique Andrade. . II. Título.

**PAULO HENRIQUE DE ALMEIDA**

**RODOGE: PROTOCOLO DE DISSEMINAÇÃO DE MENSAGENS DE ALERTA DE  
ACIDENTES COM CONTROLES E REGRAS DE REENVIO DE MENSAGENS**

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

APROVADA em 19 de Outubro de 2018.

Prof. DSc. Cristiano Maciel da Silva DTECH UFSJ  
Prof. DSc. Renata Lopes Rosa DCC UFLA

Prof. PhD Luiz Henrique Andrade Correia  
Orientador

**LAVRAS – MG  
2018**

*Aos meus pais, Gilmar e Sônia, e ao meu irmão Rafael por todo o apoio, carinho e compreensão ao longo desta caminhada.*

## **AGRADECIMENTOS**

Agradeço a toda a minha família pelo carinho e apoio, aos meus pais e irmão pelo porto seguro, conselhos e incentivos, que não mediram esforços para que eu chegasse até esta etapa na caminhada da minha vida.

A Pamela, por ter ajudado com todos os conselhos, ombro e companheirismo nos momentos difíceis e todo apoio e carinho que poderia me proporcionar.

Ao professor Luiz Correia por toda a orientação, paciência, ensinamentos e auxílios no desenvolvimento e implementação do trabalho.

Aos meus companheiros de mestrado, que muito ajudaram na elaboração, resolução deste trabalho e nos estudos.

Aos meus amigos de São João Del Rei e de Lavras, que também participaram nesta caminhada e ajudaram a distrair nos momentos necessários.

Agradeço a Universidade Federal de Lavras, especialmente ao Programa de Pós-Graduação em Ciência da Computação, pela oportunidade. E a CAPES pela concessão da bolsa de mestrado.

Muito Obrigado!

*"O que sabemos é uma gota; o que ignoramos é um oceano."  
(Isaac Newton)*

## RESUMO

O avanço no desenvolvimento tecnológico na área de Redes Sem Fio, em especial as Redes de Celulares com a quinta geração (5G), possibilitou a comunicação em diferentes locais e propósitos, com altas taxas de dados, escalabilidade e baixa latência. Espera-se que com este desenvolvimento as comunicações de dispositivo a dispositivo (D2D) desempenhem papel fundamental, pois permitem a comunicação direta entre dispositivos com baixo grau de envolvimento das estações base. Uma das áreas de pesquisa que cresceu nos últimos anos devido a esses avanços na comunicação é a rede veicular, ou *Vehicle Ad hoc Networks* (VANET), na qual os veículos se comunicam com outros veículos e utilizam aplicações de segurança no trânsito. Entre as principais aplicações da VANET está a detecção e alerta de acidentes, que geralmente se baseia na transmissão de mensagens de alerta para outros veículos, bem como na transmissão desse alerta para dispositivos de infraestrutura. Cada veículo ou dispositivo de infraestrutura receptor também reenvia a mensagem de alerta, a fim de informar a situação a outros motoristas na região de interesse. O desafio nessas redes é reduzir a quantidade de mensagens reenviadas, que pode ocorrer de forma descontrolada e ocasionar na *broadcast storm*, que é inundação da rede por excesso de mensagens. Este trabalho propõe um protocolo para disseminação de mensagens de alerta com controle de reenvios chamado RODOGE - *Rules to prevent Over-Dissemination Of messaGEs*. As regras e funções aplicadas pela RODOGE limitam a disseminação de mensagens por veículos e dispositivos de infraestrutura, evitando a *broadcast storm*. O RODOGE foi testado e comparado com outros protocolos clássicos de disseminação de mensagens através de extensivas simulações usando OMNeT ++, simulador de rede de computadores, SUMO, simulador de tráfego veicular e Veins, *framework* que une esses simuladores. Dois cenários foram simulados com tráfego urbano e rodoviário, Grid e Colônia, com um e dois acidentes. Os resultados mostram que o RODOGE reduz a quantidade de mensagens geradas e recebidas, evitando a inundação da rede, mantendo uma área de cobertura de mensagens suficiente para permitir que os motoristas reajam a um acidente detectado.

**Palavras-chave:** *broadcast storm*, VANET, rede veicular, disseminação de mensagens, alerta de acidente.



## ABSTRACT

Advancement in technological development of Wireless Network, specially of Cellular Network with fifth generation (5G), allowed communication in different locations and purposes, with high data rates, scalability and low latency. Device-to-device (D2D) communications are expected to play a key role with this development, as they enable direct communication between devices with reduced degree of involvement from the base stations. One of the research areas that has grown in recent years due to these advances in communication is the vehicle network, or Vehicle Ad Hoc Networks (VANET), in which vehicles communicate with other vehicles and use traffic safety applications. Among VANET's main applications is accident detection and reporting, which usually relies on broadcasting alert messages to other vehicles, as well as transmitting this alert to infrastructure devices. Each receiving vehicle or infrastructure device also resends the alert message in order to inform the situation to other drivers in the region of interest. The challenge in these networks is to reduce the number of forwarded messages, which can occur uncontrollably, resulting in a broadcast storm, which is the flooding of the network by excessive messages. This paper proposes a protocol for the dissemination of alert messages with replication control called RODOGE (Rules to prevent Over-Dissemination of messaGEs). Rules and functions applied by RODOGE limit the dissemination of messages by vehicles and infrastructure devices, avoiding the broadcast storm. RODOGE has been tested and compared with other classic message dissemination protocols through extensive simulations using OM-NeT++, computer network simulator, SUMO, vehicular traffic simulator and Veins, framework that joins these simulators. Two scenarios were simulated with urban and highway traffic, Grid and Cologne, with one and two accidents. Results showed that RODOGE reduces the amount of messages generated and received, preventing network flooding, while maintaining sufficient message coverage area to allow drivers reaction to a detected accident.

**Keywords:** broadcast storm, VANET, vehicular network, message dissemination, accident alert

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 2.1 – Estabelecimento de uma VANETs. . . . .                                    | 17 |
| Figura 2.2 – Tipos de Comunicação. . . . .   | 18 |
| Figura 2.3 – Arquitetura da Rede Veicular - IEEE 1609. . . . .                         | 21 |
| Figura 2.4 – Forma de Disseminação de Mensagens . . . . .                              | 23 |
| Figura 2.5 – Disseminação por Broadcast em Redes Veiculares. . . . .                   | 25 |
| Figura 2.6 – Tipos de Protocolo <i>Broadcast</i> por Inundação Controlada . . . . .    | 27 |
| Figura 2.7 – Simulador OMNeT++. . . . .  | 29 |
| Figura 2.8 – Simulador Sumo. . . . .   | 30 |
| Figura 2.9 – Arquitetura Simulador Veins. . . . .                                      | 30 |
| Figura 2.10 – Simulador Veins. . . . .   | 31 |
| Figura 2.11 – Trecho do <i>Trace</i> de Cologne . . . . .                              | 32 |
| Figura 4.1 – Máquina de Estados do Veículo . . . . .                                   | 44 |
| Figura 4.2 – Máquina de Estados do Dispositivo de Infraestrutura . . . . .             | 44 |
| Figura 4.3 – Estrutura da Mensagem Enviada . . . . .                                   | 45 |
| Figura 4.4 – Estrutura da Tabela de Mensagens . . . . .                                | 46 |
| Figura 5.1 – Trecho de Cologne no SUMO . . . . .                                       | 56 |
| Figura 5.2 – Trecho do Grid no SUMO . . . . .  | 56 |
| Figura 5.3 – Topologia Densa - Com 1 Veículo Acidentado . . . . .                      | 58 |
| Figura 5.4 – Topologia Mista - Com 1 Veículo Acidentado . . . . .                      | 59 |
| Figura 5.5 – Topologia Esparsa - Com 1 Veículo Acidentado . . . . .                    | 60 |
| Figura 5.6 – Topologia Densa - Com 2 Veículo Acidentado . . . . .                      | 61 |
| Figura 5.7 – Topologia Mista - Com 2 Veículo Acidentado . . . . .                      | 63 |
| Figura 6.1 – Grid 1 Acidente - Média de mensagens geradas por execução. . . . .        | 67 |
| Figura 6.2 – Grid 1 Acidente - Média de mensagens geradas pelo total de nó. . . . .    | 68 |
| Figura 6.3 – Grid 1 Acidente - Média de mensagens geradas por retransmissores. . . . . | 69 |
| Figura 6.4 – Grid 1 Acidente - Média de mensagens recebidas por execução. . . . .      | 69 |
| Figura 6.5 – Grid 1 Acidente - Média de mensagens recebidas por nó. . . . .            | 70 |
| Figura 6.6 – Grid 1 Acidente - Média de mensagens recebidas por receptores. . . . .    | 71 |
| Figura 6.7 – Grid 1 Acidente - Média do Distância de Recebimento . . . . .             | 71 |
| Figura 6.8 – Grid 1 Acidente - Média de Saltos de Recebimento . . . . .                | 72 |
| Figura 6.9 – Grid 1 Acidente - Porcentagens de Remetentes e Receptores. . . . .        | 73 |

|   |    |
|---|----|
| Figura 6.10 – Grid 2 Acidentes - Média de mensagens geradas por execução. . . . .           | 73 |
| Figura 6.11 – Grid 2 Acidentes - Média de mensagens geradas pelo total de nós. . . . .      | 74 |
| Figura 6.12 – Grid 2 Acidentes - Média de mensagens geradas por retransmissores. . . . .    | 75 |
| Figura 6.13 – Grid 2 Acidentes - Média de mensagens recebidas por execução. . . . .         | 75 |
| Figura 6.14 – Grid 2 Acidentes - Média de mensagens recebidas por nós. . . . .              | 76 |
| Figura 6.15 – Grid 2 Acidentes - Média de mensagens recebidas por receptores. . . . .       | 76 |
| Figura 6.16 – Grid 2 Acidentes - Média do Distância de Recebimento . . . . .                | 77 |
| Figura 6.17 – Grid 2 Acidentes - Média de Saltos de Recebimento . . . . .                   | 78 |
| Figura 6.18 – Grid 2 Acidentes - Porcentagens de Remetentes e Receptores. . . . .           | 78 |
| Figura 6.19 – Cologne 1 Acidente - Média de mensagens geradas por execução. . . . .         | 79 |
| Figura 6.20 – Cologne 1 Acidente - Média de mensagens geradas por retransmissores. . . . .  | 80 |
| Figura 6.21 – Cologne 1 Acidente - Média de mensagens recebidas por execução. . . . .       | 80 |
| Figura 6.22 – Cologne 1 Acidente - Média de mensagens recebidas por receptores. . . . .     | 81 |
| Figura 6.23 – Cologne 1 Acidente - Média do Distância de Recebimento . . . . .              | 82 |
| Figura 6.24 – Cologne 1 Acidente - Média de Saltos de Recebimento . . . . .                 | 82 |
| Figura 6.25 – Cologne 1 Acidente - Porcentagens de Remetentes e Receptores. . . . .         | 83 |
| Figura 6.26 – Cologne 2 Acidentes - Média de mensagens geradas por execução. . . . .        | 84 |
| Figura 6.27 – Cologne 2 Acidentes - Média de mensagens geradas por retransmissores. . . . . | 84 |
| Figura 6.28 – Cologne 2 Acidentes - Média de mensagens recebidas por execução. . . . .      | 85 |
| Figura 6.29 – Cologne 2 Acidentes - Média de mensagens recebidas por receptores. . . . .    | 86 |
| Figura 6.30 – Cologne 2 Acidentes - Média do Distância de Recebimento . . . . .             | 86 |
| Figura 6.31 – Cologne 2 Acidentes - Média de Saltos de Recebimento . . . . .                | 87 |
| Figura 6.32 – Cologne 2 Acidentes - Porcentagens de Remetentes e Receptores. . . . .        | 88 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 3.1 – Comparação entre Protocolos de Disseminação . . . . . | 37 |
| Tabela 4.1 – Distâncias de Reação e Frenagem . . . . .             | 48 |

## SUMÁRIO

|          |   |    |
|----------|---|----|
| <b>1</b> | <b>INTRODUÇÃO</b>   | 12 |
| 1.1      | Motivação   | 13 |
| 1.2      | Definição do Problema   | 14 |
| 1.3      | Objetivos   | 14 |
| 1.4      | Solução Proposta  | 15 |
| 1.5      | Contribuições do Trabalho   | 15 |
| 1.6      | Estrutura do Trabalho   | 16 |
| <b>2</b> | <b>REFERENCIAL TEÓRICO</b>  | 17 |
| 2.1      | Aplicações de Redes Veiculares                                      | 19 |
| 2.2      | Arquitetura WAVE  | 20 |
| 2.3      | Disseminação de Mensagens nas Redes Veiculares                      | 22 |
| 2.3.1    | Comunicação por <i>Geocast</i>                                      | 23 |
| 2.3.2    | Comunicação por <i>Unicast</i>                                      | 23 |
| 2.3.3    | Comunicação por <i>Multicast</i>                                    | 24 |
| 2.3.4    | Comunicação por <i>Anycast</i>                                      | 24 |
| 2.3.5    | Comunicação por <i>Broadcast</i>                                    | 24 |
| 2.4      | Protocolos de Disseminação por <i>Broadcast</i> em Redes Veiculares | 26 |
| 2.4.1    | Baseados em Atraso  | 27 |
| 2.4.2    | Baseados em Cluster   | 27 |
| 2.4.3    | Baseados em Contador  | 27 |
| 2.4.4    | Baseados em Distância   | 28 |
| 2.4.5    | Baseados em Localização   | 28 |
| 2.4.6    | Baseados em Probabilidade   | 28 |
| 2.4.7    | Baseados em Topologia   | 28 |
| 2.5      | Ambientes de Simulação  | 28 |
| 2.5.1    | OMNeT++ - <i>Objective Modular Network Testbed in C++</i>           | 29 |
| 2.5.2    | SUMO - <i>Simulation of Urban MObility</i>                          | 29 |
| 2.5.3    | Veins   | 30 |
| 2.6      | <i>Traces</i> de Mobilidade Veicular                                | 31 |
| <b>3</b> | <b>TRABALHOS RELACIONADOS</b>                                       | 33 |
| <b>4</b> | <b>PROTOCOLO DE DISSEMINAÇÃO DE MENSAGENS DE ALERTA</b>             | 39 |

|              |   |           |
|--------------|---|-----------|
| <b>4.1</b>   | <b>RODOGE - <i>Rules that Oppose overdone Dissemination Of messaGEs</i></b> . . . . . | <b>39</b> |
| <b>4.1.1</b> | <b>Regras de Reenvio de Mensagem</b> . . . . .  | <b>40</b> |
| <b>4.1.2</b> | <b>Diagrama de Funcionamento</b> . . . . .  | <b>43</b> |
| <b>4.1.3</b> | <b>Estrutura da Mensagem Enviada na Rede</b> . . . . .                                | <b>45</b> |
| <b>4.1.4</b> | <b>Estrutura da Tabela de Mensagens</b> . . . . .                                     | <b>46</b> |
| <b>4.1.5</b> | <b>Funções do Protocolo</b> . . . . .   | <b>47</b> |
| <b>4.1.6</b> | <b>Classificação do RODOGE</b> . . . . .  | <b>54</b> |
| <b>5</b>     | <b>METODOLOGIA</b> . . . . .  | <b>55</b> |
| <b>5.1</b>   | <b>Cenários de Simulação</b> . . . . .  | <b>55</b> |
| <b>5.1.1</b> | <b>Cenário de Cologne</b> . . . . .   | <b>55</b> |
| <b>5.1.2</b> | <b>Grid de Manhattan</b> . . . . .  | <b>56</b> |
| <b>5.2</b>   | <b>Topologias de Simulação</b> . . . . .  | <b>57</b> |
| <b>5.2.1</b> | <b>Topologia Densa - Com 1 Veículo Acidentado</b> . . . . .                           | <b>57</b> |
| <b>5.2.2</b> | <b>Topologia Mista - Com 1 Veículo Acidentado</b> . . . . .                           | <b>59</b> |
| <b>5.2.3</b> | <b>Topologia Esparsa - Com 1 Veículo Acidentado</b> . . . . .                         | <b>60</b> |
| <b>5.2.4</b> | <b>Topologia Densa - Com 2 Veículo Acidentado</b> . . . . .                           | <b>61</b> |
| <b>5.2.5</b> | <b>Topologia Mista - Com 2 Veículo Acidentado</b> . . . . .                           | <b>62</b> |
| <b>5.3</b>   | <b>Métricas de Validação</b> . . . . .  | <b>64</b> |
| <b>5.3.1</b> | <b>Quantidade de Mensagens Geradas</b> . . . . .                                      | <b>64</b> |
| <b>5.3.2</b> | <b>Quantidade de Mensagens Recebidas</b> . . . . .                                    | <b>64</b> |
| <b>5.3.3</b> | <b>Distância para Recebimento</b> . . . . .   | <b>65</b> |
| <b>5.3.4</b> | <b>Saltos para Recebimento</b> . . . . .  | <b>65</b> |
| <b>5.3.5</b> | <b>Porcentagem de Geradores de Mensagens</b> . . . . .                                | <b>65</b> |
| <b>5.3.6</b> | <b>Porcentagem de Veículos Cobertos</b> . . . . .                                     | <b>66</b> |
| <b>5.4</b>   | <b>Protocolos Comparados</b> . . . . .  | <b>66</b> |
| <b>6</b>     | <b>RESULTADOS E DISCUSSÃO</b> . . . . .   | <b>67</b> |
| <b>6.1</b>   | <b>Ambiente Grid com 1 Acidente</b> . . . . .   | <b>67</b> |
| <b>6.2</b>   | <b>Ambiente Grid com 2 Acidentes</b> . . . . .  | <b>73</b> |
| <b>6.3</b>   | <b>Ambiente Cologne com 1 Acidente</b> . . . . .                                      | <b>79</b> |
| <b>6.4</b>   | <b>Ambiente Cologne com 2 Acidentes</b> . . . . .                                     | <b>83</b> |
| <b>7</b>     | <b>CONCLUSÃO</b> . . . . .  | <b>89</b> |
|              | <b>REFERÊNCIAS</b> . . . . .  | <b>90</b> |

## 1 INTRODUÇÃO

Todos os anos são registrados milhares de acidentes de trânsito, com vítimas fatais e vítimas com sequelas, com números que chegaram por ano a 1,25 milhão de óbitos e 50 milhões com alguma sequela (WHO, 2018). Inúmeros dispositivos de segurança têm sido implantados nos veículos para evitar que este número de vítimas continue a crescer, impedindo que mesmo com número crescente de veículos o número de vítimas não cresça na mesma proporção.

Com o avanço da tecnologia e da comunicação sem fio, o conceito de um veículo conectado em rede com outros veículos e outros dispositivos tem recebido atenção não apenas pelo meio acadêmico, mas também por grande parte das fabricantes de automóveis. A ideia de veículos e estradas conectados para tornar o transporte rodoviário mais seguro, eficiente e agradável tem se tornado frequente (CHOWDHURY; MACKENZIE; PERKINS, 2014).

Uma área crescente, que tem como objetivo diminuir o número de vítimas em acidentes, é a rede veicular, que visa a troca de mensagens entre veículos e dispositivos de infraestrutura para evitar situações de risco, auxiliar em caso de um acidente, prevenir colisões e controlar congestionamentos (KARAGIANNIS et al., 2011).

Redes Veiculares, também chamadas de VANETs (*Vehicular Ad Hoc Networks*), são um tipo de rede *Ad Hoc* formada por veículos e dispositivos de infraestrutura e que podem possuir diferentes topologias de comunicação. A forma de comunicação neste tipo de rede varia de acordo com quais dispositivos está sendo realizada a troca de informações e a topologia da rede (VEGNI; LITTLE, 2010).

Uma rede *Ad Hoc* é um tipo de rede que utiliza dispositivos sem hierarquia estabelecida, todos os dispositivos podem comunicar-se diretamente sem a gerência de comunicação. Nesse tipo de rede a informação pode transitar entre os dispositivos intermediários até alcançar o dispositivo alvo (ZAFOUNE; KANAWATI; MOKHTARI, 2007).

As VANETs trabalham com informações que são coletadas diretamente no veículo, que auxiliam na constatação de acidente. Como por exemplo, o estado em que o *airbag* se encontra, pois quando acionado é muito provável a ocorrência de um acidente. Tais informações podem ser coletadas diretamente nos carros e há diferentes tipos.

As informações coletadas devem trafegar pela VANET, de modo que todos os veículos de interesse possam receber a informação. Para isso pode-se utilizar vários nós, automóveis ou dispositivos de infraestrutura, como nós intermediários. Devido ao carácter de emergência que

algumas dados possuem, este tipo de rede deve possuir baixa latência e alta eficiência (IBNEZ et al., 2011).

Para padronizar esse tipo de rede, as VANETs, foi criado um conjunto de padrões, como IEEE 802.11p e a arquitetura *Wireless Access in Vehicular Environments*, conhecida como WAVE, que define e determina regras de controle de acesso físico e ao meio.

As VANETs possuem certas características peculiares, como topologia altamente dinâmica, desconexões frequentes, mobilidade restrita, locomoção previsível e interferência na propagação do sinal, tais aspectos influenciam profundamente esse tipo de rede. Há também um problema característico, a *Broadcast Storm*, que ocorre devido ao grande número de mensagens que podem inundar a rede e assim desencadear numa série de problemas para o seu funcionamento, como colisão, contensão e outros.

Redes Veiculares possuem um infinidade de aplicações, não apenas ligadas à segurança, ou detecção e prevenção de acidentes, mas também ligadas a eficiência no trânsito, relacionadas ao controle de trânsito. Há também aplicações na área de conforto e entretenimento, relacionadas a conectividade e informações, por exemplo, sobre o clima, sobre lojas do local e etc.

## **1.1 Motivação**

O crescente número de acidentes de trânsito, com milhares de vítimas fatais todos os anos, abre portas para o desenvolvimento e para a pesquisa em soluções para amenizar tal problema. Uma aplicação para prevenir e detectar acidentes de trânsito permite a diminuição de tais números ao ponto de tornar as estradas mais seguras e eficientes que as dos dias atuais. Além disso, a tecnologia embarcada presente nos veículos possibilita a coleta de dados e o desenvolvimento de diversas aplicações.

Atualmente a indústria automobilística é uma das mais rentáveis do mundo e tem investido cada vez mais em veículos com maior segurança. Como é o caso da montadora Volvo, que é conhecida por ter os carros mais seguros do mundo e inovadora no quesito segurança. Seus carros possuem diversos sensores e sistemas, como piloto automático que sempre mantém distância programada dos veículos a frente, há também um sistema para detecção de pedestre, sistema para identificar o cansaço do motorista e muitos outros. Outras montadoras conhecidas pela segurança e que possuem diferentes sensores e sistemas como a Volvo são a BMW, Mercedes-Benz e há várias outras investindo na mesma área.



## 1.2 Definição do Problema

Devido ao grande número de acidentes ocorridos todos os dias, que causam inúmeras vítimas fatais, inúmeros feridos e inúmeros sequelados, uma aplicação que poderia diminuir essa enorme quantidade de pessoas afetadas seria de extrema importância para a sociedade.

As VANETs auxiliariam todo o trajeto de um motorista em uma determinada via, o que poderia ajudar a tomar decisões mais rápidas e mais corretas. Essa troca de mensagens entre veículos e dispositivos de infraestrutura poderia impactar positivamente no trânsito.

O envio de mensagens em *Broadcast* em redes veiculares é um desafio para área, uma vez que sem o devido controle pode gerar um número elevado de mensagens enviadas. A disseminação de mensagens em *Broadcast* frequentemente pode ocasionar em contenção e colisão de transmissão entre vizinhos. Com toda a informação sendo disseminada pela rede, em alguns casos ocorre inundação e na chamada *Broadcast Storm*. (TSENG; NI; SHIH, 2001). Um problema crítico em redes veiculares que pode acarretar em diversos problemas e assim deve ser contornado.

## 1.3 Objetivos

O objetivo principal deste trabalho é desenvolver um protocolo de disseminação de mensagens de alerta de acidente, que deve alertar outros motoristas interessados com tempo suficiente para reação e frenagem do veículo, utilizando um protocolo de disseminação baseado em *broadcast*, porém controlando os reenvios para evitar o problema de *broadcast storm*.

Os objetivos específicos são divididos das seguintes formas:

1. Analisar cenários reais, implementar cenários fictícios e simular veículos acidentados em ambos;
2. Analisar o problema de *broadcast storm*, verificar as causas, as consequências e as soluções;
3. Implementar um protocolo de disseminação de mensagens de alerta com controle de reenvio de mensagens;
4. Realizar simulações nos cenários analisados e implementados para verificar o funcionamento do protocolo proposto e comparar a soluções encontradas.

## 1.4 Solução Proposta

Este trabalho apresenta o RODOGE - *Rules to prevent Over-Dissemination Of messages*, um protocolo de disseminação de mensagens de alerta de acidente voltado para redes veiculares. O protocolo possui diferentes mecanismos para contornar a inundação da rede, *broadcast storm*.

O protocolo RODOGE utiliza regras no seu funcionamento para controlar os reenvios de mensagens na rede. Tais regras são o fundamento do protocolo proposto e são: (i) validar mensagem com identificador único, (ii) tempo de vida da mensagem deve ser superior e inversamente proporcional à velocidade do veículo, (iii) na área do dispositivo de infraestrutura não há disseminação de mensagem pelos veículos, (iv) o tempo de *backoff* deve ser proporcional à quantidade de mensagens recebidas pelo veículo e (v) o tempo final para reenvio da mensagem é inversamente proporcional à distância calculada.

Além disso, o RODOGE utiliza a distância de reação do motorista e frenagem do veículo, ou distância segura, como base para o intervalo de tempo entre as mensagens. Utilizando essa distância é possível garantir que um veículo recebe uma mensagem com distância superior à distância segura, mesmo diminuindo o número de mensagens reenviadas.

## 1.5 Contribuições do Trabalho

O trabalho realizado e o protocolo RODOGE possui algumas contribuições para o meio científico e acadêmico, como:

1. Uma diferente abordagem e tratamento para o problema de *Broadcast Storm*;
2. Considerar a utilização de dispositivos de infraestrutura para tratar o problema de *Broadcast Storm*;
3. Lidar com diferentes cenários, urbano e rodoviário;
4. Simular com mais de um veículo envolvido no acidente;
5. Considerar o tempo de reação e frenagem do veículo na disseminação das mensagens;
6. Ajustar dinamicamente o tempo de vida da mensagem, com base na velocidade do veículo;

7. Ajustar dinamicamente o intervalo de reenvio do alerta de acidente, com base na densidade local da mensagem e distância da origem.

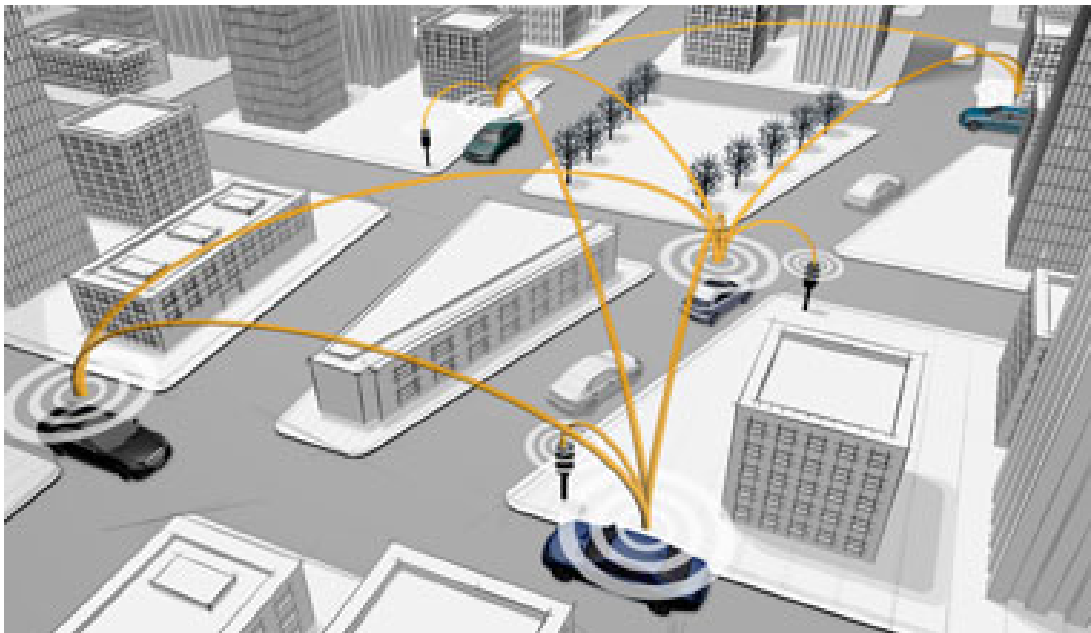
## **1.6 Estrutura do Trabalho**

Este trabalho está organizado da seguinte forma: no Capítulo 2 são mostrados conceitos fundamentais, características e fatores que envolvem Redes Veiculares para entendimento do trabalho. No Capítulo 3 são apresentados os trabalhos relacionados, na qual mostra as soluções encontradas na literatura. No Capítulo 4 é apresentada a descrição detalhada do protocolo de disseminação RODOGE. No Capítulo 5 é apresentada a metodologia utilizada no desenvolvimento do trabalho, como os cenários utilizados, diferentes topologias nas simulações e métricas utilizadas para comparação. No Capítulo 6, os resultados obtidos com os testes são apresentados e discutidos, são avaliadas e comparadas as métricas descritas com os protocolos clássicos para um e dois acidentes nas simulações. Por fim, o Capítulo 7 apresenta as conclusões obtidas com este trabalho e propostas para a evolução do protocolo em trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Nos últimos anos, com o avanço das tecnologias de comunicação, tem surgido o conceito de carros conectados que trocam mensagens entre si, apresentado na Figura 2.1. Neste cenário, as redes veiculares propõem um serviço de segurança no trânsito, com informações sobre engarrafamentos, acidentes, condições da estrada e informações meteorológicas. Além disso, essas redes buscam aumentar a eficiência do trânsito, como a redução de engarrafamento, aumento do fluxo de veículos, rotas mais curtas, redução do consumo de combustível, controle da rede viária e outros infinitos avanços possíveis (CHOWDHURY; MACKENZIE; PERKINS, 2014; CHEN; LIU; CHEN, 2010; TOSTES et al., 2013).

Figura 2.1 – Estabelecimento de uma VANETs.



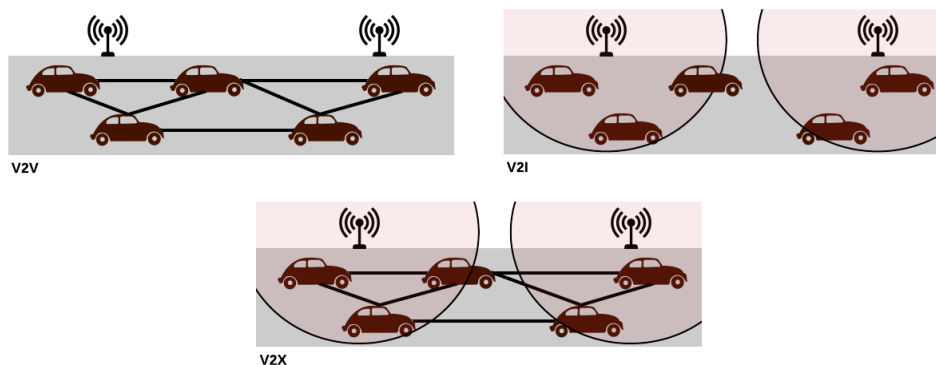
Fonte <<https://www.gta.ufrj.br/>>.

As VANETs são um tipo de rede *Ad Hoc*, que utilizam tecnologia sem fio e que estabelece comunicação entre veículos e dispositivos de infraestrutura. Neste cenário o padrão IEEE 1609 define dois tipos de dispositivos presentes nesse tipo de rede, o *Roadside Unit* (RSU), que é o dispositivo de infraestrutura presente nas vias, e o *Onboard Unit* (OBU), que é o dispositivo presente nos veículos (IEEE1609.0, 2014).

Com estes dois dispositivos distintos que compõem a rede, que são denominados nós, é possível estabelecer e classificar as comunicações entre eles em três tipos, que pode ser observadas na Figura 2.2 (SICHITIU; KIHLE, 2008):

- Comunicação entre veículos: que pode ser observada na Figura 2.2(a), também chamada de *vehicle to vehicle* ou V2V, é exclusivamente entre um ou mais veículos, cada um utiliza seu OBU e assim é estabelecida a comunicação entre eles. Pode ser com apenas um salto ou com múltiplos saltos, caso o receptor não esteja na área de cobertura do emissor;
- Comunicação entre veículo e dispositivo de infraestrutura: que pode ser observada na Figura 2.2(b), também chamada de *vehicle to infrastructure* ou V2I, é a comunicação exclusiva entre um veículo, com seu OBU e um dispositivo de infraestrutura, com seu RSU;
- Comunicação híbrida: que pode ser observada na Figura 2.2(c), esse tipo de comunicação envolve dois ou mais veículos e um ou mais dispositivos de infraestrutura, também chamada de V2X, é uma comunicação híbrida que envolve os dois outros tipos de comunicação, V2V e V2I.

Figura 2.2 – Tipos de Comunicação.



Fonte do Autor.

As VANETs possuem algumas características diferentes e peculiares se comparadas com outros tipos de redes devido as suas particularidades, como:

- Topologia altamente dinâmica: uma via é um ambiente extremamente dinâmico, com veículos em alta velocidade e por muitas vezes em sentidos opostos. Devido a isso, os automóveis podem entrar e sair da rede rapidamente num curto período de tempo ocasionando mudanças frequentes na topologia (MEJRI; BEN-OTHTMAN; HAMDI, 2014);
- Desconexão frequente: devido à topologia altamente dinâmica e mudanças frequentes na conectividade, dois veículos ou veículo e infraestrutura podem perder a conexão en-

tre si rapidamente e interromper a troca de dados entre eles (MEJRI; BEN-OTHMAN; HAMDI, 2014);

- Densidade variável: a quantidade de veículos no decorrer do percurso é extremamente dinâmica, há momentos em que existe um grande número de veículos e instantes seguintes o número de veículos cai drasticamente (AL-SULTAN et al., 2014);
- Sem restrição energética: veículos possuem capacidade energética constante, capazes de suportar unidades de processamento, armazenamento e comunicação (HARTENSTEIN; LABERTEAUX, 2008);
- Poder computacional: os veículos possuem um grande número de sensores e recursos, como posição geográfica, computadores de bordo e auxiliares de estacionamento, que estão relacionados ao poder computacional e que podem obter dados mais confiáveis (AL-SULTAN et al., 2014);
- Comunicação direcionada: além dos veículos poderem ser identificados por um identificador único, eles também podem ser identificados por sua posição geográfica (KARAGIANNIS et al., 2011);
- Mobilidade restrita e previsível: veículos normalmente seguem um fluxo, que são as ruas, estradas e avenidas, com semáforos e limites de velocidade, assim seguem um certo padrão de locomoção, sendo possível prever uma posição futura do automóvel (LI; WANG, 2007);
- Propagação do sinal: as VANETs podem ser implantadas em diferentes ambientes, como auto-estradas, ruas, avenidas e estradas rurais, cada um dos locais possui sua interferência de sinal e assim assume-se que o espaço não é totalmente livre para propagação do sinal (MEJRI; BEN-OTHMAN; HAMDI, 2014).

## 2.1 Aplicações de Redes Veiculares

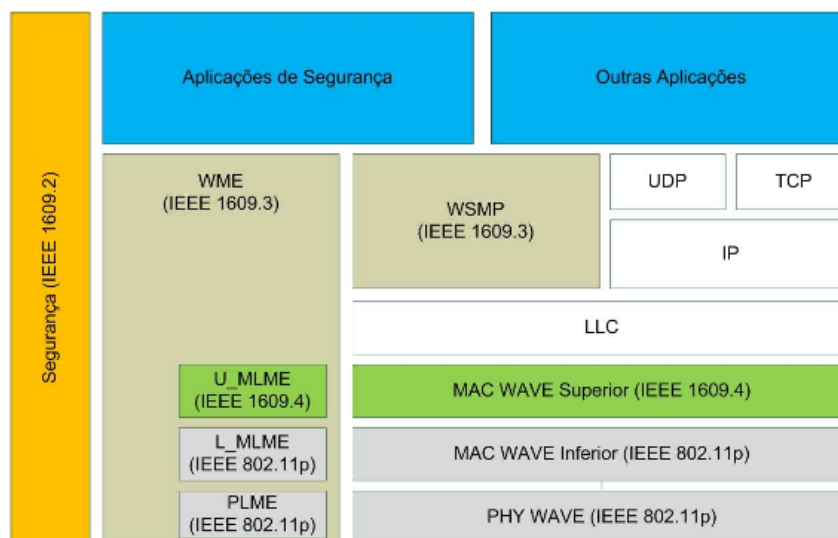
As Redes Veiculares possuem inúmeras aplicações, sejam de segurança, de entretenimento ou de conforto. O tipo de informação trocada entre os veículos é crucial para determinar qual o tipo de aplicação, que podem ser (KARAGIANNIS et al., 2011):

- Aplicações de Segurança: visa a prevenção e detecção de acidentes nas vias, para isso utiliza a troca de informações ligadas ao estado do veículo, como o estado do *airbag*, cálculo da desaceleração, controle de estabilidade, entre outros (AMARANTE et al., 2015). Para esse tipo de aplicação a confiabilidade na entrega das informações é extremamente importante, uma vez que é crucial que todos os veículos próximos do acidente recebam o alerta de acidente.
- Aplicações de Eficiência no Trânsito: para essas aplicações o conhecimento da posição dos veículos é crucial, pois com isso é possível alcançar o principal objetivo dessa aplicação, que é melhorar a mobilidade nas ruas e estradas. A alta disponibilidade de rede é extremamente importante, já que são necessárias diversas informações para que o motorista, ou mesmo o próprio veículo, tome alguma decisão em relação à rota. Como exemplo dessa aplicação, pode-se citar o controle de semáforos e engarrafamentos.
- Aplicações de Entretenimento: este tipo de aplicação está relacionada ao bem estar do motorista, fornece dados sobre o clima, lanchonetes, postos de gasolina, entre outras informações. Além disso, esse tipo de aplicação pode fornecer também conexão à internet, acesso a *chats*, músicas, vídeos, entre outras coisas. Este tipo de aplicação está relacionada a conectividade e disponibilidade, já que a troca de informação em muitos casos deve ser contínua e rápida.
- Aplicações de Sensoriamento: essas aplicações visam o monitoramento de determinadas vias e estradas, podendo monitorar as condições ambientais de uma determinada região de interesse. Uma vez que os veículos não possuem restrições energéticas como nas tradicionais redes de sensores sem fio, estes podem ser equipados com poderosas unidades de processamento, eficientes dispositivos de comunicação, sistema de navegação e uma infinidade de sensores.

## 2.2 Arquitetura WAVE

Devido as características presentes nas Redes Veiculares, como alta mobilidade e dependência das vias, no ano de 2004, o IEEE desenvolveu um padrão de comunicações para este tipo de rede. O padrão IEEE 1609, que serve como base para as redes veiculares, este é dividido em diferentes documentos e trata das diferentes características desse tipo de rede. A Figura 2.3 exhibe a arquitetura 1609 em relação aos documentos que a definem. Estes documentos são:

Figura 2.3 – Arquitetura da Rede Veicular - IEEE 1609.



Fonte (ALVES et al., 2009).

- IEEE 1609.0 *Architecture*: documento introdutório que apresenta o funcionamento, visão geral e componentes da Arquitetura. Define as unidades usadas na rede, RSU e OBU, auxilia no encaminhamento e processamento dos dados na rede (IEEE1609.0, 2014).
- IEEE 1609.1 *Resource Manager*: especifica a utilização da rede em aplicações de redes veiculares. Permite que aplicações remotas se comuniquem com aplicações executadas nos dispositivos presentes no veículo. Este padrão define a interoperabilidade de aplicações WAVE a fim de melhorar a performance e diminuir custos (IEEE1609.1, 2006).
- IEEE 1609.2 *Security Services for Applications and Management Messages*: padrão para desenvolver e executar as técnicas de segurança nas redes veiculares. Nesse tipo de rede o processamento e sobrecarga gastos com segurança devem ser o mínimo possível, já que as aplicações veiculares são muitas das vezes aplicações críticas. Contém mecanismos que gerenciam a autenticação de mensagens, entre as que requerem ser anônimas e as que necessitam ser criptografadas (IEEE1609.2, 2016).
- IEEE 1609.3 *Networking Services*: padrão para definir os serviços das camadas, de enlace, de rede e transporte, definindo plano de gerenciamento, responsável pela configuração e manutenção, e pelo plano de dados, responsável pelos serviços de comunicação e envio de dados (IEEE1609.3, 2016).



- IEEE 1609.4 *Multi-Channel Operations*: padrão que define a gerência e coordenação dos múltiplos canais da rede. Dispositivos devem monitorar canais a espera de serviços disponíveis que possui o canal correto para um determinado serviço (ALVES et al., 2009).

### **Padrão IEEE 802.11p**

O padrão IEEE 802.11p é uma extensão do padrão IEEE 802.11 e baseado no padrão IEEE 802.11a, criado para definir as camadas física e de enlace das redes veiculares. Este padrão foi incorporado à Arquitetura WAVE para padronizar a comunicação nas VANETs.

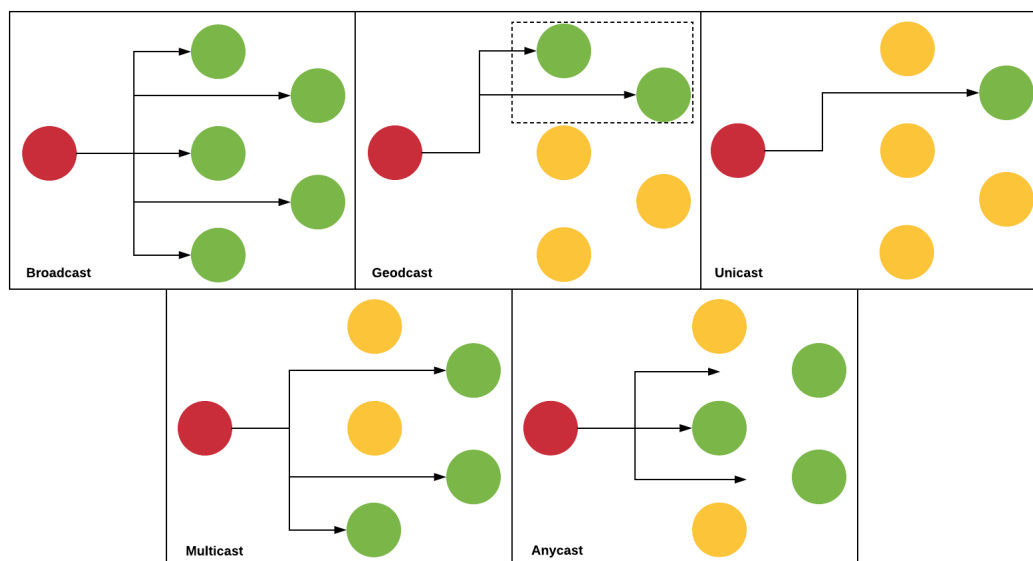
Nos Estados Unidos este padrão opera entre 5,850 GHz e 5,925 GHz, diferentemente dos padrões 802.11b/g que operam na faixa de 2,4 GHz. Na Europa o padrão 802.11p opera em diferentes frequências de acordo com o tipo da aplicação, de 5,855 GHz a 5,875 GHz é para aplicações gerais, de 5,875 GHz a 5,905 GHz para aplicações de emergência e segurança e de 5,905 GHz a 5,925 GHz para aplicações futuras. Utilizando essas frequências é possível obter um alcance de até 1000 metros, com taxa de transmissão no máximo de 27 Mbps e com movimentação entre os veículos de até 200 km/h (JIANG; DELGROSSI, 2008).

As redes de celulares 5G tem sido estudadas para utilização, juntamente com o padrão IEEE 802.11p, em redes veiculares, uma vez que possuem área de cobertura mais ampla, baixa latência e altas taxas de dados (QI et al., 2018). Com as redes 5G a comunicação direta de dispositivo para dispositivo (D2D) seria beneficiada, a utilização de dispositivos de infraestrutura com esta tecnologia poderia centralizar o controle e dados a eles (TEHRANI; UYSAL; YANIKOMEROGLU, 2014).

### **2.3 Disseminação de Mensagens nas Redes Veiculares**

A forma de disseminação de mensagens pode seguir diferentes padrões e tipos de comunicação, como *Broadcast*, *Geocast*, *Unicast*, *Multicast* e *Anycast*, como pode ser observado na Figura 2.4.

Figura 2.4 – Forma de Disseminação de Mensagens



Fonte do Autor.

### 2.3.1 Comunicação por *Geocast*

O *Geocast* pode ser dividido em duas formas distintas, *GeoBroadcast*, que envia informações a todos os veículos presente em uma região, e *GeoUnicast*, que destina uma mensagem a um único veículo e que pode realizar múltiplos saltos (FESTAG et al., 2008).

A disseminação por *Geocast* tem como vantagem a utilização da tecnologia de posicionamento para determinar a direção que a informação deve percorrer para alcançar seu destinatário. A utilização do *Geocast* em algumas aplicações de segurança torna a rede mais eficiente, uma vez que a utilizando pode-se direcionar uma informação para uma região de interesse e não enviar uma informação a todos (ALLAL; BOUDJIT, 2012).

Porém, há algumas restrições neste tipo de protocolo, uma dessas está ligada a ausência de veículos na direção do destinatário, ou mesmo quando não há dispositivos de infraestrutura.

### 2.3.2 Comunicação por *Unicast*

A forma de comunicação *Unicast* pode ser basicamente descrita como a comunicação entre dois veículos, um veículo envia uma mensagem a um veículo alvo. A mensagem é endereçada a um único veículo, podendo ou não realizar múltiplos saltos para alcançar o veículo alvo (FERREIRO-LAGE et al., 2009).

Este tipo de abordagem pode sofrer alguns problemas em redes veiculares, principalmente em redes esparsas. Em redes esparsas onde há trechos de rede sem cobertura ou sem conexão entre veículos, uma dada mensagem poderia não ser entregue.

### **2.3.3 Comunicação por *Multicast***

O *Multicast* tem funcionamento similar a comunicação do *Unicast*, porém a mensagem não possui apenas um destinatário mas um grupo de destinatários. A vantagem de se usar esse tipo de comunicação é que as mensagens são direcionadas apenas para veículos interessados em receber tal mensagem, fazendo com que se diminua o tráfego na rede (FAROOQ et al., 2015).

Este tipo de comunicação em redes veiculares pode não ser muito interessante, uma vez que apenas alguns veículos recebem a mensagem. Em aplicações de segurança que todos os veículos tem interesse em receber mensagens da rede tornaria o *Multicast* implantado em um *Broadcast*.

### **2.3.4 Comunicação por *Anycast***

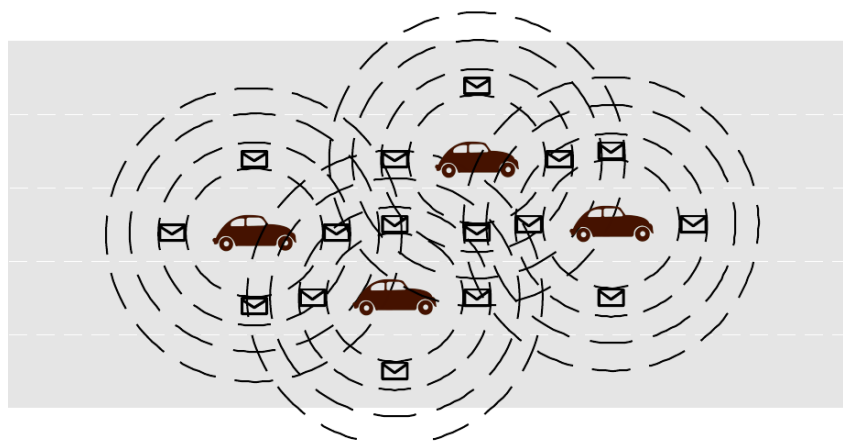
O *Anycast* funciona com uma mensagem, ou várias mensagens, de um único veículo remetente é que encaminhada para um veículo topologicamente mais próximo de um grupo de receptores potenciais. A mensagem pode ser enviada a vários veículos ao mesmo tempo, todos identificados pelo mesmo endereço de destino (KHAN; AYUB; SAEED, 2013).

Em comparação as outras formas de comunicação, o *Anycast* utiliza como endereçamento uma associação de um veículo para um veículo mais próximo. Diferente por exemplo do *Broadcast*, que utiliza como endereçamento uma associação de um veículo para muitos.

### **2.3.5 Comunicação por *Broadcast***

Conceitualmente a comunicação por *Broadcast*, que também é conhecida por difusão, é quando uma determinada mensagem ou dado é entregue a todos os veículos presentes na rede. A forma como essa mensagem é entregue a todos os veículos pode variar de acordo com o tamanho que essa rede possui, muitas das vezes é necessário que a mensagem realize múltiplos saltos pelos veículos para que todos eles possam receber a mensagem desejada (TSENG; NI; SHIH, 2001).

Figura 2.5 – Disseminação por Broadcast em Redes Veiculares.



Fonte do Autor.

A disseminação por *Broadcast* é a forma mais eficiente de se transmitir mensagens de segurança em uma rede veicular, devido a alta latência que este tipo de rede possui. Neste tipo de rede é importante que todos os veículos recebam um alerta de colisão e assim é realizada a inundação da mensagem na rede (WISITPONGPHAN et al., 2007). Porém, essa inundação não controlada pode acarretar em mensagens redundantes, devido a retransmissão de mensagens a veículos que já tenham recebido e reenviado dados.

### ***Broadcast Storm***

A *Broadcast Storm* se caracteriza pela inundação de mensagens na rede por vários veículos e está ligada a redundância de dados e retransmissão descontrolada. As redes veiculares utilizam *broadcast* como forma de disseminação para mensagens de alerta e segurança, o que pode ocasionar em um grande número de mensagens enviadas. Este tipo de transmissão pode causar colisão de pacotes, disputa na camada de enlace, uso ineficiente da largura de banda e interrupção do serviço devido à alta contenção (TSENG; NI; SHIH, 2001).

Os principais problemas causados pela *Broadcast Storm* podem ser (TSENG; NI; SHIH, 2001):

- **Redundância:** quando um veículo retransmite uma mensagem a seus vizinhos, todos os seus vizinhos podem ter recebido essa mesma mensagem anteriormente;
- **Contenção:** após a transmissão por um veículo, os vizinhos que recebem a mensagem devem retransmiti-la, ao retransmitirem muitas mensagens podem ser recebidas ao mesmo tempo pelos veículos e assim ao retransmitirem as mensagens ficarão prendendo umas as outras;

- Colisão: ausência de comunicação RTS (*request to send*) e CTS (*clear to send*) e deficiência do mecanismo de *backoff* podem ocasionar em envios simultâneos pelos veículos.

Uma abordagem que claramente ocorre a *Broadcast Storm* é quando um veículo ao receber um dado por *Broadcast* pela primeira vez deve obrigatoriamente reenviar este dado a todos os seus vizinhos e assim sucessivamente.

A forma de diminuir os efeitos causados pela *Broadcast Storm* é utilizar mecanismos de controle para o reenvio. Este controle faz com que haja um seleção de vizinhos mais propícios a serem os disseminadores da mensagem, tal seleção reduz o número de que veículos realizam o *broadcast*. Tal seleção pode ocorrer de diversas formas, como baseados na localização do remetente, na direção do receptor e outras muitas formas (KORKMAZ et al., 2004).

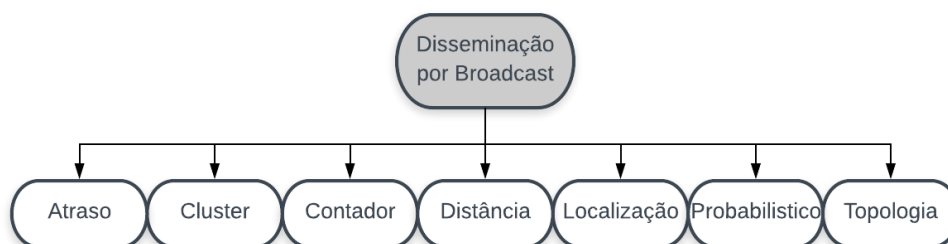
### **Protocolos de Disseminação por *Broadcast***

Protocolos baseados em *broadcast* podem ser classificados de diferentes formas, de acordo como ocorre a forma de inundação pela rede, eles pode ser divididos em dois grupos (KUROSE; ROSS, 2007):

- Inundação Não Controlada: a forma simples de *broadcast*, onde um veículo remetente envia uma mensagem a todos os seus vizinhos, quando um veículo recebe essa mensagem, ele a reenvia a seus vizinhos, assim sucessivamente. Caso a rede contenha laços, ocorrerá uma troca de mensagens entre veículos que já receberam a mensagem e estes mesmos veículos irão reenviar a mensagem novamente, causando a chamada *Broadcast Storm*.
- Inundação Controlada: para controlar a inundação de mensagens e assim evitar o *Broadcast Storm*, esse tipo de protocolo utiliza de mecanismos para diminuir a quantidade de mensagens reenviadas. Pode utilizar comparações entre as mensagens recebidas para decidir o reenvio, pode selecionar veículos para realizar o reenvio e outras formas.

## **2.4 Protocolos de Disseminação por *Broadcast* em Redes Veiculares**

Para contornar os problemas do *Broadcast Storm* existem diferentes tipos de protocolos de disseminação de mensagens em redes veiculares. Estes protocolos podem ser classificados e divididos em classes, de acordo com as formas que utilizam para controlar os reenvios de mensagens e contornar o problema do *Broadcast Storm* (CHA, 2014; CUNHA, 2017). A classificação dos protocolos de disseminação em *broadcast* pode ser observada na Figura 2.6:

Figura 2.6 – Tipos de Protocolo *Broadcast* por Inundação Controlada

Fonte (CHA, 2014; CUNHA, 2017) Adaptado.

### 2.4.1 Baseados em Atraso

Uma das formas de diminuir o número de mensagens reenviadas na rede é utilizando um tempo aleatório de atraso para realizar as retransmissões. Os veículos vizinhos ao receberem uma mensagem de um veículo devem aguardar o tempo aleatório selecionado para realizar a sua retransmissão, e assim sucessivamente. Com esta técnica é possível diminuir o problema de *broadcast storm*.

### 2.4.2 Baseados em Cluster

Para tentar diminuir os problemas gerados com reenvio de mensagens descontrolado, pode ser utilizada a estratégia de líder para realizar o reenvio, na qual é selecionado um líder para o cluster, para realizar os reenvios em uma determinada área. A seleção pode ocorrer com os veículos trocando *beacon* com seus vizinhos, com identificação e localização, e assim eles possuem conhecimento dos veículos ao seu redor e então é selecionado o veículo nesta área de acordo com a proposta do projeto, como pro exemplo, selecionar o veículo com menor identificador, ou com o maior número de vizinhos.

### 2.4.3 Baseados em Contador

Esta forma de controle de reenvio analisa as mensagens recebidas por um veículo para tomar a decisão de realizar o seu reenvio. Antes de um veículo realizar a sua retransmissão ele pode receber a mesma mensagem inúmeras vezes, assim é definido um valor ideal e caso o número de vezes que este veículo recebeu a mensagem a ser retransmitida supere o valor ideal selecionado, ocorre um cancelamento do reenvio e um novo agendamento de retransmissão.

#### **2.4.4 Baseados em Distância**

Esta forma de controle de reenvio analisa a distância entre remetente e receptor para a tomada de decisão no reenvio da mensagem. Este método toma como base que veículos mais distantes do remetente devem ter prioridades em relação aos veículos mais próximos, pois ao reenviarem a sua mensagem eles a disseminam em áreas que a mensagem anterior não alcançou, diferentemente dos veículos mais próximos.

#### **2.4.5 Baseados em Localização**

Para controlar o reenvio de mensagens e tentar inibir a *broadcast storm* é utilizada como base a localização do veículo e seus vizinhos. Os veículos realizam trocas de *beacons* com suas localizações e identificadores. Um veículo ao receber uma mensagem repetida de diferentes veículos possui conhecimento para analisar a sua cobertura e tomar a decisão de reenvio, caso muitos veículos conhecidos não tenham reenviado a mensagem.

#### **2.4.6 Baseados em Probabilidade**

Esta forma de controle de reenvio de mensagens utiliza de probabilidade para a tomada decisão do reenvio. Essa probabilidade é calculada com base na chance do veículo em retransmitir a mensagem e alcançar outros veículos que ainda não tenham recebido a mensagem. Em alguns casos pode ocorrer de veículos possuírem a mesma probabilidade e o funcionamento ocorrer similar ao *broadcast* puro.

#### **2.4.7 Baseados em Topologia**

Para realizar o contro de reenvio de mensagens é utilizada como base a topologia local para tomada de decisão do reenvio. O veículo analisa informações de densidade e conectividade antes de realizar seu reenvio ou cancelamento. O cenário é dividido em zonas com base no seu alcance de cobertura e um veículo é selecionado para realizar a retransmissão.

### **2.5 Ambientes de Simulação**

A utilização de simuladores para realizar experimentos em redes veiculares é fundamental, dada toda a dificuldade relacionada a um ambiente real. Em experimentos em redes

veiculares, principalmente para disseminação de alerta de mensagens, exigem um grande número de veículos capazes de se comunicar e assim estabelecer a rede e, além disso, para criar um alerta é necessário a ocorrência de um acidente com um veículo, essas necessidades tornam inviáveis experimentos reais em larga escala (ZEADALLY et al., 2012).

Existe um grande número de simuladores que podem ser utilizados para simulações em redes veiculares, como NCTUns, VANETSIm, TRANS, VSimTri e Veins (OMNeT++ e SUMO). O Veins e seus componentes foram selecionados, pois são simuladores *open source*, com grande difusão no meio acadêmico e com vasta documentação disponível para estudos.

### 2.5.1 OMNeT++ - *Objective Modular Network Testbed in C++*

O OMNeT++ é um simulador de redes de computadores, modular, baseado em componentes de C++, um *framework* e um biblioteca de simulação, sendo usado principalmente para construção de simulações de redes (OMNET++, 2018). Abrange uma grande infinidade de redes, como redes com e sem fio, redes de sensores sem fio, redes ad hoc e outras. A IDE fornecida pelo simulador é baseado no Eclipse, possui uma interface gráfica e inúmeras ferramentas. Existem extensões para simulação em tempo real, emulação de rede, integração de banco de dados e outros. Para o meio acadêmico seu licenciamento é de software livre, utilizado o modelo GNU. A sua interface pode ser verificada na Figura 2.7.

Figura 2.7 – Simulador OMNeT++.



Fonte <<http://omnet-tutorial.com/>>.

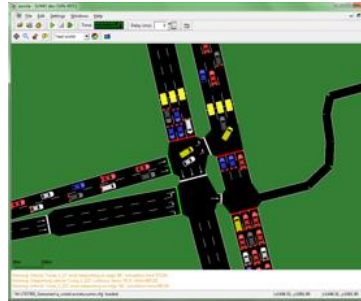
### 2.5.2 SUMO - *Simulation of Urban MObility*

O SUMO é um simulador de tráfego urbano e seu licenciamento é de software livre (SUMO, 2016). O SUMO cria simulações de tráfego rodoviário de pequena e larga escala, com quantidade de veículos variáveis. Para criação de uma simulação, pode-se criar um mapa qualquer ou utilizar um mapa real de qualquer cidade desejada. Existe uma comunidade ativa



do SUMO com toda documentação necessária, tutoriais e projetos para melhor entendimento do seu funcionamento, sua interface pode ser verificada na Figura 2.8.

Figura 2.8 – Simulador Sumo.

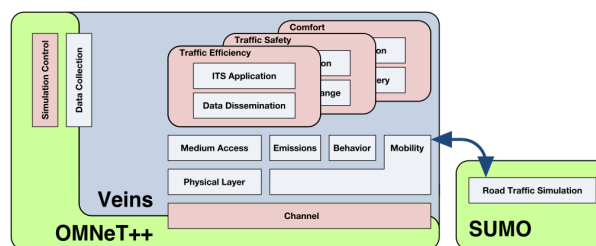


Fonte <[http://sumo.dlr.de/wiki/Main\\_Page/](http://sumo.dlr.de/wiki/Main_Page/)>.

### 2.5.3 Veins

O Veins é um *framework* de código aberto, utilizado para a simulação de redes veiculares. É baseado no simuladores OMNeT++ e SUMO, estes são estendidos pelo Veins e assim é fornecido um amplo conjunto de modelos para a realização de simulações em redes veiculares. Sua arquitetura, utilizando o SUMO e OMNeT++, pode ser observada na Figura 2.9 (SOMMER, 2016).

Figura 2.9 – Arquitetura Simulador Veins.



Fonte <<http://veins.car2x.org/>>.

Para que a execução ocorra em paralelo os simuladores OMNeT++ e SUMO são interligados por meio de uma socket TCP, e assim seguido de um protocolo de padronização, TraCI (*Traffic Control Interface*), com isto é possível que a simulação seja acoplada pela simulação de rede e pela simulação de tráfego. Os movimentos realizados na simulação de tráfego pelos veículos no SUMO são refletidos em movimentos dos nós contidos na simulação de rede no OMNeT++ (SOMMER, 2016).

O Veins possui alguns detalhes que auxiliam na simulação, como um modelo de propagação de chamado *Two-Ray Interference*, que consegue captar os efeitos de reflexão do solo.

Há também um modelo de sombreamento de obstáculos, que visa capturar os efeitos de sombreamento feitos por edifícios que bloqueiam a propagação de sinal. A interface do Veins pode ser verificada na Figura 2.10 (SOMMER, 2016).

Figura 2.10 – Simulador Veins.



Fonte <<http://veins.car2x.org/>>.

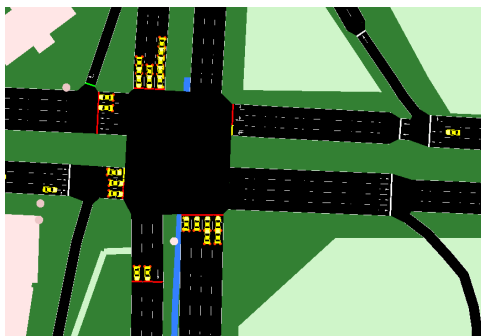
## 2.6 *Traces* de Mobilidade Veicular

As informações sobre trajetória, localização e velocidade de veículos são armazenados em um arquivo de dados denominados *trace*. O objetivo do uso dos *traces* é poder analisar a mobilidade real dos veículos durante trajetos em cidades reais. Esse tipo de dado se torna interessante para utilização, uma vez que fornecem o comportamento real dos veículos em cenários reais. Os *Traces* reais possuem informações coletadas de veículos equipados com algum dispositivo de localização, como o GPS.

Para criação destes *traces* reais, realiza-se a junção das informações coletadas pelos veículos com mapas reais da localidade onde foram coletadas tais informações e assim são utilizadas em um simulador de mobilidade. Como exemplo de *traces* disponíveis na literatura:

- Cidade de Luxemburgo: *trace* criado para representar o tráfego de deslocamento na Cidade de Luxemburgo, Luxemburgo. Criado a partir do modelo VehiLux e acrescido com o algoritmo de atribuição de rota Gawron (PIGNÉ; DANOY; BOUVRY, 2011).
- Cologne: o *trace* da cidade de Cologne, Alemanha, é um projeto chamado TAPAS Cologne, é um dos *trace* mais realistas devido a grande quantidade de informação que foi coletada. Todo o processo de criação deste *trace* esteve atrelado a ferramentas de simulação de mobilidade urbana já conhecidas, OpenStreetMap e SUMO. A Figura 2.11 ilustra o *trace* criado para cidade de Cologne (VARSCHEN; WAGNER, 2006).

Figura 2.11 – Trecho do *Trace* de Cologne



Fonte do Autor.

- São Francisco: este *trace* foi criado a partir da coleta de informação da mobilidade dos táxis na cidade de São Francisco, EUA. Em cada táxi foi implantando um sistema GPS e assim o *trace* contém as coordenadas de movimentação dos táxis a cada minuto, sendo possível criar uma simulação da movimentação dos veículos pela cidade (PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009).
- Roma: para este *trace* também foram utilizados táxis para determinar as rotas e movimentação de veículos na cidade de Roma, Itália. Foi utilizado o mesmo princípio de coleta de coordenadas para determinar a mobilidade dos veículos no centro de Roma. Porém, foi utilizado um tablet que realizava coletas periodicamente e enviava a um servidor central (BRACCIALE L.; RABUFFI, 2014).
- Xangai: o *trace* da cidade de Xangai, China, chamado de SUVnet, é realista e baseado na informação coletada de táxis equipados com algum dispositivo GPS, que possibilitou a criação do tráfego e mobilidade dos táxis pela cidade (UNIVERSITY, 2007).
- Zurique: desenvolvido pela ETH Zurich, este *trace* é extremamente realista da cidade de Zurique, Suíça. Contém o tráfego e mobilidade de automóveis públicos e privados, reproduz o comportamento da cidade em um período de 24 horas. Este *trace* cobre uma área de 250 km x 260 km, com registros de 260 mil veículos (NAUMOV; BAUMANN; GROSS, 2006).

Após a apresentação do embasamento teórico necessário para o desenvolvimento desse trabalho e projeto proposto, na seção seguinte, serão apresentados os Trabalhos Relacionados, utilizados como base e exemplos para o desenvolvimento do RODOGE.

### 3 TRABALHOS RELACIONADOS

Nesta seção são apresentados alguns trabalhos relacionados a disseminação de mensagens em redes veiculares e que foram utilizados como base para o desenvolvimento do projeto proposto. São apresentados: descrição, características, pontos positivos e negativos e carências em relação ao protocolo proposto.

#### ACORE

A aplicação *Algorithm for COLLision waRning Error correction* (ACORE) tem como objetivo emitir para o veículo um alerta relacionado a uma possível colisão com outro veículo parado a frente (NETO et al., 2016). Periodicamente os veículos enviam mensagens a seus vizinhos com informações sobre sua posição geográfica e velocidade. Quando um veículo recebe uma nova mensagem ele calcula a distância com base na posição recebida e na sua posição, comparando o resultado com a distância segura de frenagem, calculada com base na fórmula de (CHEN; SHEN; WANG, 2013). Se na comparação o valor da distância entre os veículos for menor que a distância segura de frenagem é emitido um alerta de colisão para o motorista. O problema dessa aplicação é que apenas alerta os veículos com uma possível colisão para um veículo a frente.

#### ADD

O protocolo *Data Dissemination Solution for Highly Dynamic Environments* (ADD) possui controle para diminuir a quantidade de mensagens na rede com base na seleção de veículos em zonas de preferência para realizar a disseminação da mensagem (SOUZA; MAIA; VILLAS, 2014). Os veículos presentes na zona de preferência e mais distantes do veículo remetente retransmitem o alerta de acidente com um atraso menor que os demais veículos e estes ao receberem essas mensagens cancelam os seus reenvios. A zona de preferência é definida por veículos mais propensos a disseminar a mensagem aos seus vizinhos. Este protocolo não trata dois veículos geradores de mensagens de alerta em um mesmo acidente, nem de dois ou mais acidentes.

#### ADDHV

O protocolo *Autonomic Data Dissemination in Highway for VANETs* (ADDHV) tem como objetivo diminuir a quantidade de mensagens disseminadas, evitando a *broadcast storm*,

mantendo a área de cobertura da mensagem e diminuindo o tempo de atraso (MENEGUETTE et al., 2014). Para maximizar a capacidade de disseminação o protocolo utiliza de dois meios: (i) prioriza o reenvio de mensagens por veículos que estejam na área de interesse e (ii) utiliza computação autônoma para decidir se um dado veículo deve reenviar a mensagem ou não, com base em sua eficiência de propagação. No primeiro meio os veículos dentro da área de interesse recebem tempo para reenvio menor que os veículos fora dessa área. No segundo meio a eficiência de propagação é calculada, periodicamente, com base na quantidade de mensagens transmitidas pela quantidade de mensagens recebidas e um veículo deve retransmitir a mensagem quando atender essa eficiência. Este protocolo não trata dois veículos geradores de mensagens de alerta em um mesmo acidente.

### **AID**

O protocolo *Approach for Information Dissemination* (AID) possui controle para diminuir a quantidade de mensagens reenviadas na rede com base no número de mensagens recebidas, utilizando um contador (BAKHOUYA; GABER; LORENZ, 2011). O controle existente no AID está relacionado a quantidade de mensagens recebidas, na qual ele decide reenviar ou não com base na quantidade de mensagens duplicadas recebidas pelo veículo em um intervalo de tempo. Essa decisão está relacionada ao resultado do cálculo que utiliza o tempo de espera para recebimento da mensagem, a quantidade de mensagens duplicadas neste tempo de espera e o intervalo de tempo de cada mensagem recebida, para resultados superiores a 0 há reenvio, caso contrário não reenvia. O problema do AID está relacionado a partição da rede, que ocorre quando o número de veículos na área de interesse não é suficiente para realizar a retransmissão. O problema deste protocolo está relacionado a partição da rede, que pode ocasionar não entrega da mensagem a veículos interessados.

### **DBRS**

O protocolo *Distance Based Relay Selection* (DBRS) possui controle para retransmissão de mensagem, na qual o tempo para reenvio é baseado na distância entre veículo remetente e veículo receptor (KIM et al., 2008). A distância calculada é utilizada para calcular o tempo de retransmissão, o tempo para o veículo retransmitir deve ser inversamente proporcional a distância entre os dois veículos. Este cálculo faz com que veículos mais distantes do remetente recebam intervalos de tempo menores e um veículo ao receber uma mensagem cancela o seu re-

envio. Com essa premissa o protocolo prioriza o reenvio de mensagens por veículos que estejam mais a borda do veículo remetente. O problema do DBRS está relacionado ao cancelamento desordenado das retransmissões ao receber uma mensagem duplicada. O problema deste protocolo está relacionado ao possível alto atraso de disseminação e a cobertura pode ser reduzida devido aos cancelamentos de reenvios.

## **DECA**

O protocolo *Density-Aware Reliable Broadcast* (DECA) possui controle de reenvio de mensagens, na qual verifica um veículo com maior número de vizinhos para realizar a retransmissão (NAKORN; ROJVIBOONCHAI, 2010). O protocolo realiza troca de *beacons* entre os veículos de um salto e assim determina a densidade local. Os veículos possuem uma lista, sempre atualizada, de identificação de vizinhos de um salto, com suas densidades, e uma lista de mensagens, com tempo de espera e as mensagens, que podem ser removidas caso o tempo expire ou a mensagem seja recebida novamente. O veículo com maior valor de densidade deve realizar o reenvio imediatamente e os demais veículos aguardam, se durante um tempo não houver o reenvio, outro veículo é selecionado pelo mesmo critério e se houver o reenvio, os veículos que a recebem e tenham mensagens agendadas para reenvio devem cancelá-las. Este protocolo depende da troca de *beacons* entre os veículos, o que pode acarretar, ainda, em um número elevado de mensagens trocadas.

## **DRIVE**

O protocolo *Data Dissemination Protocol in Vehicular Networks* (DRIVE) possui controle de retransmissões com base na escolha de veículos que estejam em um ponto ideal, *sweet spot*, para realizar a disseminação (VILLAS et al., 2014). O *sweet spot* é definido como o local em que os veículos são mais propícios a realizar a retransmissão a mais veículos que não receberam a mensagem anterior, assim possuem tempo de espera menor que os demais para realizar o reenvio. O protocolo DRIVE realiza a disseminação das mensagens de três maneiras: (i) quando não há partição na rede, (ii) quando a área de interesse está particionada e (iii) para disseminação em ambientes rodoviários. Na primeira maneira, a disseminação ocorre com os veículos dentro do ponto ideal, sendo selecionadas para realizar a retransmissão. Na segunda maneira, ao detectar uma rede particionada, os veículos fora do *sweet spot* que são utilizados para realizar as retransmissões, estes veículos estão na chamada zona de recuperação. Na ter-

ceira maneira, verifica antes do cancelamento de reenvio a sua distância em relação a origem da mensagem e ao remetente da mensagem. Quando um veículo possui mensagem agendada para reenvio e recebe a mensagem novamente, este deve cancelá-la. Este protocolo não trata dois veículos geradores de mensagens de alerta em um mesmo acidente.

## **DVCAST**

O protocolo *Distributed Vehicle Broadcast* (DV-CAST) possui controle nos reenvios das mensagens baseado na posição, distância e topologia local (TONGUZ; WISITPONGPHAN; BAI, 2010). A topologia local é definida pela troca periódica de *beacons* entre veículos de apenas um salto. A posição e a distância são utilizadas para determinar a direção e fluxo dos veículos vizinhos. A seleção de possível veículo a realizar a retransmissão da mensagem é realizada através da análise da topologia local, que verifica se é um veículo destinatário, se há mais veículos movendo na mesma direção que podem realizar o reenvio da mensagem e se o veículo está conectado a pelo menos mais um outro veículo. A retransmissão ocorre de acordo com esta análise topológica realizada e em casos que o veículo recebe a mensagem novamente antes de realizar o seu envio, ele deve cancelar e agendar uma nova mensagem. Este protocolo depende da troca de *beacons* entre os veículos, o que pode acarretar, ainda, em um número elevado de mensagens trocadas.

## **Flooding**

O protocolo base de disseminação de mensagens é o Flooding (XEROS et al., 2010). A ideia é disseminar as mensagens a todos os vizinhos, que armazenam a mensagem e também reenviam a mensagem a todos os vizinhos e assim por diante. Embora seja a forma mais rápida de disseminar uma mensagem em uma rede veicular, neste protocolo há uma grande quantidade de mensagens reenviadas e assim ocorre o problema de *broadcast storm*. Esta forma de disseminação não possui controles de reenvios, tratamento para mais de um veículo acidentado, comparação entre as mensagens.

## **GEDDAI**

O protocolo *Gographical Data Dissemination for Alert Information* (GEDDAI) utiliza como forma de controle para disseminação de mensagens a seleção de veículos em pontos ideais para realizarem a retransmissão da mensagem (VILLAS et al., 2012). O ponto ideal é definido

pela divisão da área de cobertura de um dado veículo em quatro quadrantes e pela angulação e distância do veículo remetente. O veículo receptor tem conhecimento se está presente ou não no ponto ideal, caso esteja o veículo deve realizar a retransmissão. Em casos que o ponto ideal esteja vazio, o veículo mais distante do veículo remetente deve realizar a retransmissão. Sempre que um veículo recebe uma mensagem, este cancela o reenvio de uma possível mensagem agendada. Este protocolo não trata dois veículos geradores de mensagens de alerta em um mesmo acidente.

A Tabela 3.1 compara as estratégias de reenvio de mensagens dos protocolos de disseminação levantados. São comparados os diferentes tipos de protocolos encontrados e seus objetivos, como:

Tabela 3.1 – Comparação entre Protocolos de Disseminação

| Protocolo | Estratégia de Reenvio |         |          |           |             |                |           | Objetivos       |                  |             |
|-----------|-----------------------|---------|----------|-----------|-------------|----------------|-----------|-----------------|------------------|-------------|
|           | Atraso                | Cluster | Contador | Distância | Localização | Probabilístico | Topologia | Broadcast Storm | Partição da Rede | N Acidentes |
| Acore     |                       |         |          | ×         | ×           |                |           |                 |                  |             |
| ADD       | ×                     |         |          | ×         | ×           |                |           | ×               | ×                |             |
| ADDHV     | ×                     |         |          |           |             |                |           | ×               | ×                |             |
| AID       |                       |         | ×        |           |             | ×              |           | ×               |                  |             |
| DBRS      | ×                     |         |          | ×         | ×           |                |           | ×               |                  |             |
| DECA      | ×                     |         |          |           |             | ×              | ×         | ×               | ×                |             |
| DRIVE     | ×                     |         |          | ×         | ×           |                |           | ×               | ×                |             |
| DVCAST    | ×                     |         |          | ×         | ×           |                | ×         | ×               | ×                |             |
| Flooding  |                       |         |          |           |             |                |           |                 |                  |             |
| GEDDAI    | ×                     |         |          | ×         | ×           |                |           | ×               | ×                |             |

Fonte do Autor.

Ao analisar a Tabela 3.1 é possível observar diferentes protocolos que tratam a *Broadcast Storm* utilizando diferentes estratégias e até mesmo unindo elas de forma híbrida. E também quais são os objetivos dos mesmos protocolos, porém nenhum deles trata diferentes quantidades de veículos acidentados e acidentes.



O protocolo desenvolvido neste trabalho, RODOGE, que utiliza de regras e funções para mitigar o problema de *Broadcast Storm* e a inundação da rede é apresentado no próximo Capítulo.

## 4 PROTOCOLO DE DISSEMINAÇÃO DE MENSAGENS DE ALERTA

Este projeto consiste na elaboração de um protocolo de disseminação de mensagens de alerta para redes veiculares chamado RODOGE (*Rules that Oppose overdone Dissemination Of messaGEs*), que utiliza como forma de disseminação o *broadcast*, ou difusão. O protocolo tem como fundamento a disseminação de mensagens de alerta de acidente para todos os veículos interessados e dentro da área crítica do acidente, tal área é a distância limite para reação do motorista e frenagem do veículo. O protocolo visa a entrega a todos os veículos alvos da rede e visa principalmente o controle de mensagens para evitar a *broadcast storm*, ou mesmo inundação da rede.

### 4.1 RODOGE - *Rules that Oppose overdone Dissemination Of messaGEs*

O protocolo RODOGE tem como objetivo a entrega de mensagem a todos os veículos e principalmente o controle no envio de mensagens em *broadcast*, de modo a evitar a inundação da rede. As mensagens na rede são enviadas em forma de difusão a todos os veículos vizinhos. Para atingir tais objetivos e obter os resultados desejados foram criadas regras e funções no protocolo para prevenir a ocorrência do *broadcast storm*, diminuindo o número de mensagens enviadas.

São utilizadas de regras e funções que visam controlar o reenvio de mensagem e assim diminuir a quantidade de mensagens na rede, sendo assim o protocolo busca a prevenção dos problemas gerados com a *broadcast storm*. RODOGE tem o funcionamento baseado em uma máquina de estados, na qual cada função do protocolo é um estado dessa máquina, e realiza as devidas tarefas para suas funcionalidades, e as transições realizadas no decorrer do funcionamento são de acordo as decisões a serem atendidas ou não, sendo tomados caminhos distintos de acordo com os resultados.

As regras para a redução do reenvio das mensagens são baseadas em premissas que consideram o tipo de comunicação, V2V, V2I ou V2X. Para comunicação V2V as seguintes premissas são consideradas: (i) validar mensagem, mensagem na tabela deve ter identificador único e posição geográfica diferente, (ii) tempo de vida válido, tempo de vida da mensagem deve ser superior e inversamente proporcional a velocidade do veículo, (iii) remetente da mensagem, na área de dispositivos de infraestrutura não há disseminação de mensagens pelos veículos, (iv) *backoff* da mensagem, o *backoff* deve ser proporcional à quantidade de mensagens recebidas pelos veículos, e (v) distância do remetente, o tempo final para reenvio da mensagem é inver-

samente proporcional à distância calculada. Na comunicação V2I somente algumas premissas, com os mesmo fundamentos, são consideradas: (i) validar mensagem e (ii) tempo de vida da mensagem.

#### 4.1.1 Regras de Reenvio de Mensagem

Para que o funcionamento do protocolo ocorra corretamente e obtenha os resultados desejados, são utilizadas regras de controle para reenvios da mensagens de alerta na rede. As regras são utilizadas para gerenciar os reenvios da mensagem na rede, controlam quais nós e quando estes devem reenviar a mensagem.

##### Regra #1: Validar Mensagem

Esta regra do protocolo está relacionada a comparação da mensagem recebida, na qual é determinado se: (i) é uma mensagem nova; (ii) se é uma mensagem existente; ou (iii) se é uma mensagem de mesmo acidente. Tais contatações são obtidas ao realizar comparação entre as mensagens armazenadas localmente, tabela de mensagens, em cada nó e a mensagem recebida.

No primeiro caso, quando a mensagem não está presente na tabela de mensagens do nó receptor, ela é adicionada à sua tabela. Alguns campos na tabela são preenchidos com os valores recebidos na mensagem e outros são preenchidos com valores do nó que a recebeu. E apenas nos veículos é iniciado o incremento do *backoff*.

No segundo caso, quando é uma mensagem existente na tabela de mensagens, a mensagem recebida possui identificador igual a alguma mensagem armazenada na tabela, os campos dessa mensagem devem ser atualizados. São atualizados os campos de tipo do remetente, posição do nó remetente, tempo de recebimento e é incrementado o valor de *backoff*. Os demais campos são mantidos na tabela de mensagens sem alterações.

No terceiro caso, quando é uma mensagem de mesmo acidente, a mensagem possui número de identificação diferente com as armazenadas na tabela. Ao realizar o cálculo de distância entre a origem das mensagens da tabela e a origem da mensagem recebida é constatado que as mensagens são de um mesmo acidente, mas com veículos diferentes. Após tal constatação, a mensagem deve ser atualizada na tabela, semelhante ao que ocorre quando é uma mensagem existente na tabela.

##### Regra #2: Tempo de Vida Válido

A segunda regra do protocolo é para que não existam mensagens circulando infinitamente na rede. Para evitar este problema de serem reenviadas desnecessariamente, cada men-

sagem tem seu tempo de vida calculado. Isso evita a circulação de mensagens ociosas na rede e que as mensagens sejam enviadas sobre um evento que tenha terminado.

O tempo de vida da mensagem é calculado com base no tempo que o veículo levaria para atravessar toda a área de cobertura do dispositivo de infraestrutura, o diâmetro. Para realizar o cálculo é utilizada a fórmula da Equação 4.1.

$$velocidade = \frac{diâmetro}{tempo} \Rightarrow \mathbf{tempo} = \frac{diâmetro}{velocidade} \quad (4.1)$$

No cálculo do tempo é utilizado a fórmula de cálculo de velocidade de um corpo, na qual utiliza deslocamento e o tempo para realizar tal deslocamento. Para obter o tempo como resultado há uma inversão dos valores e assim é utilizado o diâmetro da área de cobertura do dispositivo de infraestrutura, e a velocidade máxima do veículo acidentado. Com este cálculo é obtido o tempo que o veículo levaria para atravessar a área de cobertura do dispositivo de infraestrutura e este resultado é o valor do tempo de vida da mensagem.

Com este tempo de vida da mensagem calculado pelo veículo acidentado e reenviado aos demais nós, sempre que uma mensagem é preparada para reenvio é analisado se esta possui de tempo de vida válido e reenvia apenas em casos em que o tempo não tenha expirado.

### **Regra #3: Remetente da Mensagem**

A terceira regra do protocolo, presente apenas nos veículos, é para evitar o reenvio de mensagens em áreas com presença de dispositivo de infraestrutura. Os veículos ao entrarem em contato com um desses dispositivos devem interromper a retransmissão. Tal medida faz com que ocorra diminuição no número de mensagens enviadas nesta região, já que apenas o dispositivo de infraestrutura realiza reenvio da mensagem nessa área.

Para adotar tal funcionalidade é levado em conta que um dispositivo de infraestrutura não possui mobilidade e que possui maior poder de processamento e de comunicação.

A constatação de que uma mensagem tem como remetente um dispositivo de infraestrutura ocorre no momento de sua preparação para reenvio. É verificado o tipo de nó remetente da mensagem e em caso de dispositivo de infraestrutura é interrompida e em caso de veículo a mensagem segue para ser reenviada.

Para os casos em que o veículo perde o contato com o dispositivo de infraestrutura, o veículo deve voltar a retransmitir a mensagem. Há uma verificação do momento em que foi recebida a última mensagem do dispositivo de infraestrutura e caso o tempo tenha expirado o veículo deve voltar a retransmitir.

#### Regra #4: *Backoff* da Mensagem

Esta regra do protocolo, presente apenas nos veículos, é utilizada como base para a densidade da mensagem na rede. Quanto maior o número de mensagens semelhantes existir numa dada região, o seu valor de *backoff* será maior. Ao receber uma mensagem o valor é incrementado e ao enviar uma mensagem o valor é decrementado.

O *backoff* é utilizado para aumentar o intervalo de tempo entre os reenvios da mensagem. Uma mensagem densa na rede deve ser reenviada com um intervalo de tempo maior e uma mensagem escassa na rede deve ser reenviada com menor tempo.

Para calcular o intervalo de envio das mensagens com base no *backoff* é utilizado o tempo padrão de reenvios e o valor de *backoff* para a mensagem, como pode ser observado na fórmula da Equação 4.2.

$$timeBackoff = timeMensagem * backoff \quad (4.2)$$

#### Regra #5: Distância do Remetente

A quinta regra do protocolo, presente apenas nos veículos, é utilizada com base na distância entre o nó remetente e do nó receptor. Ao calcular a distância é invertido os valores para priorizar o reenvio de nós mais distantes. Nós mais ao extremo da área de cobertura recebem valores menores e mais no interior recebem valores maiores.

Essa regra é utilizada para calcular o tempo final para reenvio da mensagem, que utiliza o tempo de envio calculado na regra anterior e o inverso da distância calculada, como pode ser observado na fórmula da Equação 4.3. É utilizado o raio de alcance da comunicação dos nós dividido pela distância calculada e somado a 1, para em casos de veículos fora da área de cobertura receberem um valor menor que o tempo padrão de reenvio da mensagem.

$$timeFinal = timeBackoff * \left( \frac{raio}{distância} + 1 \right) \quad (4.3)$$

Após este cálculo é definido o tempo final para reenvio da mensagem pelos veículos. Os dispositivos de infraestrutura por não possuírem essas últimas regras reenviam com base no tempo padrão. Com essas regras, *backoff* e distância, o reenvio prioriza locais na qual há pouca quantidade de mensagens e com uma distância maior do nó remetente.

As funções do protocolo RODOGE são baseadas nas regras citadas acima e são diferentes tipos de nós. Para os veículos são utilizadas seguintes funções: (i) *CheckStatus*, (ii)

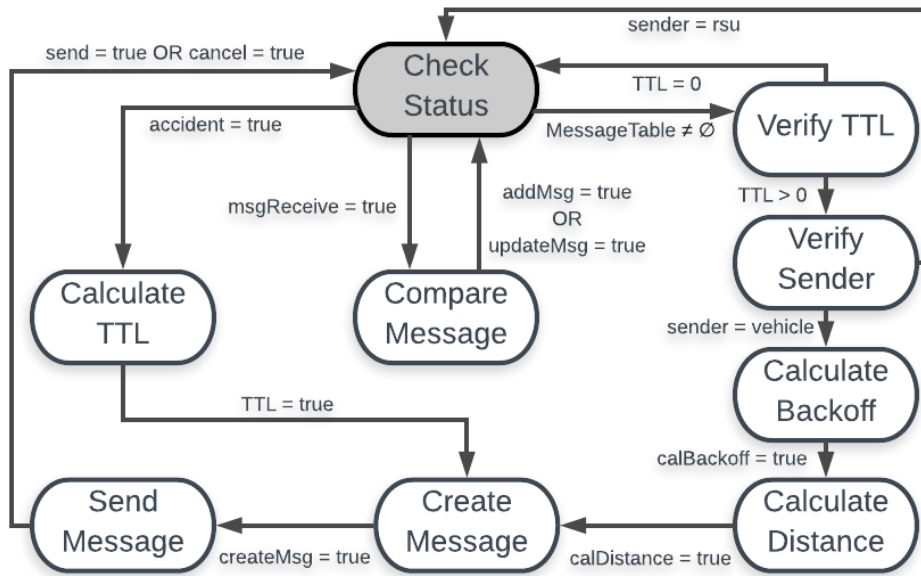
*CalculateTTL*, (iii) *CompareMessage*, (iv) *VerifyTTL*, (v) *VerifySender*, (vi) *CalculateBackoff*, (vii) *CalculateDistance*, (viii) *CreateMessage* e (ix) *SendMessage*. Para os dispositivos de infraestrutura são utilizadas as seguintes funções: (i) *CheckStatus*, (ii) *CompareMessage*, (iii) *VerifyTTL*, (iv) *CreateMessage* e (v) *SendMessage*.

#### 4.1.2 Diagrama de Funcionamento

O funcionamento do protocolo segue uma sequência similar a uma máquina de estados, na qual cada estado é uma função ou ação do protocolo e cada transição é uma tomada de decisão. As tomadas de decisões definem qual caminho e ação é tomada na execução do protocolo de acordo com as saídas das funções. Há diferenças entre as máquinas de estado dos carros e dos dispositivos de infraestrutura, pois algumas funcionalidades do protocolo presente nos veículos não são necessárias nos dispositivos de infraestrutura.

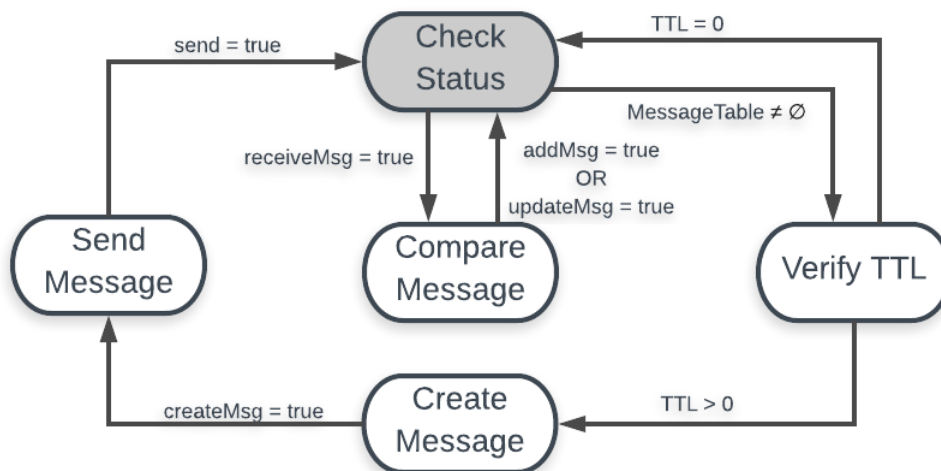
Na Figura 4.1 é apresentada a máquina de estados dos veículos, similar ao funcionamento das funções implementadas neles. O estado ou função inicial do protocolo é o *CheckStatus* que abrange alguns funções do protocolo, como detectar um acidente e calcular o intervalo padrão de reenvio das mensagens. Em *CheckStatus* é verificado se: (i) o veículo sofreu um acidente, (ii) recebeu uma mensagem ou (iii) há mensagens a serem reenviadas. Caso alguma verificação no *CheckStatus* seja verdadeira a transição é para *CalculateTTL*, *CompareMessage* ou *VerifyTTL*, respectivamente. Em *CalculateTTL* é calculado o tempo de vida da mensagem a ser enviada e após o cálculo segue para *CreateMessage*. Em *CompareMessage* é analisada a mensagem recebida: se é uma mensagem nova deve ser adicionada na tabela ou se é uma mensagem existente na tabela deve ser atualizada e em ambos os casos retorna para *CheckStatus*. Em *VerifyTTL* é analisado se a mensagem a ser enviada possui tempo de vida maior que 0, em caso verdadeiro segue para *VerifySender* e em caso falso retorna para *CheckStatus*. Em *VerifySender* é analisado o remetente da mensagem, se é um veículo segue para *CalculateBackoff* e se é um dispositivo de infraestrutura retorna para *CheckStatus*. Em *CalculateBackoff* é calculado o intervalo de reenvio com base no *backoff* da mensagem, e segue para *CalculateDistance*. Em *CalculateDistance* é calculado o intervalo de reenvio com base na distância do remetente, e segue para *CreateMessage*. Em *CreateMessage* os campos da mensagem a ser enviada são preenchidos com dados do veículo, caso seja um veículo acidentado, ou com dados da tabela de mensagens e segue para *SendMessage*. Em *SendMessage* a mensagem é escalonada para o reenvio e segue para *CheckStatus* caso seja enviada ou caso seja cancelada.

Figura 4.1 – Máquina de Estados do Veículo



Fonte do Autor.

Figura 4.2 – Máquina de Estados do Dispositivo de Infraestrutura



Fonte do Autor.

Na Figura 4.2 é apresentada a máquina de estados dos dispositivos de infraestrutura, similar ao funcionamento das funções implementadas neles. O estado ou função inicial do protocolo é o *CheckStatus* que verifica se recebeu uma mensagem ou se há mensagens a serem reenviadas, em caso verdadeiro em algum desses a transição segue para *CompareMessage* ou *VerifyTTL*, respectivamente. Em *CompareMessage* é analisada a mensagem recebida: se é uma mensagem nova deve ser adicionada na tabela ou se é uma mensagem existente na tabela deve ser atualizada e em ambos os casos retorna para *CheckStatus*. Em *VerifyTTL* é analisado se a mensagem a ser enviada possui tempo de vida maior que 0, em caso verdadeiro segue para

*CreateMessage* e em caso falso retorna para *CheckStatus*. Em *CreateMessage* os campos da mensagem a ser enviada são preenchidos com dados da tabela de mensagens e segue para *SendMessage*. Em *SendMessage* a mensagem é escalonada para o reenvio e segue para *CheckStatus* caso seja enviada.

#### 4.1.3 Estrutura da Mensagem Enviada na Rede

Cada campo da mensagem é utilizado para realizar alguma operação necessária no decorrer do protocolo. Os campos presentes na estrutura da mensagem também estão presentes na estrutura da tabela de mensagens. Os campos da mensagem possuem os seguintes atributos e sua estrutura pode ser visualizada na Figura 4.3.

Figura 4.3 – Estrutura da Mensagem Enviada

|                  |                    |                 |             |                     |             |            |             |
|------------------|--------------------|-----------------|-------------|---------------------|-------------|------------|-------------|
| <b>idMessage</b> | <b>messaOrigin</b> | <b>position</b> | <b>time</b> | <b>lastPosition</b> | <b>type</b> | <b>TTL</b> | <b>hops</b> |
|------------------|--------------------|-----------------|-------------|---------------------|-------------|------------|-------------|

Fonte do Autor.

1. *idMessage*: número de identificação da mensagem, utilizado para comparar as mensagens e não adicionar mensagens repetidas na tabela;
2. *messageOrigin*: número de identificação do nó origem, utilizado para identificar qual o nó de origem da mensagem;
3. *position*: coordenadas geográficas do nó origem, utilizado para calcular a distância entre dois acidentes e verificar se é o mesmo acidente;
4. *time*: tempo da mensagem, utilizado para calcular a latência da mensagem;
5. *lastPosition*: coordenadas geográficas do nó remetente, utilizado para calcular a distância do nó remetente e nó receptor;
6. *type*: tipo do remetente da mensagem, utilizado para identificar se o último remetente da mensagem é um veículo ou dispositivo de infraestrutura;
7. *TTL*: tempo de vida da mensagem, utilizado para armazenar o tempo de duração da mensagem na rede;

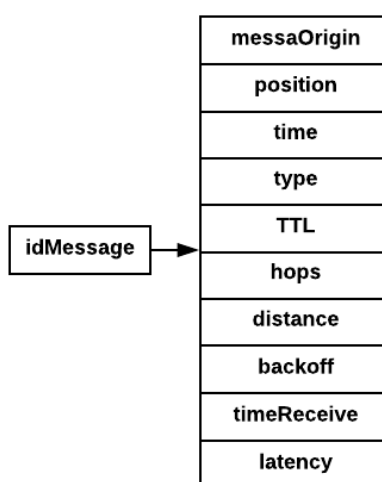


8. *hops*: número de saltos da mensagem, utilizado para verificar o número de nós intermediários.

#### 4.1.4 Estrutura da Tabela de Mensagens

A tabela de mensagens armazena as mensagens recebidas, e posteriormente, é utilizada para recriar a mensagem em um possível reenvio da mesma. Possui os mesmos campos da estrutura das mensagens e mais quatro campos para uso interno do próprio nó. Sua estrutura pode ser visualizada na Figura 4.4.

Figura 4.4 – Estrutura da Tabela de Mensagens



Fonte do Autor.

1. A estrutura da tabela de mensagens é baseada em um *hashmap*, na qual o *idMessage* da mensagem é a chave do mapa e os demais valores são para preenchimento da tabela com as devidas variáveis;
2. *timeReceive*: tempo de recebimento da mensagem, utilizado para verificar se há recebimento recente;
3. *latency*: tempo de recebimento da mensagem, utilizado para verificar a latência;
4. *backoff*: valor de densidade, utilizado para calcular o tempo de reenvio;
5. *distance*: distância do nó remetente, utilizado para calcular o tempo final de reenvio.

#### 4.1.5 Funções do Protocolo

O protocolo RODOGE foi desenvolvido com base nas regras e em diversas operações que são executadas por meio de funções que controlam o reenvio de mensagens e evitam a inundação da rede. Essas funções do protocolo são descritas a seguir:

- *CheckStatus*: analisa se há ocorrência de um acidente (para veículos), se há mensagem recebida ou se há mensagem a ser enviada;
- *CalculateTTL*: calcula o tempo de vida máximo da mensagem da mensagem a ser enviada;
- *CompareMessage*: implementada com base na Regra 1, compara mensagem recebida com as demais mensagens presentes na tabela de mensagens;
- *VerifyTTL*: implementada com base na Regra 2, verifica qual é o tempo de vida da mensagem a ser enviada;
- *VerifySender*: implementada com base na Regra 3, verifica qual é o remetente da mensagem a ser enviada;
- *CalculateBackoff*: implementada com base na Regra 4, calcula o intervalo de reenvio com base no valor de *backoff* da mensagem;
- *CalculateDistance*: implementada com base na Regra 5, calcula o intervalo de reenvio com base na distância do remetente da mensagem;
- *CreateMessage*: preenche os campos da mensagem a ser enviada;
- *SendMessage*: realiza o envio da mensagem na rede ou seu cancelamento.

##### Função #1 *CheckStatus*

A função para detectar se um veículo sofreu um acidente analisa o estado do *airbag*, sua velocidade e o tempo em que o veículo permaneceu parado. Além disso, armazena a maior velocidade que o veículo atingiu antes do acidente. Um veículo acidentado pode ser definido como sendo um veículo com alteração no estado do *airbag* ou com uma desaceleração igual ou superior a  $61.1\text{m/s}^2$  (BARCELOS, 2014).

**Algorithm 1** Detectar Acidente

---

```

1: function ACIDENTDETECT
2:   velocidade ← getMaxSpeed
3:   if getSpeed < 1 and stop >= 10 and airbag = true then
4:     calculateTTL(velocidade)
5:     createAlert()

```

---

No Algoritmo 1 na linha 2 a variável *velocidade* recebe valor referente a maior velocidade que o veículo atingiu. Na linha 3 é verificado se a velocidade do veículo é menor que 1, se o tempo de parada do veículo é maior do que o normal e se o estado do *air-bag* é verdadeiro, caso todas as condições sejam atendidas é constatado um acidente. Na linha 4 é calculado o tempo de vida da mensagem com base na velocidade máxima do veículo. Na linha 5 é criada a mensagem de alerta de acidente.

Além de detectar um acidente, em *CheckStatus* há o cálculo do tempo padrão para re-envio das mensagens baseado no tempo de reação do motorista e na frenagem do veículo. Ao definir este tempo é garantido que o veículo recebe a mensagem antes de entrar na área crítica do acidente, que é a distância de segurança. As distâncias e tempos são alterados de acordo com a variação da velocidade, como pode ser observado na Tabela 4.1 (CHEN; SHEN; WANG, 2013).

Tabela 4.1 – Distâncias de Reação e Frenagem

| Velocidade | Distância de Reação | Distância de Frenagem | Distância Total |
|------------|---------------------|-----------------------|-----------------|
| 40km/h     | 11.11m              | 8.00m                 | 19.11m          |
| 50km/h     | 13.89m              | 12.50m                | 26.39m          |
| 60km/h     | 16.67m              | 18.00m                | 34.67m          |
| 70km/h     | 19.44m              | 24.50m                | 43.94m          |
| 80km/h     | 22.22m              | 32.00m                | 54.22m          |
| 90km/h     | 25.00m              | 40.50m                | 65.50m          |
| 100km/h    | 27.78m              | 50.00m                | 77.78m          |
| 110km/h    | 30.56m              | 60.50m                | 91.06m          |

Fonte do Autor.

O tempo padrão de envio das mensagens é calculado com base na área de cobertura do dispositivo de infraestrutura, na velocidade máxima da via, na distância para reação e frenagem para tal velocidade. Com esses valores é possível calcular o tempo gasto para um veículo atravessar a área de cobertura de um dispositivo de infraestrutura, fórmula da Equação ?? e posteriormente o tempo utilizado como padrão para intervalo entre as mensagens, na fórmula da Equação 4.4.

$$\begin{aligned} \text{tempoMsg} &= \frac{\text{distânciaSegura} * \text{tempo}}{\text{diâmetro}} \Rightarrow \text{tempoMsg} = \frac{\text{distânciaSegura} * \frac{\text{diâmetro}}{\text{velocidade}}}{\text{diâmetro}} \Rightarrow \\ \text{tempoMsg} &= \frac{\text{distânciaSegura}}{\text{velocidade}} \end{aligned} \quad (4.4)$$

A Fórmula 4.1 é utilizada para calcular o tempo que um veículo necessita para atravessar a área de cobertura do dispositivo de infraestrutura com a velocidade máxima da via. A Fórmula 4.4, que utiliza com princípio a Fórmula 4.1 como base, calcula o tempo padrão de reenvio com base na distância seguro e no diâmetro de cobertura. Seguindo a Fórmula 4.4 há uma substituição do tempo para diâmetro e velocidade e posteriormente uma abreviação da Fórmula 4.4, na qual utiliza apenas a distância segura e a velocidade do veículo.

### **Função #2** *CalculateTTL*

Esta função realiza o cálculo do tempo de vida da mensagem de acordo com o diâmetro de cobertura e a velocidade máxima do veículo acidentado. É utilizada a função de cálculo de velocidade para calcular o tempo que um veículo gasta para atravessar a área de cobertura de um nó.

---

#### **Algorithm 2** *CalculateTTL*

---

```

1: function CALCULATETTL(id)
2:   TTLcalculate ← diametro/maxSpeed
3:   return TTLcalculate

```

---

No Algoritmo 2, na linha 2, é calculado o tempo de vida da mensagem, utilizando o diâmetro da área de cobertura, dividido pela velocidade máxima do veículo. Na linha 3 a função retorna o valor calculado do tempo de vida da mensagem.

### **Função #3** *CompareMessage*

Essa função realiza a comparação entre mensagens recebidas e mensagens armazenadas na tabela de mensagens. Tal comparação evita que existam mensagens repetidas e mensagens de um mesmo acidente sejam armazenadas.

No Algoritmo 3, na linha 2 é verificado se a tabela de mensagens do nó está vazia, em caso verdadeiro a mensagem deve ser adicionada na tabela sem outras comparações, preenchendo os campos similares com dados obtidos na mensagem recebida e os demais campos com dados obtidos pelo nó. Para os casos em que a tabela não está vazia, a linha 5 busca na

tabela as chaves de identificação das mensagens armazenadas. Na linha 6 é verificado se a mensagem recebida está armazenada na tabela. Em caso verdadeiro, na linha 7, a mensagem armazenada deve ser atualizada, na linha 8 o *backoff* deve ser incrementado e na linha 9 deve encerrar a busca na tabela por ter achado uma mensagem correspondente. Na linha 10, caso a mensagem não esteja armazenada na tabela, a distância entre o veículo origem da mensagem recebida e os veículos origem de outras mensagens é calculada na linha 11. Na linha 12, para casos de um acidente diferente, na qual a distância calculada é maior que a distância segura, a mensagem deve ser adicionada na tabela na linha 13. Na linha 14, para caso de uma mensagem de mesmo acidente estar presente na tabela, ela deve ser atualizada na linha 15 e o *backoff* deve ser incrementado na linha 16.

---

**Algorithm 3** CompareMessage
 

---

```

1: function COMPAREMESSAGE
2:   if messageTable.empty then
3:     messageTable ← addMessage
4:   else
5:     for (messageTable.begin; messageTable.end; findKeyTable) do
6:       if IdMessage = KeyTable then
7:         messageTable ← updateMessage
8:         messageTable.Backoff ++
9:         messagemessageTable.end
10:      else
11:        calculateDistance
12:        if distance > safeDistance then
13:          messageTable ← addMessage
14:        else
15:          messageTable ← updateMessage
16:          messageTable.Backoff ++
  
```

---

**Função #4** *VerifyTTL*

Esta função que verifica o tempo de vida de uma mensagem que está armazenada na tabela de mensagens. Esta verificação é realizada para evitar o reenvio de mensagens que possuem tempo expirado.

---

**Algorithm 4** *VerifyTTL*


---

```

1: function VERIFYTTL(keyTable)
2:   if messageTable.TTL > 0 then
3:     verifySender(keyTable)
4:   else
5:     deleteMsg(keyTable)
  
```

---

No Algoritmo 4, na linha 2, é verificado se a mensagem possui tempo de vida maior que 0 ou não. Na linha 3, caso a mensagem possua tempo de vida válido, a mensagem segue para próxima função. Na linha 5, caso a mensagem possua tempo de vida inválido, a mensagem deve ser deletada.

#### **Função #5** *VerifySender*

Para evitar que veículos em área de cobertura de um dispositivo de infraestrutura realizem o reenvio, é implementada a função que verifica o remetente das mensagens antes de reenviar. Tal medida faz com que apenas os dispositivos de infraestrutura realizem o reenvio em suas áreas, fazendo com que ocorra diminuição de mensagens geradas. Após sair da área de cobertura do dispositivo de infraestrutura o veículo volta a reenviar a mensagem.

---

#### **Algorithm 5** *VerifySender*

---

```

1: function VERIFYSENDER(keyTable)
2:   if messageTable.Type = vehicle then
3:     calculateBackoff(keyTable)
4:   if messageTable.Type = RSU and Time > messageTable.TimeReceive + timeMsg then
5:     messageTable.Type = vehicle
6:     calculateBackoff(keyTable)

```

---

No Algoritmo 5, na linha 2, é verificado qual é o remetente da mensagem armazenada na tabela, caso seja um veículo a mensagem segue para próxima função na linha 3. Na linha 4 é verificado se o veículo perdeu o contato com o dispositivo de infraestrutura, verifica se o tempo decorrido é maior que o tempo de último recebimento da mensagem mais o tempo padrão de reenvio. Caso seja verdadeiro, o veículo perdeu o contato e deve retornar o reenvio, alterando o tipo do remetente da mensagem armazenada na tabela na linha 5 e segue para próxima função na linha 6.

#### **Função #6** *CalculateBackoff*

Esta função calcula o tempo de reenvio com base no *backoff* da mensagem. Este backoff está relacionado à quantidade de mensagens recebidas e de mensagens reenviadas. Tal cálculo faz com que em locais com maior número de mensagens os veículos recebam um intervalo de tempo maior para reenvio.

---

#### **Algorithm 6** *CalculateBackoff*

---

```

1: function CALCULATEBACKOFF(keyTable)
2:   timeBack = timeMsg * messageTable.Backoff
3:   calculateDistance(keyTable)

```

---

No Algoritmo 6, na linha 2, é calculado o tempo com base no *backoff* da mensagem, multiplicando o tempo padrão de reenvio da mensagem pelo valor relacionado a densidade da mensagem na rede, *backoff*. Na linha 3 segue para próxima função.

#### **Função #7** *CalculateDistance*

Esta função calcula o tempo de reenvio com base na distância entre o veículo remetente e o veículo receptor da mensagem. Com isso, o tempo para reenvio é inversamente proporcional a distância. Tal cálculo faz com que veículos mais ao extremo da área de cobertura do remetente recebam um intervalo menor para retransmissão.

---

#### **Algorithm 7** *CalculateDistance*

---

```

1: function CALCULATEDISTANCE(keyTable)
2:   calucaleDistance
3:   timeFinal = timeBack * (radius/distance) + 1
4:   createMessage(keyTable)

```

---

No Algoritmo 7, na linha 2, é calculada a distância entre o veículo remetente e o veículo receptor. Na linha 3 é calculado o tempo final para reenvio, que utiliza o tempo calculado anteriormente com base no *backoff* e na divisão entre o raio da área de cobertura e a distância calculado. Na linha 4 a mensagem segue para a próxima função.

#### **Função #8** *CreateMessage*

Esta função cria a mensagem a ser enviada na rede, seja uma mensagem de um novo acidente ou seja uma mensagem armazenada na tabela.

Para criar uma mensagem de um novo acidente, por um veículo acidentado, a função preenche os campos da mensagens com dados obtidos pelo próprio veículo acidentado, como pode ser observado no Algoritmo 8. Nas linhas 2 a 8 a mensagem é preenchida com valores obtidos pelo próprio veículo e na linha 9 a mensagem segue para ser enviada na próxima função.

---

#### **Algorithm 8** *CreateMessage*

---

```

1: function CREATEMESSAGE
2:   msg.IdMessage ← randon()
3:   msg.MessageOrigin ← vehicleId
4:   msg.Position ← coordanate()
5:   msg.Time ← time()
6:   msg.Type ← vehicle
7:   msg.TTL ← TTLcalculat
8:   msg.Hops ← 0
9:   SendMsg(msg)

```

---

Para criar uma mensagem de reenvio, a função preenche os campos da mensagem com dados correspondentes armazenados na tabela de mensagens. Utiliza a chave de identificação da mensagem e utiliza os dados referentes a ela armazenados na tabela. No Algoritmo 9, nas linhas 2 a 8 a mensagem é preenchida com valores armazenados na tabela por mensagens recebidas anteriormente, exceto na linha 6 que é preenchida com o tipo do remetente da mensagem. Na linha 9 a mensagem segue para ser enviada na próxima função.

---

**Algorithm 9** CreateMessage
 

---

```

1: function CREATEMESSAGE(keyTable)
2:   msg.IdMessage ← keyTable
3:   msg.MessageOrigin ← messageTable.MessageOrigin
4:   msg.Position ← messageTable.Position
5:   msg.Time ← messageTable.Time
6:   msg.Type ← nodeType
7:   msg.TTL ← messageTable.TTL
8:   msg.Hops ← messageTable.Hops
9:   SendMsg(msg)

```

---

**Função #9** *SendMessage*

Após a passagem em todas as funções anteriores e nos controles de reenvios presentes para cada tipo de dispositivo, a mensagem enfim, deve ser reenviada. Para os dispositivos de infraestrutura, enquanto a mensagem possuir tempo de vida válido e assim que o tempo para reenvio é expirado a mensagem é reenviada. Para os veículos há diferenças, pois em casos que o veículo não tenha enviado a mensagem e receber uma cópia dela, a mensagem deve ser cancelada.

---

**Algorithm 10** SendMessage
 

---

```

1: function SENDMSG(keyTable)
2:   if Time ≥ messageTable.TimeReceive + timeCal and messageTable.TTL > 0 then
3:     messageTable.Backoff --
4:     sendMsg()
5:   else
6:     cancelMsg()

```

---

No Algoritmo 10, presente apenas nos veículos, na linha 2, é verificado se o tempo decorrido é maior que soma do último tempo de recebimento e o tempo de reenvio calculado e também se a mensagem possui tempo de vida válido. Em caso verdadeiro, na linha 3 o *backoff* da mensagem é decrementado e na linha 4 a mensagem é reenviada na rede. Caso seja falso, na linha 6 é cancelado o reenvio da mensagem na rede.



#### 4.1.6 Classificação do RODOGE

De acordo com as comparações realizadas na Tabela 3.1, o RODOGE pode obter a seguinte classificação de acordo com a Estratégia de Reenvio:

- **Atraso:** utiliza tempos de reenvio com atrasos, é calculado tempos de reenvios com base no *backoff* da mensagem e na distância do remetente;
- **Cluster:** seleciona nós para realizarem o reenvio, seleciona primeiramente dispositivos de infraestrutura e na ausência deles seleciona veículos que menos receberam a mensagem e mais distantes do remetente;
- **Contador:** calcula o tempo de *backoff* com base na contagem de mensagens reenvias e mensagens duplicadas recebidas;
- **Distância:** calcula a distância entre nó remetente e nó receptor para utilização no atraso de reenvio e também calcula a distância entre dois alertas de acidente para verificar se é um mesmo acidente ou acidentes diferentes;
- **Localização:** utiliza a posição geográfica dos veículos para determinar se dois alertas são de um mesmo acidente ou não.

Com base na Tabela 3.1, a classificação do RODOGE de acordo com seus objetivos:

- **Broadcast Storm:** utiliza das regras de reenvio e das funções implementadas com base nas regras para diminuir o número de mensagens reenviadas na rede e assim não inundar a rede;
- **Partição da Rede:** com a utilização do tempo de reenvio com base na distância segura, os veículos na área de interesse recebem a mensagem com tempo e distância suficiente para tomada de decisão do motorista;
- **N Acidentes:** trata diferentes quantidades de acidentes e com diferentes quantidades de veículos, para casos em que haja mais de um veículo em um mesmo acidente o protocolo trata reenviando apenas um dos alertas criados.

No próximo Capítulo será descrita a metodologia utilizada para avaliação do protocolo, topologias possíveis no decorrer das simulações e as métricas para sua comparação com outras propostas.

## 5 METODOLOGIA

Para verificar o funcionamento do protocolo e realizar testes com o mesmo foram utilizados simuladores de rede, de tráfego e um *framework* que une os dois simuladores e suas funcionalidades para que se tenha as funções e características de uma rede veicular. O simulador de rede utilizado foi o Omnet++, para a mobilidade dos veículos foi utilizado o SUMO e o *framework* para junção dos dois foi o Veins.

Os cenários criados para simulação foram dois, o trace de Cologne e o trace em Grid, que possuem variações na quantidade de veículos próximo ao acidente.

Para verificar se o RODOGE reduz a quantidade de mensagens na rede são avaliadas e utilizadas métricas durante as simulações. O protocolo proposto foi comparado com protocolos do estado da arte, encontrados na literatura corrente.

### 5.1 Cenários de Simulação

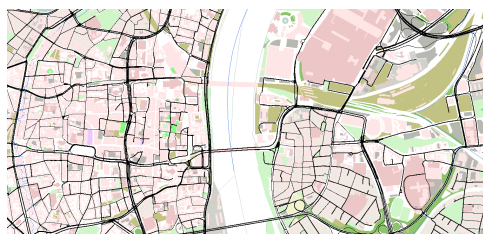
Para realizar os experimentos do projeto RODOGE utilizando o protocolo proposto são utilizados dois cenários diferentes, com cada um deles variando a quantidade de veículos acidentados em um e dois. Estes cenários são utilizados para observar diferentes comportamentos do protocolo proposto e para coleta de resultados para comparação com outros protocolos encontrados na literatura.

Destes cenários um é de ambiente fictício, implementado exclusivamente para os testes, e o outro cenário é de um ambiente real da cidade de Cologne, Alemanha.

#### 5.1.1 Cenário de Cologne

O cenário da cidade de *Cologne*, Alemanha, é um projeto chamado *TAPASCologne*, é um dos cenários mais realistas devido à grande quantidade de informações coletadas. Todo o processo de criação deste cenário esteve atrelado a ferramentas de simulação de mobilidade urbana já conhecidas, *OpenStreetMap* e *SUMO*. A Figura 5.1 ilustra o *trace* criado para cidade de Cologne.

Figura 5.1 – Trecho de Cologne no SUMO



Fonte do Autor.

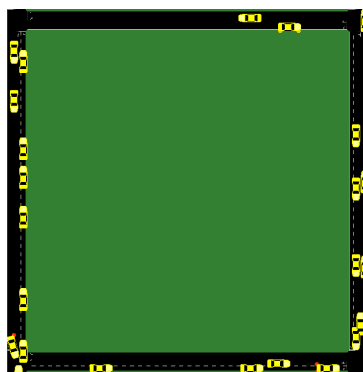
Este cenário utiliza uma sequência de veículos com movimentação, fluxo e rota variável baseada em dados obtidos nos táxis da cidade alemã. O acidente, momento em que ocorre e seu local, é implementado diretamente no *Veins*.

Para as simulações do projeto foram implementados os acidentes em três locais distintos, na qual correspondem a locais com quantidade baixa, média e alta de veículos, e 600 segundos após o início da simulação. São utilizados quatro dispositivos de infraestrutura que são alocados nas proximidades do acidente para interação com os veículos da região.

### 5.1.2 Grid de Manhattan

Este cenário foi criado utilizando o simulador de mobilidade SUMO. Este cenário é comumente utilizado em simulações de redes veiculares para avaliar protocolos em ambientes urbanos. É conhecido como mapa Manhattan Grid (4x4), que pode ser observado na Figura 5.2.

Figura 5.2 – Trecho do Grid no SUMO



Fonte do Autor.

A Figura 5.2 representa um trecho do mapa utilizado para simulação, cada segmento de via possui extensão de 250 metros, duas vias para cada direção. Este mapa foi gerado através do *netgenerate*, uma das ferramentas presentes no *SUMO*. Foram geradas rotas aleatórias com origem e destino a distância mínima de 1 km.

Para as simulações do projeto foram implementados duas situações de acidente, com um e dois veículos, escalonados para 65 segundos após o início da simulação. Com variação a quantidade de veículos em 25, 50, 100, 150 e 200, para que fossem coletados resultados com quantidade de veículos baixa, média e alta. Foram utilizados quatro dispositivos de infraestrutura alocados de forma central no mapa, porém sem estarem na área de cobertura um do outro.

## 5.2 Topologias de Simulação

No decorrer das simulações a topologia da rede pode alterar frequentemente, os veículos podem possuir diferentes posições em relação ao dispositivo de infraestrutura, ao veículo acidentado e aos demais veículos vizinhos. Essa mudança constante e rápida da topologia da rede é uma das características da rede veicular.

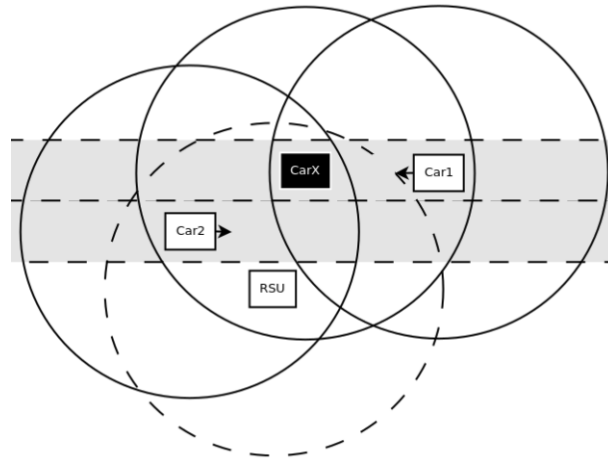
Inúmeras topologias podem ser geradas nas simulações, dependendo do veículo referência, e que acarretam em diferentes acontecimentos no decorrer do desenvolvimento da rede veicular. Há diferentes variantes, como quando o veículo acidentado está ou não na área de cobertura do dispositivo de infraestrutura, quais e quantos veículos estão na área de cobertura do veículo acidentado e do dispositivo de infraestrutura, quais veículos estão na área de cobertura do dispositivo de infraestrutura, e outras inúmeras possíveis.

### 5.2.1 Topologia Densa - Com 1 Veículo Acidentado

Nesta topologia, que pode ser observada na Figura 5.3, todos os nós da rede estão na área de cobertura do veículo acidentado (CarX), na área de cobertura do veículo 1 (Car1) está o CarX, na área de cobertura do veículo 2 (Car2) estão o CarX e o dispositivo de infraestrutura (RSU) e na área de cobertura da RSU estão o CarX e o Car2.

No instante em que o veículo sofre um acidente ele cria uma mensagem alerta e preenche os campos da mensagem com dados obtidos por ele mesmo. Após criar a mensagem, ele a envia em *broadcast* para os vizinhos que estão em sua área de cobertura, Car1, Car2 e RSU. Cada nó ao receber a mensagem realiza uma comparação com a tabela de mensagens de cada e verifica se é uma mensagem nova, de mesmo acidente, ou existem, e assim adiciona, ignora ou atualiza a mensagem, respectivamente.

Figura 5.3 – Topologia Densa - Com 1 Veículo Acidentado



Fonte do Autor.

Os veículos ao prepararem a mensagem recebida para reenvio realizam verificações relacionadas às regras e as funções do protocolo destinadas a eles, como o tempo de vida da mensagem, o tipo de remetente da mensagem, o valor do *backoff* da mensagem e a distância do remetente. Como é uma mensagem recém criada na rede ela possui tempo de vida válido.

A RSU ao preparar a mensagem recebida realiza verificações relacionadas às regras e funções do protocolo destinadas a ele.

A RSU reenvia a mensagem antes dos demais veículos, pois ele não leva em conta a densidade da mensagem na rede e assim não há incremento no tempo de envio em relação ao valor de *backoff*. Este reenvio da mensagem pelo RSU, com intervalo entre as mensagens calculado anteriormente, ocorre até o momento em que a mensagem possuir tempo de vida válido, quando este tempo da mensagem expirar ela é deletada da tabela de mensagens.

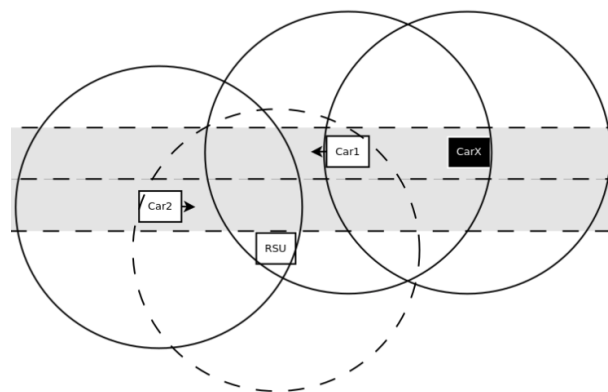
Os veículos ao receberem a mensagem enviada pelo RSU realizam a comparação com a tabela de mensagem e atualizam os campos de remetente, tempo de recebimento e valor de *backoff*. Assim, cancelam a mensagem escalonada para reenvio e iniciam a preparação de reenvio da mensagem novamente. Há uma nova verificação nas regras e funções do protocolo e enquanto for constatado que o tipo de remetente da mensagem é um RSU não ocorre reenvio.

Para verificar se um veículo distanciou do RSU é verificado com relação ao tempo padrão de envio da mensagem. Caso um veículo continue recebendo a mensagem dentro do tempo calculado, este está na área de cobertura e caso não tenha recebido a mensagem é porque está fora da área de cobertura. Ao sair da área de cobertura o veículo retorna à preparação de reenvio da mensagem e a reenvia caso sejam atendidas as regras e funções do protocolo destinadas a eles.

### 5.2.2 Topologia Mista - Com 1 Veículo Acidentado

Nesta topologia, que pode ser observada na Figura 5.4, apenas o veículo 1 (Car1) está na área de cobertura do veículo acidentado (CarX), na área de cobertura do Car1 estão o veículo acidentado e o dispositivo de infraestrutura (RSU), na área de cobertura do veículo 2 (Car2) está a RSU e na área de cobertura da RSU estão Car1 e Car2.

Figura 5.4 – Topologia Mista - Com 1 Veículo Acidentado



Fonte do Autor.

No instante em que o veículo sofre um acidente ele cria a mensagem, a preenche e a envia em *broadcast* para os vizinhos que estão em sua área de cobertura, Car1. Ao receber a mensagem o Car1 realiza uma comparação com a sua tabela de mensagens, verifica se é uma mensagem nova, de mesmo acidente, ou existente na tabela, e assim adiciona ou atualiza a mensagem.

O Car1 ao preparar a mensagem para reenvio verifica se recebe novamente a mensagem, caso a receba antes ele cancela o reenvio, compara novamente a mensagem recebida e prepara a mensagem para reenvio, caso ele não receba a mensagem ele a reenvia em *broadcast* para seus vizinhos. Este veículo reenvia a mensagem enquanto ela tiver tempo válido e atender as regras e funções do protocolo.

A RSU ao receber a mensagem do Car1, a compara com sua tabela de mensagens e a prepara para o reenvio em *broadcast* para os seus nós vizinhos, Car1 e Car2. A mensagem é reenviada enquanto ela tiver tempo válido.

O veículo acidentado ao receber a mensagem do Car1 compara com a tabela de mensagens, atualiza os campos necessários e em caso de ter uma mensagem escalonada a cancela e prepara uma nova mensagem para reenvio. Este veículo reenvia a mensagem enquanto ela tiver tempo válido e atender as regras e funções do protocolo.

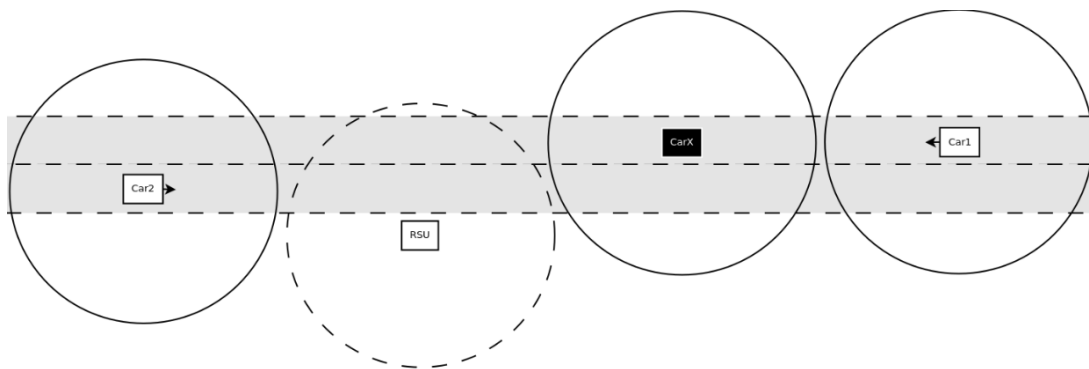
O Car1 ao receber a mensagem da RSU a compara com a tabela de mensagens, atualiza os campos necessários, em caso de ter uma mensagem escalonada a cancela. Ao realizar a preparação da mensagem para reenvio é interrompido, pois é constatado que a mensagem possui como remetente um dispositivo. Essa interrupção ocorre até o momento em que o veículo está na área de cobertura. Ao constatar a perda de contato com o dispositivo, o veículo retorna a preparação da mensagem e caso sejam atendidas as regras e funções o reenvio é retomado.

O Car2 ao receber a mensagem da RSU compara a mensagem com a tabela de mensagens e adiciona a mensagem da RSU. Ao iniciar sua preparação para reenvio da mensagem é constatado que o remetente é um dispositivo e assim é interrompido.

### 5.2.3 Topologia Esparsa - Com 1 Veículo Acidentado

Nesta topologia, que pode ser observada na Figura 5.5, que nenhum dos nós está na área de cobertura de algum outro nó, o veículo acidentado (CarX), veículo 1 (Car1), veículo 2 (Car2) e dispositivo de infraestrutura (RSU), estão todos fora de contato entre si. Os nós nessa topologia recebem a mensagem apenas instantes após ocorrência do acidente e não de imediato como nas demais topologias.

Figura 5.5 – Topologia Esparsa - Com 1 Veículo Acidentado



Fonte do Autor.

No instante em que o veículo sofre um acidente ele cria a mensagem, a preenche e a envia em *broadcast* e o Car1 recebe a mensagem ao entrar na área de cobertura do veículo acidentado. O tempo gasto para receber a mensagem pode variar de acordo com a velocidade do veículo. O veículo acidentado reenvia a mensagem enquanto esta possuir tempo de vida válido.

O Car1 ao receber a mensagem do veículo acidentado compara a mensagem com a tabela de mensagens, adiciona a mensagem e inicia a preparação para reenvio. Até passar por toda a

área de cobertura do veículo acidentado, ele recebe várias mensagens e as compara com a tabela de mensagens. Neste tempo, ele escalona tempos para reenvios da mensagens e as cancela ou não. Ao sair da área de cobertura do veículo acidentado, o Car1 continua a preparação de reenvio da mensagem, caso atenda as regras e funções, ele a reenvia para a RSU.

A RSU ao receber a mensagem do Car1 inicia a comparação com sua tabela de mensagens e a adiciona. Ele inicia sua preparação para reenvio, caso atenda as regras e as funções do protocolo. O Car1 ao receber a mensagem do dispositivo interrompe os seus reenvios e cancela mensagem escalonada. A RSU reenvia a mensagem enquanto esta possuir tempo de vida.

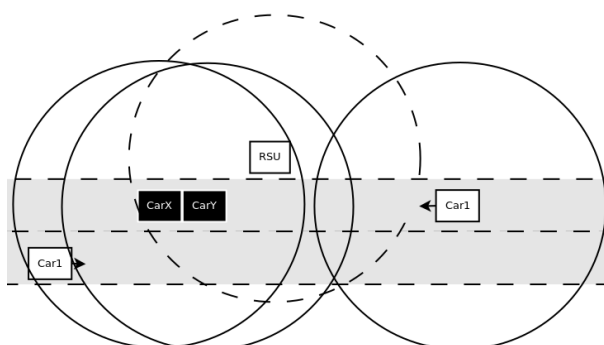
O Car2 ao receber a mensagem da RSU não reenvia a mensagem enquanto estiver em sua área de cobertura. Ao sair da área de cobertura da RSU o Car2, caso a mensagem possua tempo de válido, passa a reenviar a mensagem.

O Car1 continua seu trajeto e ao sair da área de cobertura da RSU e a mensagem possua tempo de vida válido, reinicia o reenvio da mensagem até o tempo da expirar.

#### 5.2.4 Topologia Densa - Com 2 Veículo Acidentado

Nesta topologia, que pode ser observada na Figura 5.6, há dois veículos acidentados em um mesmo acidente, sendo gerada uma mensagem por cada do mesmo evento. Na área de cobertura do primeiro veículo acidentado (CarX), estão o segundo veículo acidentado (CarY), o veículo 2 (Car2) e o dispositivo de infraestrutura (RSU). Na área de cobertura do segundo carro acidentado (CarY), estão o primeiro carro acidentado (CarX) e a RSU. Na área de cobertura do veículo 1 (Car1) não está nenhum outro nó. Na área de cobertura do Car2, está apenas o primeiro veículo acidentado (CarX).

Figura 5.6 – Topologia Densa - Com 2 Veículo Acidentado



Fonte do Autor.



No instante em que ocorre o acidente cada um dos veículos acidentados cria uma mensagem de alerta distinta, porém relacionadas ao mesmo acidente. CarX e CarY iniciam o envio de suas mensagens aos seus vizinhos.

A RSU ao receber as mensagens do primeiro e segundo veículos acidentados compara com a sua tabela de mensagens. A primeira mensagem recebida é armazenada na tabela, a segunda mensagem recebida é comparada com a tabela de mensagens e é constatado que é uma mensagem de mesmo acidente e então é descartada. A RSU inicia o reenvio da mensagem armazenada até a mensagem possuir tempo válido.

Os veículos acidentados ao receberam a mensagem um do outro a comparam com suas tabelas e as ignoram. Ao receberem a mensagem da RSU realizam a comparação com suas respectivas tabelas, um verifica que é sua própria mensagem e outro que a mensagem é do mesmo acidente. Não realizam o reenvio da mensagem, pois estão na área de cobertura da RSU.

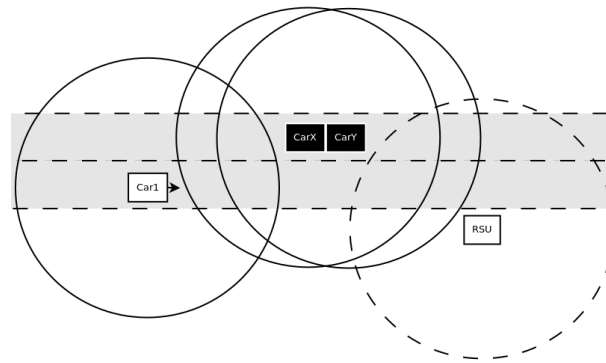
O Car1 ao entrar na área de cobertura da RSU recebe apenas uma das mensagens, pois uma delas foi descartada. O Car1 não realiza o reenvio enquanto estiver na área de cobertura da RSU e ao sair dessa área caso a mensagem possua tempo ainda válido ele inicia o reenvio.

O Car2 recebe a mensagem do primeiro veículo acidentado, compara com a sua tabela de mensagens e a adiciona. Inicia o reenvio da mensagem e ao entrar na área de cobertura da RSU e receber a mensagem, ele a compara com sua tabela. A mensagem recebida pode ser a mesma do primeiro veículo acidentado ou a do segundo veículo acidentado. No primeiro caso, a mensagem não é adicionada por ser repetida e no segundo caso, a mensagem não é adicionada por ser de um mesmo acidente. O Car2 não realiza o reenvio enquanto estiver na área de cobertura da RSU e ao sair dessa área caso a mensagem possua tempo ainda válido ele inicia o reenvio.

### **5.2.5 Topologia Mista - Com 2 Veículo Acidentado**

Nesta topologia, que pode ser observada na Figura 5.7, há dois veículos envolvidos em um mesmo acidente, sendo gerada uma mensagem por cada do mesmo evento. Na área de cobertura do primeiro carro acidentado (CarX) está apenas o segundo veículo acidentado (CarY) e na área de cobertura do segundo veículo acidentado está apenas o primeiro veículo acidentado. O veículo 1 (Car1) e o dispositivo de infraestrutura (RSU) não possuem nós em suas áreas de cobertura.

Figura 5.7 – Topologia Mista - Com 2 Veículo Acidentado



Fonte do Autor.

No momento que ocorre o acidente ambos os veículos criam uma mensagem de alerta distinta para o evento, porém relacionadas ao mesmo acidente. Os veículos, CarX e CarY, iniciam o envio de suas mensagens.

Os veículos acidentados ao receberem a mensagem um do outro a comparam com a tabela de mensagens e constatam que se trata de uma mensagem de mesmo acidente. Eles não adicionam a mensagem recebida e não atualizam suas tabelas. Preparam suas mensagens para reenvio enquanto elas atenderem a todas as regras e funções do protocolo.

O Car1 recebe a mensagem do CarX ao entrar em sua área de cobertura, o que ocorre instantes depois. Ao receber a mensagem, ela é adicionada na sua tabela e a preparada para reenvio caso atenda as regras e funções do protocolo. Alguns instantes depois, o Car1 recebe a mensagem do CarY e a compara com sua tabela, é constatado que se trata de um mesmo acidente e a ignora. Continuando sua rota e preparação para reenvios, o Car1 reenvia a mensagem a RSU ao entrar em sua área, caso a mensagem atenda as regras e funções do protocolo. Ao receber a mensagem de volta, o Car1 interrompe seus reenvios.

A RSU ao receber a mensagem realiza o reenvio até que as funções e regras do protocolo não sejam mais atendidas.

A não presença de Topologia com 2 veículos em acidentes diferentes se dá ao fato de ser basicamente o mesmo funcionamento da Topologia com 1 veículo acidentado, o que difere é em uma maior quantidade de mensagens, porém ainda havendo controles de reenvio.

### 5.3 Métricas de Validação

Para analisar o protocolo proposto foram utilizadas as métricas relacionadas a quantidade de mensagens geradas e recebidas, a distância, tempo e saltos para recebimento da mensagem e a porcentagem de veículos geradores e cobertos pela mensagem.

#### 5.3.1 Quantidade de Mensagens Geradas

A quantidade de mensagens geradas está relacionada a quantidade de mensagens reenviadas na rede. Analisar essa métrica é fundamental para verificar se as regras e funções do protocolo proposto diminuem a quantidade de mensagens na rede. São analisadas as quantidades de três formas diferente: total de mensagens geradas na simulação, média de mensagens geradas por nó e média de mensagens geradas por total de nós.

O total de mensagens geradas é baseado na simulação como um todo, é realizado um somatório de todas as mensagens que foram reenviadas por todos os nós durante uma dada repetição. Após este somatório é calculada a média e intervalo de confiança dos valores obtidos nas repetições da simulação.

A média e intervalo de confiança de mensagens geradas por nó é calculada com base no total de mensagens geradas na simulação e no número de nós que geraram essas mensagens na mesma simulação.

A média e intervalo de confiança de mensagens geradas por total de nós é calculada com base no total de mensagens geradas na simulação e no total de nós presentes na simulação.

#### 5.3.2 Quantidade de Mensagens Recebidas

A quantidade mensagens recebidas está relacionada a quantidade de mensagens geradas e a quantidade de mensagens repetidas recebidas pelos nós na rede. Analisar essa métrica é fundamental para verificar se com as regras e funções que restringem o reenvio também diminuem a quantidade de mensagens duplicadas recebidas. São analisadas as quantidades de três formas diferente: total de mensagens recebidas na simulação, média de mensagens recebidas por nó e média de mensagens recebidas por total de nós.

O total de mensagens recebidas é baseado nos resultados de cada repetição, na qual é realizado um somatório com todas as mensagens recebidas pelo nós na rede. Após este somatório é calculada a média e intervalo de confiança dos valores obtidos nas repetições da simulação.

A média e intervalo de confiança de mensagens recebidas por nó é calculada com base no total de mensagens recebidas na simulação e no número de nós que receberam a mensagem durante uma repetição.

A média e intervalo de confiança de mensagens recebidas por total de nós é calculada com base no total de mensagens recebidas na simulação e no total de nós presentes na simulação.

### **5.3.3 Distância para Recebimento**

A distância de recebimento da mensagem é analisada para verificar se as regras e funções do protocolo para diminuir a quantidade de mensagens reenviadas na rede afeta negativamente a área de cobertura do alerta. É analisado se os veículos recebem a mensagem com distância suficiente para a reação do motorista e frenagem do veículo.

Os dados são calculados com base na distância entre o veículo acidentado e o veículo que recebeu a mensagem pela primeira vez. Assim é calculada a média de cada repetição e após calculada a média e intervalo de confiança das repetições.

### **5.3.4 Saltos para Recebimento**

A quantidade de saltos para recebimento é analisada para verificar se as regras e funções para diminuir a quantidade de mensagens reenviadas na rede afeta negativamente na propagação da mensagem na rede. É analisado se os veículos recebem a mensagem com o menor número de saltos possível.

Os resultados são coletados no veículos e o número de saltos de uma mensagem é incrementada de acordo com sua passagem por nós intermediários até ela chegar ao nó destino. Assim é calculada a média de saltos para cada repetição e após é calculada a média e intervalo de confiança das repetições.

### **5.3.5 Porcentagem de Geradores de Mensagens**

A porcentagem de geradores de mensagens é analisada para verificar se as regras e funções de remetente da mensagem e de incremento no tempo da mensagem afetam positivamente na rede. É analisado se ocorre diminuição na quantidade de nós geradores de mensagem.

Os resultados são obtidos de acordo com a quantidade de nós que reenviaram a mensagem em relação ao total de nós presentes na simulação. É calculada a porcentagem para cada repetição e após é calculada a média e intervalo de confiança das repetições.

### **5.3.6 Porcentagem de Veículos Cobertos**

A porcentagem de veículos cobertos pela mensagem é analisada para verificar se as regras e funções para diminuir a quantidade de mensagens reenviadas na rede afeta negativamente na porcentagem de veículos que recebem a mensagem. É analisado se mesmo com a diminuição na quantidade de mensagens o percentual de veículos permanece crescente na variação da quantidade de veículos.

Os resultados são obtidos de acordo com a quantidade de nós que recebem a mensagem em relação ao total de nós presente na simulação. É calculada a porcentagem para cada repetição e após é calculada a média e intervalo de confiança das repetições.

## **5.4 Protocolos Comparados**

Para avaliar o funcionamento, analisar o desempenho e comparar os resultados do protocolo RODOGE, foram realizadas simulações com protocolos clássicos de disseminação de mensagens, Flooding, AID e DBRS, descritos no Capítulo 3, e realizando análises e comparações com as métricas descritas. As simulações, resultados e comparações são apresentados no próximo Capítulo.

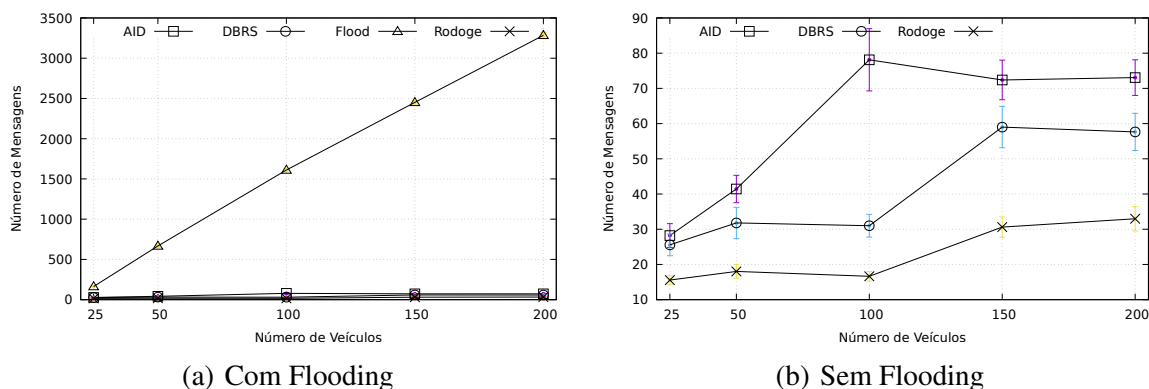
## 6 RESULTADOS E DISCUSSÃO

Neste trabalho foram simulados a disseminação de mensagens de alerta de acidentes por veículos. Para avaliar o protocolo RODOGE foram utilizados dois ambientes, o primeiro de 4 vias rodoviárias (2 em cada sentido) e o segundo o cenário de Cologne, e comparado aos protocolos clássicos de disseminação de mensagens Flooding, AID e DBRS. Em cada ambiente foram realizados experimentos com um e dois acidentes, para analisar os resultados dos protocolos ao tratar as diferentes quantidades de veículos acidentados.

### 6.1 Ambiente Grid com 1 Acidente

A Figura 6.1 apresenta os resultados da simulação para média de mensagens geradas por cada protocolo considerando a variação da quantidade de veículos (25, 50, 100, 150 e 200). Na Figura 6.1(a) são apresentados os resultados com Flooding, que foram superiores aos demais protocolos e seguem uma tendência de crescimento com o aumento de veículos. Na Figura 6.1(b) são apresentados os resultados dos outros protocolos. O RODOGE possui resultados melhores que os demais, gerando menor quantidade de mensagens em todas as variações de veículos, cerca de metade que os outros protocolos. A redução do número de mensagens geradas pelo RODOGE está relacionada às regras para reenvio de mensagens utilizadas no trabalho, como a regra para verificar o remetente, que centraliza os reenvios aos dispositivos de infraestrutura e também as regras que incrementam o tempo da mensagem de acordo com sua densidade e distância do remetente.

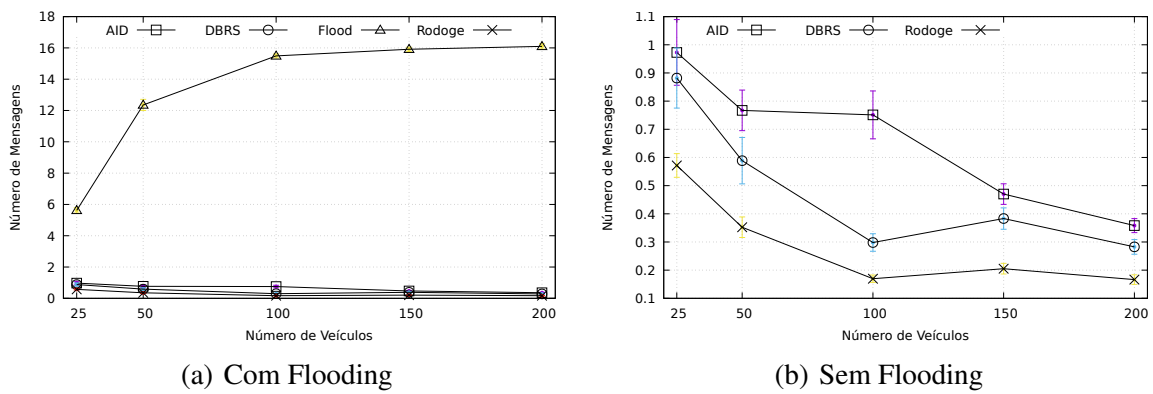
Figura 6.1 – Grid 1 Acidente - Média de mensagens geradas por execução.



Os resultados para verificar a quantidade média de mensagens geradas, calculado com base no total de nós presentes nas simulações, podem ser observados na Figura 6.2. Em todas as

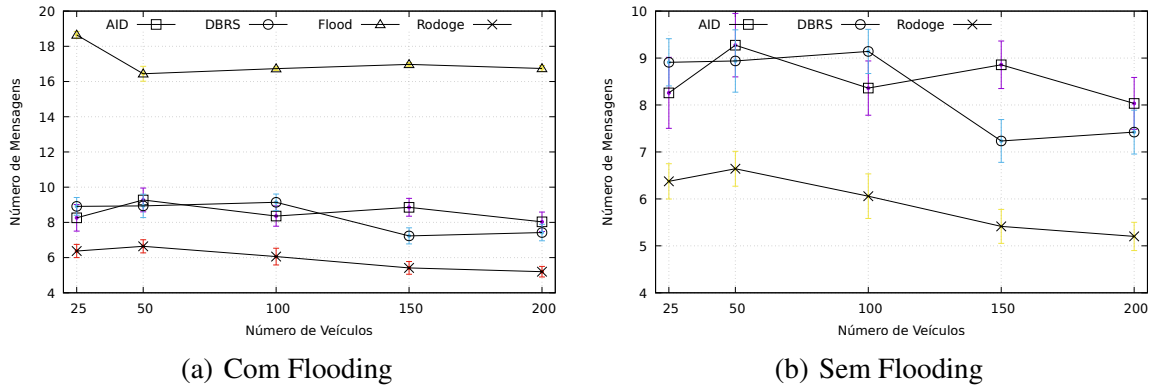
variações de veículos o protocolo Flooding obteve resultados superiores aos demais protocolos, como pode ser observado na Figura 6.2(a), pois gerou uma quantidade de mensagens por nó superior aos demais. Os resultados para os demais protocolos podem ser observados na Figura 6.2(b). O RODOGE possui os melhores resultados, gerando uma quantidade de mensagens por nó menor que os demais protocolos. A diferença entre os resultados se dá de acordo as regras de reenvio implementadas no RODOGE, que geram um número total de mensagens menor que os demais protocolos e assim um menor número de mensagens por nó.

Figura 6.2 – Grid 1 Acidente - Média de mensagens geradas pelo total de nó.



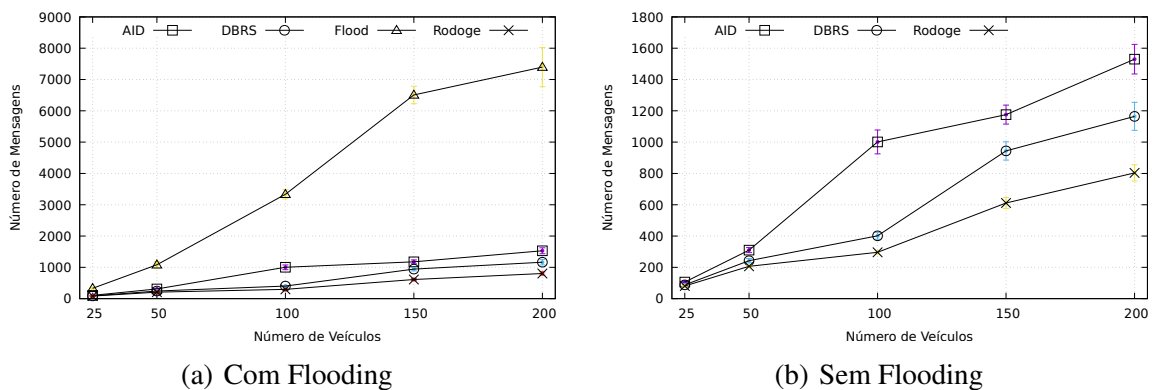
A Figura 6.3 apresenta os resultados a quantidade média de mensagens geradas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que geraram mensagens. Na Figura 6.3(a) são apresentados os resultados com Flooding, em todas as variações de veículos os resultados foram superiores aos demais protocolos. Na Figura 6.3(b) são apresentados os resultados dos demais protocolos. O RODOGE obteve as melhores médias nas variações de veículos, gerando menor número de mensagens. O menor número de mensagens geradas pelo RODOGE está relacionada ao menor número total de mensagens geradas, devido aos controles de reenvio.

Figura 6.3 – Grid 1 Acidente - Média de mensagens geradas por retransmissores.



A Figura 6.4 apresenta os resultados da simulação para média de mensagens recebidas por cada protocolo considerando a variação da quantidade de veículos (25, 50, 100, 150 e 200). Na Figura 6.4(a) são apresentados os resultados com Flooding, que foram superiores aos demais protocolos e seguem uma tendência de crescimento com o aumento de veículos. Na Figura 6.4(b) são apresentados os resultados dos outros protocolos. Com 25 veículos os protocolos possuem resultados estatisticamente similares, com 50 veículos o protocolo AID possui número de mensagens recebidas maior que DBRS e RODOGE e a partir de 100 veículos os resultados de RODOGE são melhores que os demais, pois os veículos recebem menos mensagens. Essa diferença entre os protocolos ocorre devido ao menor número de mensagens geradas pelo RODOGE, devido as regras que controlam o reenvio de mensagens, como a regra centraliza os reenvios aos dispositivos de infraestrutura e o incremento no tempo de reenvio dos veículos.

Figura 6.4 – Grid 1 Acidente - Média de mensagens recebidas por execução.

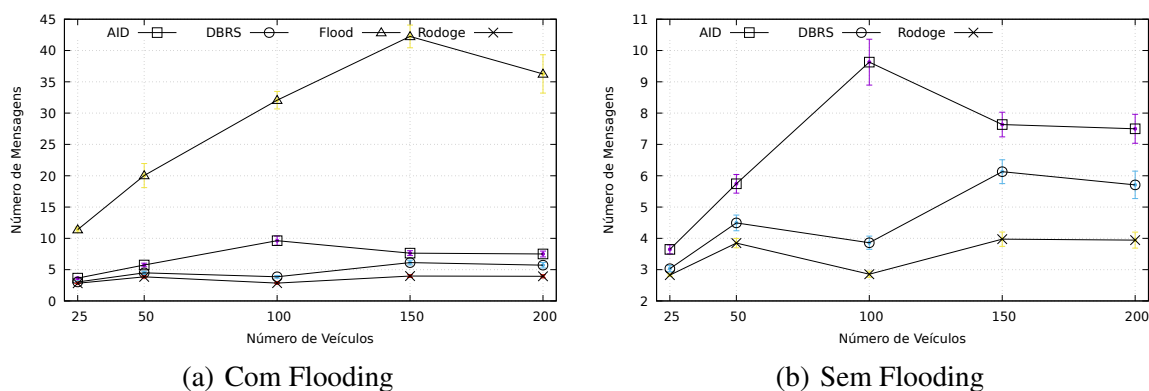


Os resultados para verificar a quantidade média de mensagens recebidas, calculado com base no total de nós presentes nas simulações, podem ser observados na Figura 6.5. Em todas as variações de veículos o protocolo Flooding obteve resultados superiores aos demais protocolos,



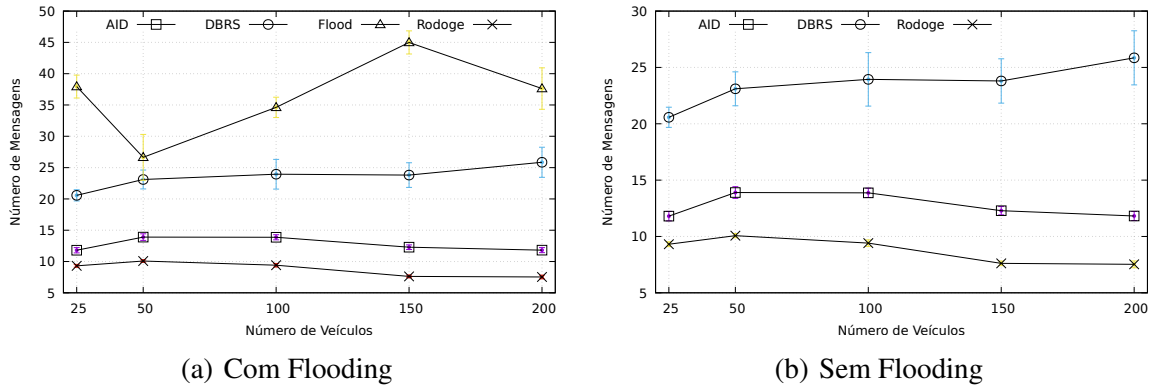
como pode ser observado na Figura 6.5(a), pois gerou uma quantidade de mensagens por nó superior aos demais. Os resultados para os demais protocolos podem ser observado na Figura 6.5(b). O protocolo AID possui número de mensagens recebidas por nó superior aos demais em todas as repetições, os protocolos DBRS e RODOGE possuem resultados estatisticamente similares para 25 veículos e a partir de 50 veículos RODOGE obteve os melhores resultados, com os veículos recebendo um menor número de mensagens duplicadas. A diminuição na quantidade de mensagens recebidas por nó está relacionada na diminuição na quantidade de mensagens geradas devido as regras implementadas no protocolo e na diminuição de mensagens geradas, que acarretam em menor número total de mensagens recebidas.

Figura 6.5 – Grid 1 Acidente - Média de mensagens recebidas por nó.



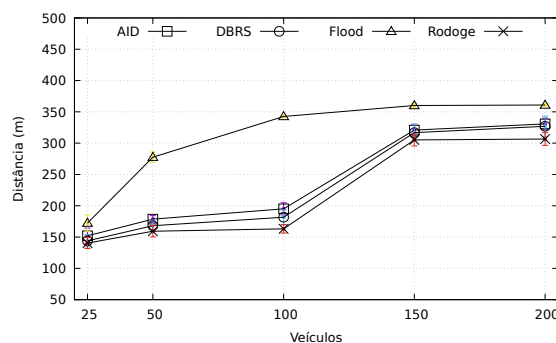
A Figura 6.6 apresenta os resultados a quantidade média de mensagens geradas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que geraram mensagens. Na Figura 6.6(a) são apresentados os resultados com Flooding, em todas as variações de veículos os resultados foram superiores aos demais protocolos. Na Figura 6.6(b) são apresentados os resultados dos demais protocolos. Em todas as variações da quantidade de veículos o protocolo RODOGE recebeu uma quantidade média de mensagens por receptor inferior aos demais protocolos. A menor média de mensagens recebidas está relacionada a menor quantidade total recebidas e também no menor número de mensagens geradas devido aos controles de reenvio, como de centralizar os reenvios aos dispositivos de infraestrutura e aos atrasos de reenvio calculados para os veículos.

Figura 6.6 – Grid 1 Acidente - Média de mensagens recebidas por receptores.



A Figura 6.7 apresenta os resultados da distância média para recebimento da mensagem de alerta. Neste cenário a velocidade máxima dos veículos é de 50km/h e assim a distância segura para reação do motorista e frenagem do veículo é de 30,6m. Em todas as variações de veículos e protocolos a distância de recebimento foi superior a distância segura. Os resultados do Flooding superaram os demais protocolos devido ao grande número de mensagens reenviadas por ele. Para 25, 50, 150 e 200 veículos, os protocolos AID, DBRS e RODOGE obtiveram resultados estatisticamente similares. Essa menor distância nos três protocolos se deve ao controle de reenvio de mensagens que eles possuem. Apesar da menor quantidade de mensagens geradas pelo RODOGE, a distância de recebimento está sempre acima da distância segura.

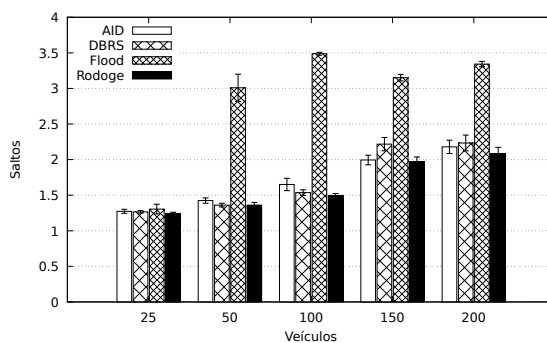
Figura 6.7 – Grid 1 Acidente - Média do Distância de Recebimento



Para verificar a entrega das mensagens aos veículos com o menor número de intermediários possíveis, foi calculado a média de saltos da mensagem de alerta, como mostrado na Figura 6.8. No cenário de 25 veículos, os protocolos obtiveram resultados estatisticamente similares. Para os demais cenários o Flooding obteve resultados superiores aos demais, que obtiveram resultados estatisticamente similares. A justificativa para controle na quantidade de saltos do

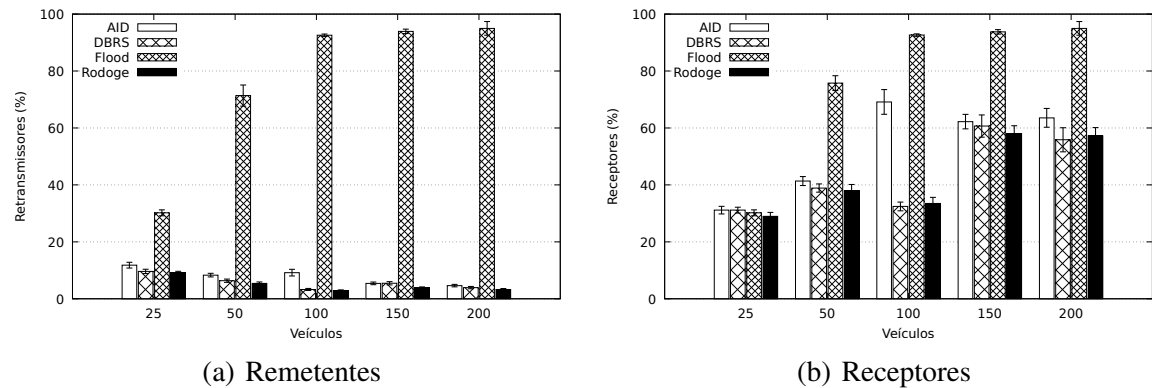
RODOGE é devido a centralização de reenvio pelo dispositivo de infraestrutura, de acordo com a regra de verificar o remetente da mensagem.

Figura 6.8 – Grid 1 Acidente - Média de Saltos de Recebimento



A Figura 6.9 mostra as médias de porcentagens de remetentes e receptores de mensagens nas simulações para as variações de veículos. Na Figura 6.9(a) é calculado o percentual de dispositivos que reenviam as mensagens de alerta. Em todas as simulações o Flooding obteve a maior porcentagem de remetentes, pois não há controles de reenvio de mensagens. Os resultados dos protocolos AID, DBRS e RODOGE foram estatisticamente similares. A justificativa para o RODOGE obter tais resultados está relacionada a utilização da Regra 3, que limita o reenvio de mensagens pelos veículos dentro da área de cobertura do dispositivo de infraestrutura. A Figura 6.9(b) mostra os resultados da porcentagem de veículos cobertos pela mensagem de alerta. Em todas as simulações o Flooding obteve as maiores porcentagens, devido ao grande número de mensagens que reenvia e assim alcança os nós mais rapidamente. O AID, DBRS e RODOGE obtiveram resultados estatisticamente similares para 25, 50, 150 e 200 veículos e menores que o Flooding, isso ocorre devido aos seus controles de reenvios. Além do RODOGE reduzir a quantidade de mensagens reenviadas, o percentual de veículos que recebem a mensagem de alerta em distância segura é semelhante ao dos protocolos que reenviam mais mensagens.

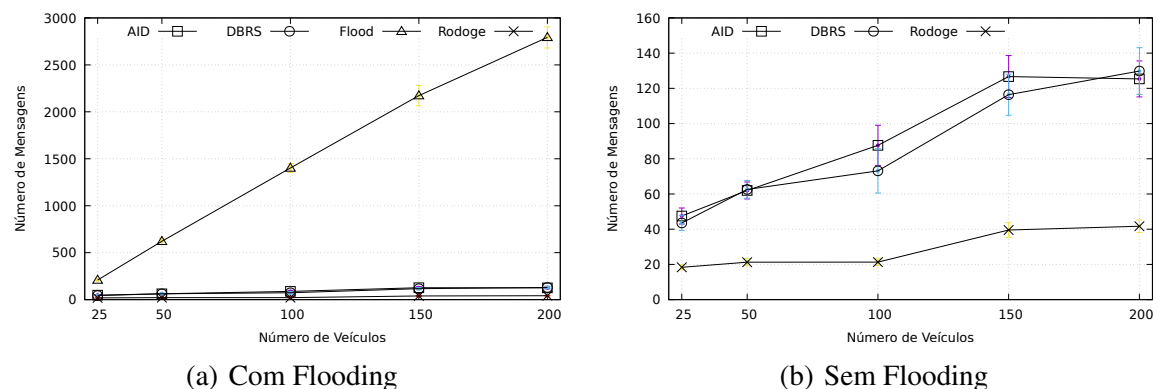
Figura 6.9 – Grid 1 Acidente - Porcentagens de Remetentes e Receptores.



## 6.2 Ambiente Grid com 2 Acidentes

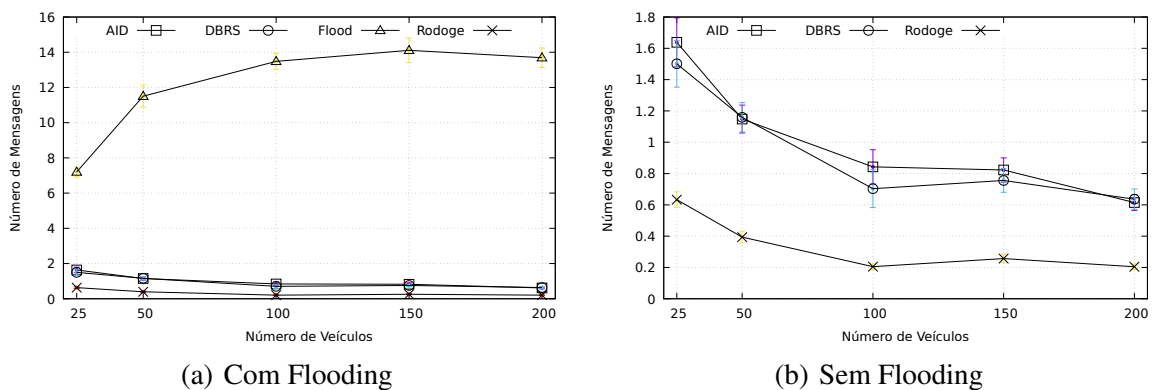
A Figura 6.10 apresenta os resultados da simulação para média de mensagens geradas por cada protocolo considerando a variação da quantidade de veículos (25, 50, 100, 150 e 200). Na Figura 6.10(a) são apresentados os resultados com Flooding, que são superiores aos demais protocolos em todas as variações de veículos e possui uma tendência de crescimento. Na Figura 6.10(b) são apresentados os resultados sem Flooding. O protocolo RODOGE obteve melhores resultados em todas as variações de quantidade de veículos, gerando cerca de um terço das mensagens geradas pelos outros protocolos, mas mantendo a mesma cobertura de disseminação de alertas. A redução do número de mensagens geradas pelo RODOGE está relacionada as regras para reenvio de mensagens utilizadas, como a regra para verificar o remetente, que centraliza os reenvios aos dispositivos de infraestrutura, a regra que compara as mensagens, devido aos dois acidentes que há neste cenário e também as regras que incrementam o tempo da mensagem de acordo com sua densidade e distância do remetente.

Figura 6.10 – Grid 2 Acidentes - Média de mensagens geradas por execução.



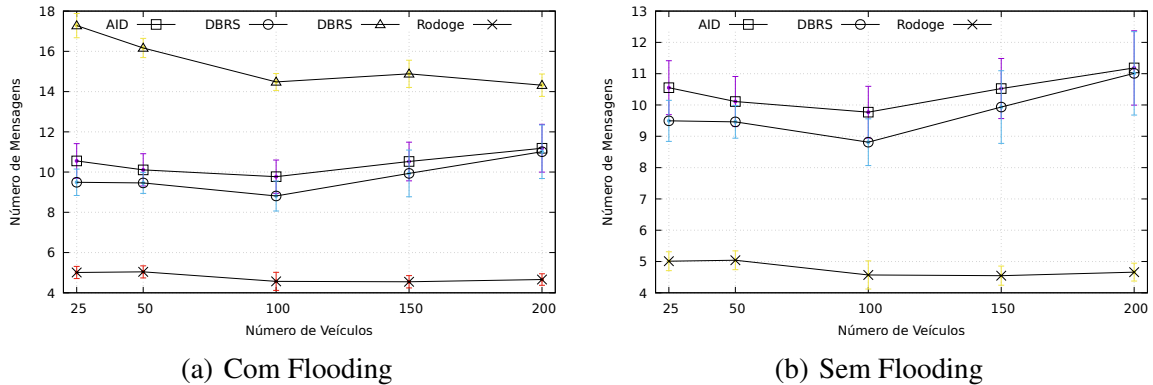
Os resultados para verificar a quantidade média de mensagens geradas, calculado com base no total de nós presentes nas simulações, podem ser observados na Figura 6.11. Em todas as variações de veículos o protocolo Flooding obteve resultados superiores aos demais protocolos, como pode ser observado na Figura 6.11(a), pois gerou uma quantidade de mensagens por nó superior aos demais. Os resultados para os demais protocolos podem ser observado na Figura 6.11(b). O RODOGE possui os melhores resultados, pois gerou um menor número de mensagens por nó em comparação aos outros protocolos. Esta diferença nos resultados ocorre devido aos controles de reenvios implementados com base nas regras do RODOGE, que geram um número total de mensagens menor que os demais protocolos e assim um menor número de mensagens por nó.

Figura 6.11 – Grid 2 Acidentes - Média de mensagens geradas pelo total de nós.



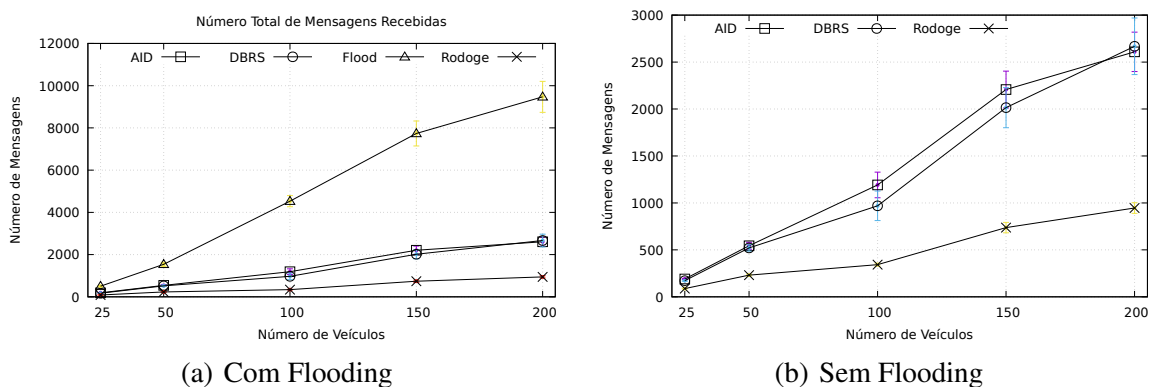
A Figura 6.12 apresenta os resultados a quantidade média de mensagens geradas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que geraram mensagens. Na Figura 6.12(a) são apresentados os resultados com Flooding, em todas as variações de veículos os resultados foram superiores aos demais protocolos. Na Figura 6.12(b) são apresentados os resultados dos demais protocolos. O protocolo RODOGE gerou uma quantidade menor de mensagens por retransmissores e assim os melhores resultados. O menor número de mensagens geradas pelo RODOGE está relacionada ao menor número total de mensagens geradas, devido aos controles de reenvio.

Figura 6.12 – Grid 2 Acidentes - Média de mensagens geradas por retransmissores.



A Figura 6.13 apresenta os resultados da simulação para média de mensagens recebidas por cada protocolo considerando a variação da quantidade de veículos (25, 50, 100, 150 e 200). Na Figura 6.13(a) são apresentados os resultados com Flooding, que foram superiores aos demais protocolos e seguem uma tendência de crescimento com o aumento de veículos. Na Figura 6.13(b) são apresentados os resultados dos outros protocolos. O protocolo RODOGE obteve os melhores resultados, com uma quantidade de mensagens inferior a 1000 em todos os cenários. Os protocolos AID e DBRS aumentaram significativamente a quantidade de mensagens recebidas, isso se justifica pela maior quantidade de mensagens geradas como pode mostrado na Figura 6.10. O menor número de mensagens geradas pelo RODOGE está relacionada ao menor número total de mensagens geradas, devido aos controles de reenvio.

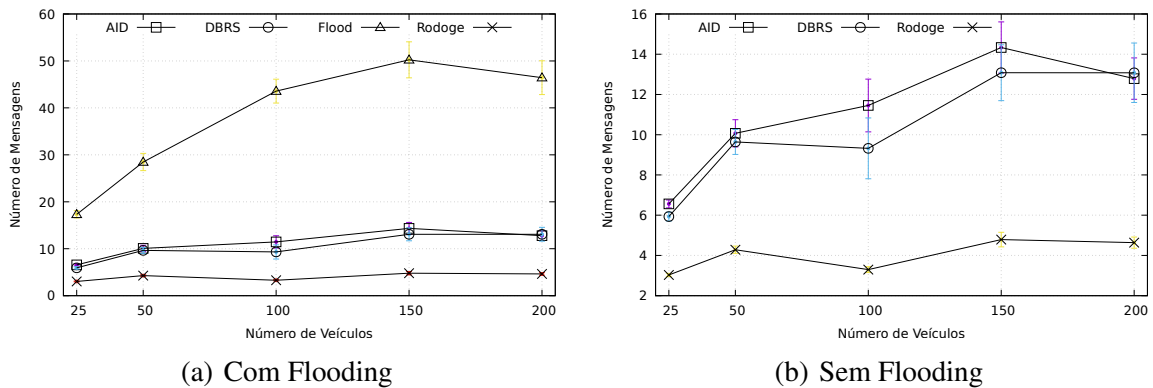
Figura 6.13 – Grid 2 Acidentes - Média de mensagens recebidas por execução.



Os resultados para verificar a quantidade média de mensagens recebidas, calculado com base no total de nós presentes nas simulações, podem ser observados na Figura 6.14. Em todas as variações de veículos o protocolo Flooding obteve resultados superiores aos demais protocolos, como pode ser observado na Figura 6.14(a), pois gerou uma quantidade de mensagens

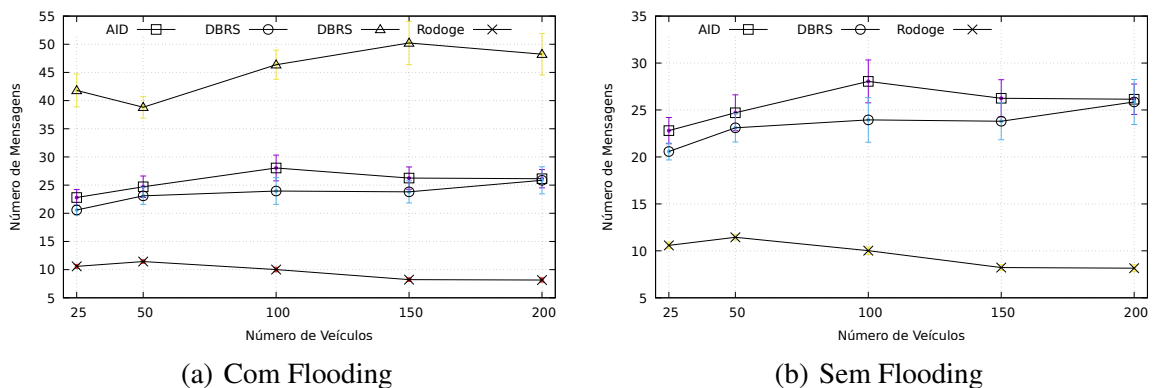
por nó superior aos demais. Os resultados para os demais protocolos podem ser observado na Figura 6.14(b). O protocolo RODOGE obteve os melhores resultados, com os veículos recebendo um menor número de mensagens duplicadas. A diminuição na quantidade de mensagens recebidas por nó está relacionada na diminuição na quantidade de mensagens geradas devido as regras implementadas no protocolo.

Figura 6.14 – Grid 2 Acidentes - Média de mensagens recebidas por nós.



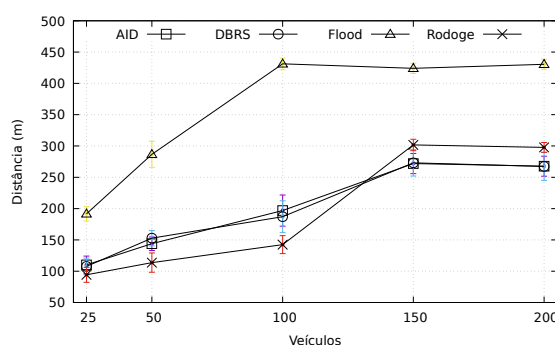
A Figura 6.15 apresenta os resultados a quantidade média de mensagens geradas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que geraram mensagens. Na Figura 6.15(a) são apresentados os resultados com Flooding, em todas as variações de veículos os resultados foram superiores aos demais protocolos. Na Figura 6.15(b) são apresentados os resultados dos demais protocolos. Em todas as variações da quantidade de veículos o protocolo RODOGE recebeu uma quantidade média de mensagens por receptor inferior aos demais protocolos. A menor média de mensagens recebidas está relacionada a menor quantidade de mensagens geradas, que também influencia na diferença entre os valores médios de mensagens recebidas por veículos entre os protocolos analisados.

Figura 6.15 – Grid 2 Acidentes - Média de mensagens recebidas por receptores.



A Figura 6.16 mostra a distância média de recebimento de mensagens pelos veículos. Neste cenário foi considerada que a velocidade máxima dos veículos é de 50 km/h e que a soma das distâncias de reação do motorista e frenagem do veículo, ou distância segura, é de 30,6 m. Os resultados encontrados por todos os protocolos, considerando a variação da quantidade de veículos, foi superior à distância segura. O protocolo Flooding superou todos os protocolos, apresentando distâncias de recebimento bem acima da distância segura. Isso se deve a grande quantidade de mensagens reenviadas por ele, que se espalham por uma área maior que os demais protocolos. Para 25 veículos, os protocolos AID, DBRS e RODOGE obtiveram resultados estatisticamente similares. Para 50 e 100, veículos AID e DBRS obtiveram resultados estatisticamente similares e superiores ao RODOGE. Para 150 e 200 veículos, o protocolo RODOGE obteve um aumento na distância de recebimento e estatisticamente similar ao DBRS, que obteve resultados estatisticamente similares ao AID. Essa menor distância nos três protocolos se deve ao controle de reenvio de mensagens que eles possuem. Apesar da menor quantidade de mensagens geradas pelo RODOGE, a distância de recebimento está sempre acima da distância segura.

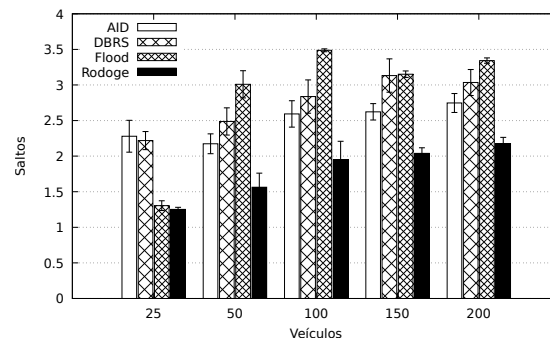
Figura 6.16 – Grid 2 Acidentes - Média do Distância de Recebimento



Para verificar a entrega das mensagens aos veículos com o menor número de intermediários possíveis, foi calculado a média de saltos da mensagem de alerta, como mostrado na Figura 6.17. No cenário de 25 veículos, os protocolos Flooding e RODOGE obtiveram resultados estatisticamente similares e melhores em relação aos demais protocolos. Para os demais cenários, de 50 a 200 veículos, o protocolo RODOGE obteve os melhores resultados, com a quantidade de saltos de mensagens inferior aos demais protocolos. A justificativa para a menor quantidade de saltos do RODOGE é devido a centralização de reenvio pelo dispositivo de infraestrutura, de acordo com a regra de verificar o remetente da mensagem no dispositivo de infraestrutura.

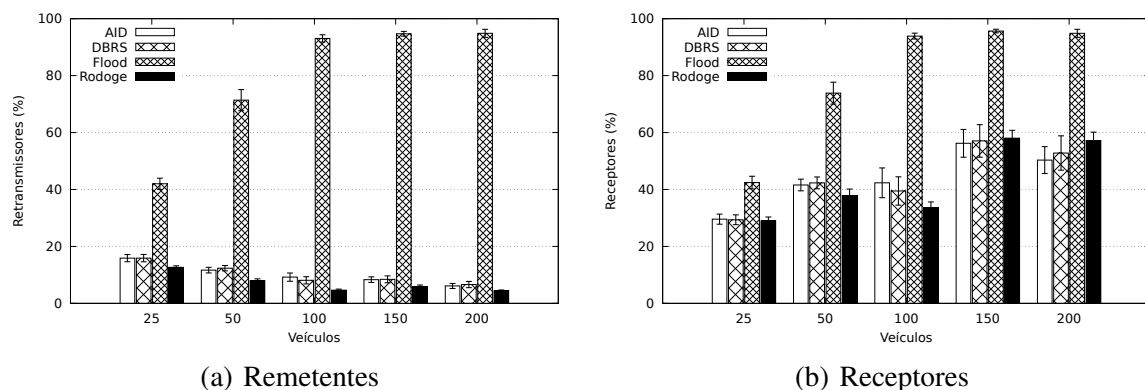


Figura 6.17 – Grid 2 Acidentes - Média de Saltos de Recebimento



A Figura 6.18 mostra as médias de porcentagens de remetentes e receptores de mensagens nas simulações para as variações de veículos. Na Figura 6.18(a) é calculado o percentual de dispositivos que reenviam as mensagens de alerta. Em todas as simulações o Flooding obteve a maior porcentagem de remetentes, pois não há controles de reenvio de mensagens. Os resultados dos protocolos AID e DBRS foram estatisticamente similares. O protocolo RODOGE obteve os melhores resultados, com menores porcentagens de dispositivos que reenviam mensagens. Isso é justificado pela aplicação da Regra 3, que limita o reenvio de mensagens pelos veículos dentro da área de cobertura do dispositivo de infraestrutura. A Figura 6.18(b) mostra os resultados da porcentagem de veículos cobertos pela mensagem de alerta. Em todas as simulações o Flooding obteve as maiores porcentagens, devido ao grande número de mensagens que reenvia e assim alcança nós mais rapidamente. O AID, DBRS e RODOGE obtiveram resultados estatisticamente similares e menores que o Flooding, isso ocorre devido aos seus controles de reenvios. Além do RODOGE reduzir a quantidade de mensagens reenviadas, o percentual de veículos que recebem a mensagem de alerta em distância segura é semelhante aos protocolos que reenviam mais mensagens.

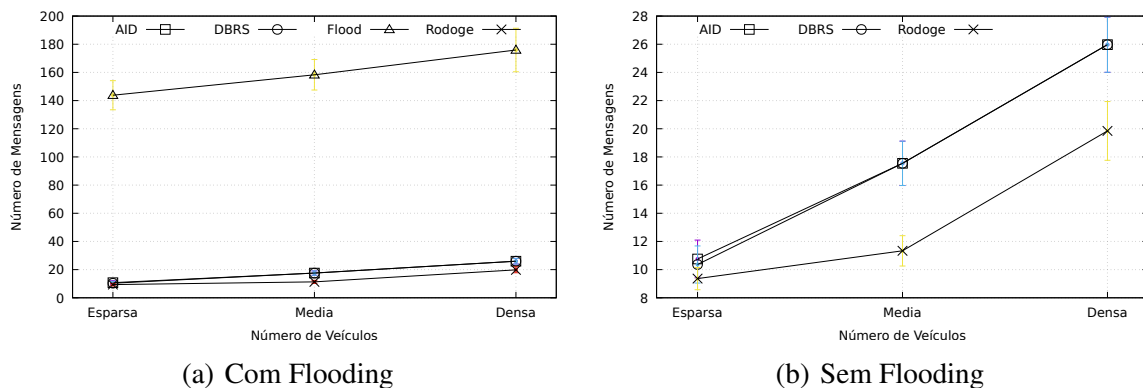
Figura 6.18 – Grid 2 Acidentes - Porcentagens de Remetentes e Receptores.



### 6.3 Ambiente Cologne com 1 Acidente

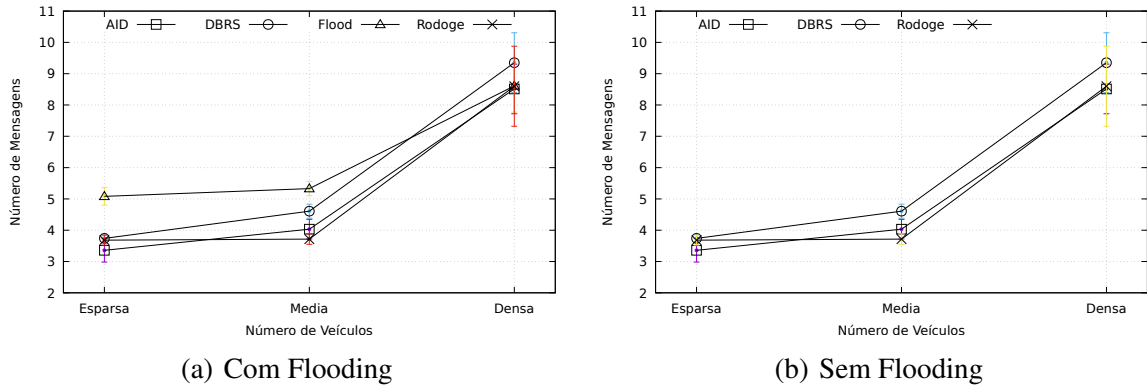
A Figura 6.19 apresenta os resultados da simulação para média de mensagens geradas por cada protocolo considerando a variação da quantidade de veículos (Densa, Média e Esparsa). Na Figura 6.19(a) são apresentados os resultados com Flooding, que foram superiores aos demais protocolos e seguem uma tendência de crescimento com o aumento de veículos. Na Figura 6.19(b) são apresentados os resultados dos outros protocolos. O RODOGE possui resultados melhores que os demais, gerando menor quantidade de mensagens em todas as variações de veículos, cerca de metade que os outros protocolos. A redução do número de mensagens geradas pelo RODOGE está relacionada as regras para reenvio de mensagens utilizadas no trabalho, como a regra para verificar o remetente, que centraliza os reenvios aos dispositivos de infraestrutura e também as regras que incrementam o tempo da mensagem de acordo com sua densidade e distância do remetente.

Figura 6.19 – Cologne 1 Acidente - Média de mensagens geradas por execução.



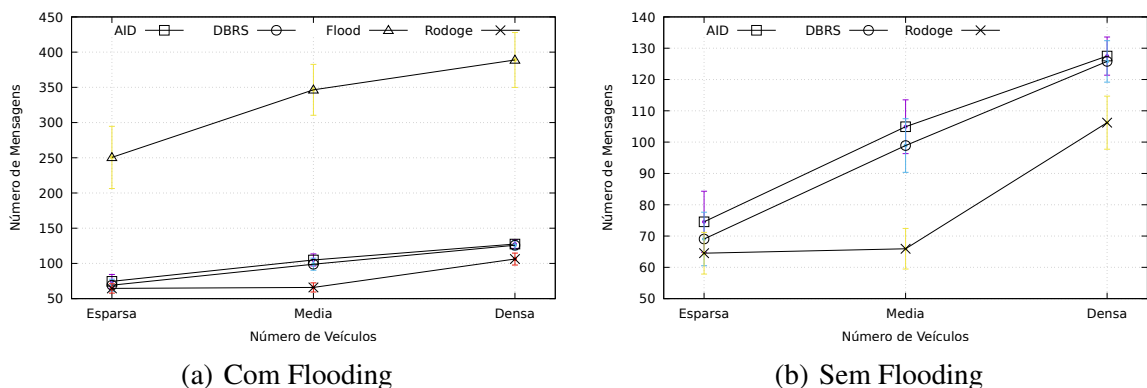
Os resultados para verificar a quantidade média de mensagens geradas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que geraram mensagens, podem ser observados na Figura 6.20. Na Figura 6.20(a) são apresentados os resultados com Flooding, em todas as variações de veículos os resultados foram superiores aos demais protocolos. Na Figura 6.20(b) são apresentados os resultados dos demais protocolos. Os protocolos AID, DBRS e RODOGE obtiveram resultados estatisticamente similares em todas as variações, porém a quantidade total de mensagens geradas, observada em 6.20, é menor em RODOGE, mostrando uma maior concentração de mensagens em apenas alguns nós geradores.

Figura 6.20 – Cologne 1 Acidente - Média de mensagens geradas por retransmissores.



A Figura 6.21 apresenta os resultados da simulação para média de mensagens recebidas por cada protocolo considerando a variação da quantidade de veículos (Esparsa, Média e Densa). Na Figura 6.21(a) são apresentados os resultados do protocolo com Flooding, que foram superiores aos demais protocolos em todas as variações de veículos. Na Figura 6.21(b) é apresentado os relutados dos demais protocolos, na quantidade Esparsa de veículos obtiveram resultados estatisticamente similares e nas demais quantidades o protocolo RODOGE obteve os melhores resultados, com uma quantidade de mensagens recebidas inferior aos demais. Os protocolos AID e DBRS aumentaram significativamente a quantidade de mensagens recebidas, isso se justifica pela maior quantidade de mensagens geradas como pode mostrado na Figura 6.19. A diminuição de mensagens recebidas no protocolo proposto está relacionada a diminuição de mensagens geradas devido aos controles de reenvio que existem no protocolo.

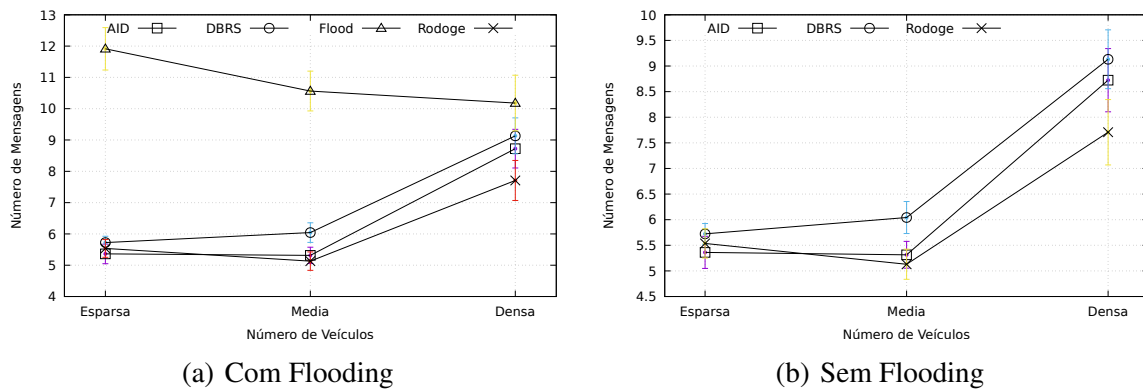
Figura 6.21 – Cologne 1 Acidente - Média de mensagens recebidas por execução.



Os resultados para verificar a quantidade média de mensagens recebidas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que geraram mensagens, podem ser observados na Figura 6.22. Na Figura 6.22(a) são apresentados os resultados com Flooding,

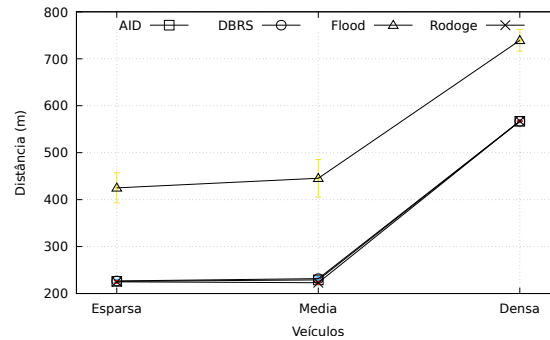
que foram elevados em relação aos demais protocolos para quantidade Esparsa e Média de veículos e estatisticamente similar a todos para quantidade Densa de veículos. Na Figura 6.22(b) são apresentados os resultados detalhados dos demais protocolos, na quantidade Esparsa os protocolos obtiveram resultados estatisticamente similares, na quantidade Média os protocolos AID e RODOGE obtiveram resultados similares e melhores que o DBRS e na quantidade Densa os protocolos obtiveram resultados estatisticamente similares.

Figura 6.22 – Cologne 1 Acidente - Média de mensagens recebidas por receptores.



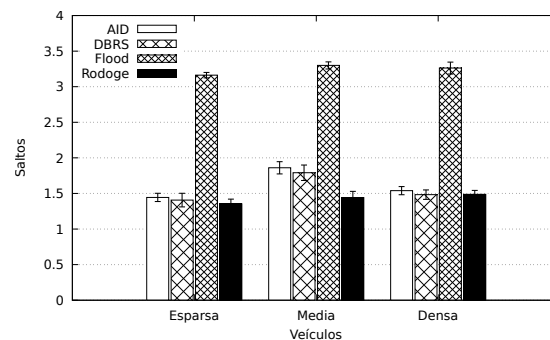
A Figura 6.23 mostra a distância média de recebimento de mensagens pelos veículos. Neste cenário há variações de velocidades devido as diferentes vias presentes no mapa, como base é tomada a velocidade de 110km/h e a distância segura de 91,6m. Os resultados encontrados por todos os protocolos, considerando a variação da quantidade de veículos, foram superiores à distância segura. O protocolo Flooding superou todos os protocolos, apresentando distâncias de recebimento bem acima da distância segura, devido a grande quantidade de mensagens reenviadas por ele, que se espalham por uma área maior que os demais protocolos. Em todas as variações na quantidade de veículos, os protocolos AID, DBRS e RODOGE foram estatisticamente similares e superiores a distância segura. Essa menor distância nos três protocolos se deve ao controle de reenvio de mensagens que eles possuem. Apesar da menor quantidade de mensagens geradas pelo RODOGE, a distância de recebimento está sempre acima da distância segura.

Figura 6.23 – Cologne 1 Acidente - Média do Distância de Recebimento



Para verificar a entrega das mensagens aos veículos com o menor número de intermediários possíveis, foi calculado a média de saltos da mensagem de alerta, como mostrado na Figura 6.24. Em todas as variações de quantidade de veículos o Flooding obteve os resultados mais altos, com maior número de saltos para as mensagens. Para quantidade Esparsa e Densa os demais protocolos obtiveram resultados estatisticamente similares, para Média o RODOGE obteve os melhores resultados, com menor número de saltos das mensagens. Nas variações da quantidade de veículos, o protocolo RODOGE obteve menor variação nos resultados. A justificativa para tal é que o RODOGE centraliza o reenvio pelo dispositivo de infraestrutura, de acordo com a regra de verificar remetente da mensagem no dispositivo de infraestrutura.

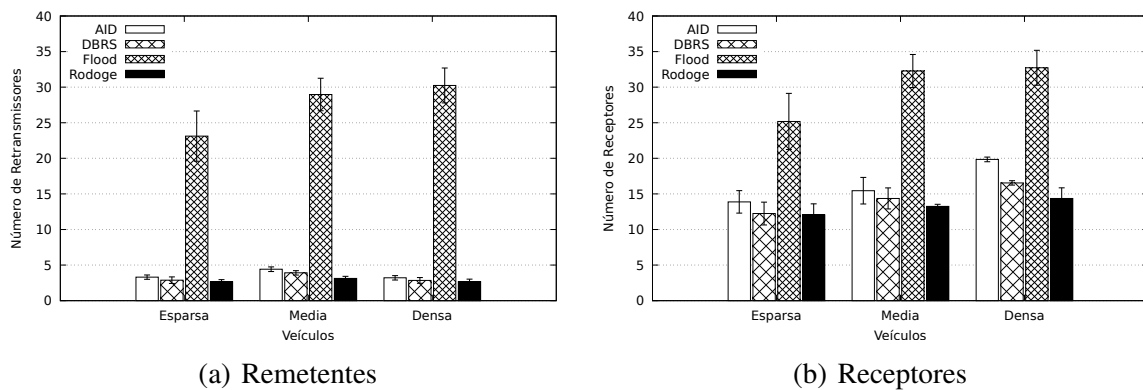
Figura 6.24 – Cologne 1 Acidente - Média de Saltos de Recebimento



A Figura 6.25 mostra as médias de porcentagens de remetentes e receptores de mensagens nas simulações para as variações de veículos. Na Figura 6.25(a) é apresentado a quantidade de veículos que geram e reenviam as mensagens na rede. Em todas as simulações o Flooding obteve a maior quantidade de remetentes, pois não há controles de reenvio de mensagens. Os resultados dos demais protocolos foram estatisticamente similares. O protocolo RODOGE obteve resultados mais constantes, isso é justificado pela aplicação da Regra 3, que limita o reenvio de mensagens pelos veículos dentro da área de cobertura do dispositivo de infraestrutura. A Fi-

gura 6.25(b) mostra os resultados da quantidade de veículos cobertos pela mensagem de alerta. Em todas as simulações o Flooding obteve os maiores resultados, devido ao grande número de mensagens que reenvia e assim alcança mais veículos. O AID, DBRS e RODOGE obtiveram resultados estatisticamente similares para quantidade Esparsa e Média. Para quantidade Densa, o AID obteve resultados superiores ao DBRS e RODOGE, que foram estatisticamente similares.

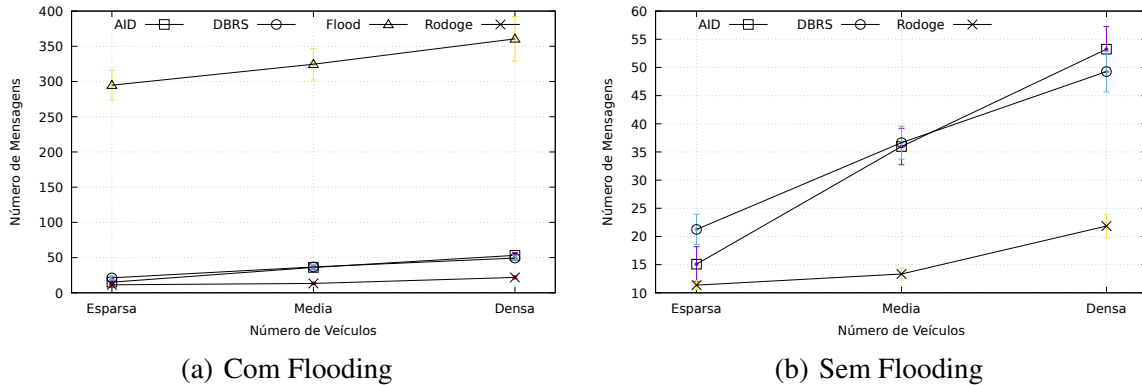
Figura 6.25 – Cologne 1 Acidente - Porcentagens de Remetentes e Receptores.



#### 6.4 Ambiente Cologne com 2 Acidentes

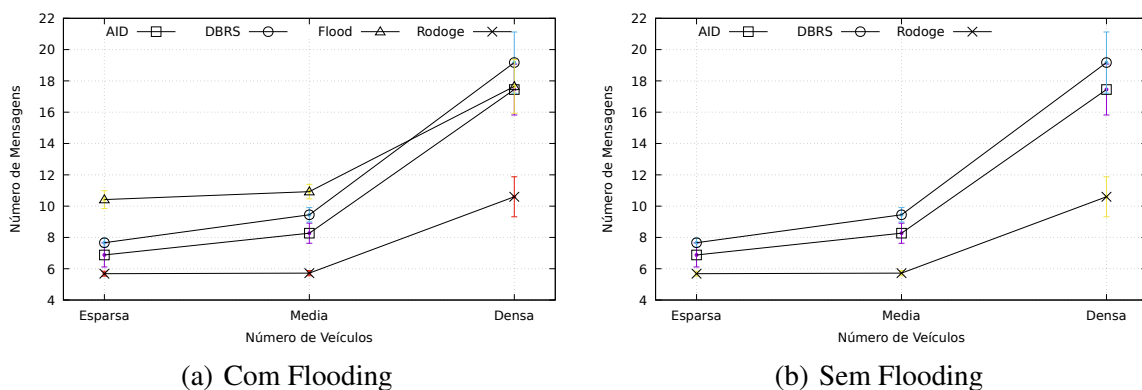
A Figura 6.26 apresenta os resultados da simulação para média de mensagens geradas por cada protocolo considerando a variação da quantidade de veículos (Esparsa, Média e Densa). Na Figura 6.26(a) são apresentados os resultados com Flooding, que são superiores aos demais protocolos em todas as variações de veículos e possui uma tendência de crescimento. Na Figura 6.26(b) são apresentados os resultados sem Flooding, o protocolo RODOGE obteve os melhores resultados em todas as variações de quantidade de veículos, gerando cerca de metade das mensagens geradas pelos outros protocolos. A redução do número de mensagens geradas pelo RODOGE está relacionada às regras para reenvio de mensagens utilizadas no trabalho, como a regra para verificar o remetente, que centraliza os reenvios aos dispositivos de infraestrutura, a regra que compara as mensagens, devido aos dois acidentes que há neste cenário e também as regras que incrementam o tempo da mensagem de acordo com sua densidade e distância do remetente.

Figura 6.26 – Cologne 2 Acidentes - Média de mensagens geradas por execução.



Os resultados para verificar a quantidade média de mensagens geradas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que geraram mensagens, podem ser observados na Figura 6.27. Na Figura 6.27(a) são apresentados os resultados com Flooding, que são superiores aos demais protocolos em todas as variações de veículos. Na Figura 6.27(b) são apresentados os resultados mais detalhados dos demais protocolos, o RODOGE gerou uma quantidade menor de mensagens por nó, quando comparado aos outros protocolos. Esta diferença significativa da média de mensagens geradas por veículo é justificada pelas regras limitantes de reenvio de mensagens do RODOGE, que fazem gerar um total de mensagens menor que os demais protocolos.

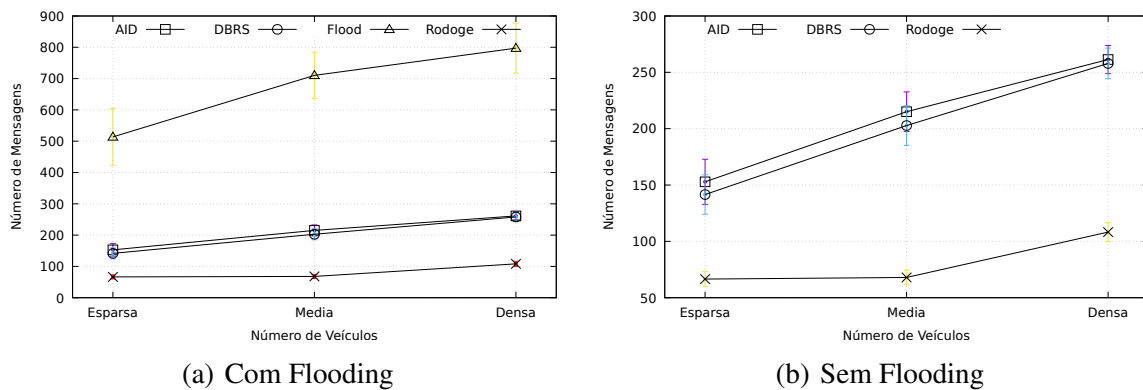
Figura 6.27 – Cologne 2 Acidentes - Média de mensagens geradas por retransmissores.



A Figura 6.28 apresenta os resultados da simulação para média de mensagens geradas por cada protocolo considerando a variação da quantidade de veículos (Esparsa, Média e Densa). Na Figura 6.28(a) é apresentado os resultados dos protocolo com Flooding, que foram superiores aos demais protocolos em todas as variações de veículos. Na Figura 6.28(b) são apresentados os resultados dos demais protocolos. O RODOGE obteve os melhores resultados,

com uma quantidade de mensagens inferior a 120 em todos os cenários. Os protocolos AID e DBRS aumentaram significativamente a quantidade de mensagens recebidas, isso se justifica pela maior quantidade de mensagens geradas, como pode mostrado na Figura 6.26. A diminuição de mensagens recebidas no protocolo proposto está relacionada a diminuição de mensagens geradas devido aos controles de reenvio que ele utiliza.

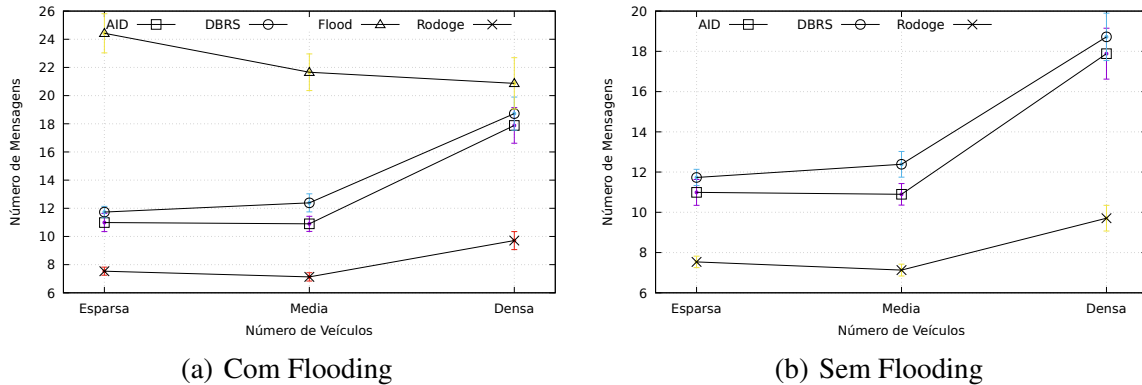
Figura 6.28 – Cologne 2 Acidentes - Média de mensagens recebidas por execução.



O resultado da média de mensagens recebidas, calculado com base na quantidade de veículos e dispositivos de infraestrutura que receberam mensagens nas simulações, pode ser observado na Figura 6.29. Na Figura 6.29(a) são apresentados os resultados com Flooding, que foram elevados em relação aos demais protocolos para quantidade Esparsa e Média de veículos e estatisticamente similar a todos para quantidade Densa de veículos. Na Figura 6.29(b) são apresentados os resultados detalhados dos demais protocolos. Em todas as variações da quantidade de veículos o protocolo RODOGE recebeu uma quantidade média de mensagens inferior aos demais protocolos. A quantidade de mensagens geradas, também influencia na diferença entre os valores médios de mensagens recebidas por veículos entre os protocolos analisados.

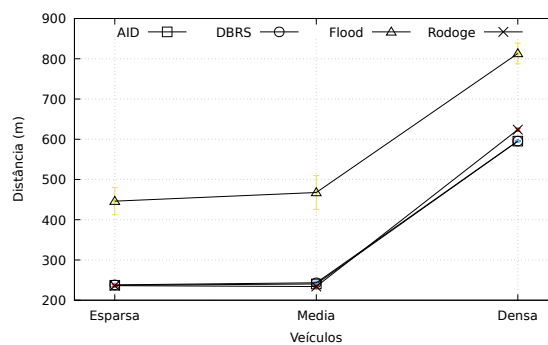


Figura 6.29 – Cologne 2 Acidentes - Média de mensagens recebidas por receptores.



A Figura 6.30 apresenta a distância média de recebimento de mensagens pelos veículos. Neste cenário há variações de velocidades devido as diferentes vias presentes no mapa, como base é tomada a velocidade de 110km/h e a distância segura de 91,6m. Os resultados encontrados por todos os protocolos, considerando a variação da quantidade de veículos, foi superior à distância segura. O protocolo Flooding superou todos os protocolos, apresentando distâncias de recebimento bem acima da distância segura, devido a grande quantidade de mensagens reenviadas por ele, que se espalham por uma área maior que os demais protocolos. Em todas as variações na quantidade de veículos, os protocolos AID, DBRS e RODOGE foram estatisticamente similares e superiores a distância segura. Essa menor distância nos três protocolos se deve ao controle de reenvio de mensagens que eles possuem. Apesar da menor quantidade de mensagens geradas pelo RODOGE, a distância de recebimento está sempre acima da distância segura.

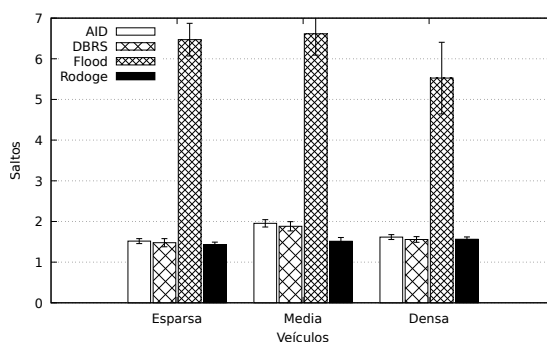
Figura 6.30 – Cologne 2 Acidentes - Média do Distância de Recebimento



Para verificar a entrega das mensagens aos veículos com o menor número de intermediários possíveis, foi calculado a média de saltos da mensagem de alerta, como mostrado na Figura 6.31. No cenário com quantidade Esparsa de veículos, o Flooding obteve os piores resultados,

com maior número de saltos, e os demais protocolos obtiveram resultados estatisticamente similares. Para quantidade Média de veículos o Flooding manteve os piores resultados, AID e DBRS obtiveram resultados estatisticamente similares e RODOGE obteve os melhores resultados, com menor número médio de saltos. Para quantidade Densa de veículos os protocolos AID, DBRS e RODOGE obtiveram resultados estatisticamente similares e o Flooding manteve os resultados superiores. O RODOGE manteve uma constância nos resultados, a justificativa para controle na quantidade de saltos é devido a centralização de reenvio pelo dispositivo de infraestrutura, de acordo com a regra de verificar remetente da mensagem.

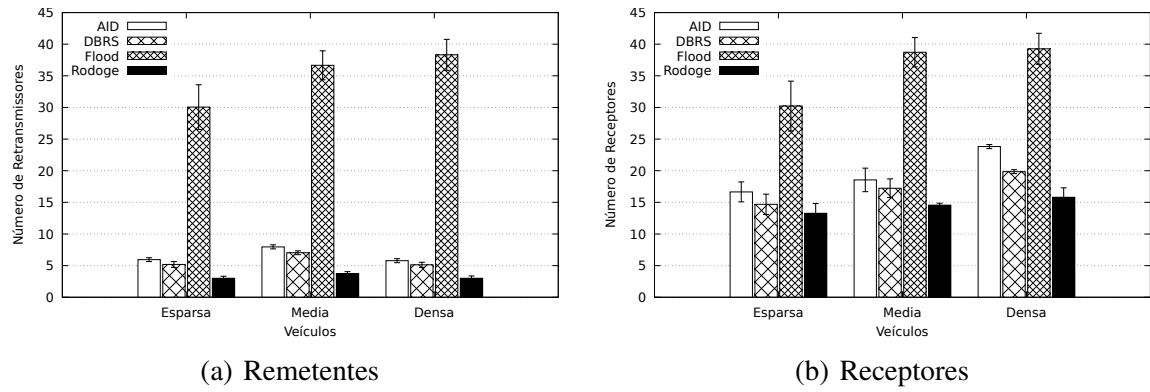
Figura 6.31 – Cologne 2 Acidentes - Média de Saltos de Recebimento



A Figura 6.32 mostra as médias de porcentagens de remetentes e receptores de mensagens nas simulações para as variações de veículos. Na Figura 6.32(a) é apresentada a quantidade de nós que reenviam as mensagens de alerta. Em todas as simulações o Flooding obteve a maior porcentagem de remetentes, pois não há controles de reenvio de mensagens. Os resultados dos protocolos AID e DBRS foram estatisticamente similares. O RODOGE obteve os melhores resultados, com menor número de remetentes de mensagens, tais resultados estão relacionados a utilização da Regra 3, que limita o reenvio de mensagens pelos veículos dentro da área de cobertura do dispositivo de infraestrutura. A Figura 6.32(b) mostra os resultados da porcentagem de veículos cobertos pela mensagem de alerta. Em todas as simulações o Flooding obteve as maiores porcentagens, devido ao grande número de mensagens que reenvia e assim alcança nós mais rapidamente. O AID, DBRS e RODOGE obtiveram resultados estatisticamente similares para quantidade Esparsa de veículos. Para quantidade Média de veículos, AID e DBRS obtiveram resultados similares e superiores ao RODOGE. Para quantidade Densa de veículos, AID obteve resultados superiores ao DBRS, que obteve resultados superiores ao RODOGE. Além do RODOGE reduzir a quantidade de mensagens reenviadas, o percentual de veículos que recebem

a mensagem de alerta em distância segura é semelhante ao dos protocolos que reenviam mais mensagens.

Figura 6.32 – Cologne 2 Acidentes - Porcentagens de Remetentes e Receptores.



## 7 CONCLUSÃO

O protocolo RODOGE diminuiu a quantidade de mensagens reenviadas utilizando as regras implementadas, sem reduzir drasticamente a área de cobertura do alerta de acidente e com distância de recebimento de alerta de acidente suficiente para reação do motorista e frenagem do veículo. A Regra 1 tratou os dois veículos acidentados como apenas um evento, o que acarreta em um menor número de mensagens reenviadas. A Regra 2 limitou o tempo de vida da mensagem, fazendo com que ela fosse reenviada apenas durante o acidente e assim diminuindo a quantidade de mensagens reenviadas. A Regra 3 centralizou o reenvio de mensagens em áreas cobertas por dispositivos de infraestrutura, reduzindo o número de mensagens reenviadas nessas regiões. As Regras 4 e 5 controlaram o tempo de reenvio das mensagens, na qual influenciaram na escolha de veículos que receberam menor número de mensagens duplicadas e veículos mais distantes dos remetentes. Os resultados das simulações nos dois cenários propostos mostraram que o RODOGE reenvia menor quantidade de mensagens, quando comparado aos protocolos clássicos AID, DBRS e Flooding, mesmo aumentando o número de veículos. Além disso, o RODOGE mesmo reduzindo a quantidade de mensagens reenviadas, possui o percentual de veículos que recebem a mensagem de alerta em distância segura semelhante aos protocolos que reenviam mais mensagens.

Como trabalhos futuros, pretende-se realizar outras simulações considerando a interligação de vários dispositivos de infraestrutura para propagar as mensagens de acidente, realizar um estudo para otimizar a alocação dos dispositivos de infraestrutura, realizar simulações com maior número de acidentes em diferentes locais e usar outros cenários reais de rastreamento de veículos em áreas urbanas.

## REFERÊNCIAS

- AL-SULTAN, S. et al. A comprehensive survey on vehicular ad hoc network. **J. Netw. Comput. Appl.**, Academic Press Ltd., London, UK, UK, v. 37, p. 380–392, jan. 2014. ISSN 1084-8045. Disponível em: <<http://dx.doi.org/10.1016/j.jnca.2013.02.036>>.
- ALLAL, S.; BOUDJIT, S. Geocast routing protocols for vanets: Survey and guidelines. In: **2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing**. [S.l.: s.n.], 2012. p. 323–328.
- ALVES, R. S. et al. Redes veiculares: Princípios, aplicações e desafios. In: SBRC, M. do Simpósio Brasileiro de Redes de C. (Ed.). [S.l.: s.n.], 2009. cap. 5, p. 199–254.
- AMARANTE, T. C. et al. Detecção automática e alerta de acidentes em redes veiculares reais. **XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, 2015.
- BAKHOUYA, M.; GABER, J.; LORENZ, P. An adaptive approach for information dissemination in vehicular ad hoc networks. **Journal of Network and Computer Applications**, v. 34, n. 6, p. 1971 – 1978, 2011. ISSN 1084-8045. Control and Optimization over Wireless Networks.
- BARCELOS, V. P. **ANÁLISE E EXPERIMENTAÇÃO DO PADRÃO IEEE 802.11P EM REDES VEICULARES HÍBRIDAS**. [S.l.]: Dissertação de Mestrado em Ciência da Computação, Universidade Federal de Lavras, Lavras, 2014. 106 p.
- BRACCIALE L., B. M. L. P. B. G. A. R.; RABUFFI, A. **CRAWDAD data set roma/taxi (v. 2014-07-17)**. 2014. Disponível em: <<http://crawdad.org/roma/taxi/>>. Acesso em: 02/10/2018.
- CHA, S.-H. **A Survey of Broadcast Protocols for Vehicular Ad-hoc Networks**. [S.l.]: Smart Computing Review, 2014. 9 p.
- CHEN, P. Y.; LIU, J. W.; CHEN, W. T. A fuel-saving and pollution-reducing dynamic taxi-sharing protocol in vanets. In: **Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd**. [S.l.: s.n.], 2010. p. 1–5. ISSN 1090-3038.
- CHEN, Y.; SHEN, K.; WANG, S. Forward collision warning system considering both time-to-collision and safety braking distance. In: **2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)**. [S.l.: s.n.], 2013. p. 972–977. ISSN 2156-2318.
- CHOWDHURY, N. M.; MACKENZIE, L.; PERKINS, C. Requirement analysis for building practical accident warning systems based on vehicular ad-hoc networks. In: **Wireless On-demand Network Systems and Services (WONS), 2014 11th Annual Conference on**. [S.l.: s.n.], 2014. p. 81–88.
- CUNHA, L. H. C. S. e Raquel A. F. Mini e F. D. Um protocolo de disseminação de dados geo-orientado em redes veiculares. **Workshop de Computação Urbana**, v. 1, n. 1/2017, 2017. ISSN 2595-2706. Disponível em: <<https://portaldeconteudo.sbc.org.br/index.php/courb/article/view/2587>>.
- FAROOQ, W. et al. A survey of multicast routing protocols for vehicular ad hoc networks. **International Journal of Distributed Sensor Networks**, v. 11, n. 8, p. 923086, 2015. Disponível em: <<https://doi.org/10.1155/2015/923086>>.

FERREIRO-LAGE, J. A. et al. Analysis of unicast routing protocols for vanets. In: **2009 Fifth International Conference on Networking and Services**. [S.l.: s.n.], 2009. p. 518–521.

FESTAG, A. et al. **NoW – Network on Wheels: Project Objectives, Technology and Achievements**. 2008.

HARTENSTEIN, H.; LABERTEAUX, L. P. A tutorial survey on vehicular ad hoc networks. **IEEE Communications Magazine**, v. 46, n. 6, p. 164–171, June 2008. ISSN 0163-6804.

IBNEZ, A. G. et al. A performance study of the 802.11p standard for vehicular applications. In: **Intelligent Environments (IE), 2011 7th International Conference on**. [S.l.: s.n.], 2011. p. 165–170.

IEEE1609.0. Ieee guide for wireless access in vehicular environments (wave) - architecture. **IEEE Std 1609.0-2013**, p. 1–78, March 2014.

IEEE1609.1. Trial-use standard for wireless access in vehicular environments (wave) - resource manager. **IEEE Std 1609.1**, p. 1–71, Oct 2006.

IEEE1609.2. Ieee standard for wireless access in vehicular environments–security services for applications and management messages. **IEEE Std 1609.2-2016 (Revision of IEEE Std 1609.2-2013)**, p. 1–240, March 2016.

IEEE1609.3. Ieee standard for wireless access in vehicular environments (wave) – networking services. **IEEE Std 1609.3-2016 (Revision of IEEE Std 1609.3-2010)**, p. 1–160, April 2016.

JIANG, D.; DELGROSSI, L. Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In: **VTC Spring 2008 - IEEE Vehicular Technology Conference**. [S.l.: s.n.], 2008. p. 2036–2040. ISSN 1550-2252.

KARAGIANNIS, G. et al. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. **IEEE Communications Surveys Tutorials**, v. 13, n. 4, p. 584–616, Fourth 2011. ISSN 1553-877X.

KHAN, L.; AYUB, N.; SAEED, A. Anycast based routing in vehicular adhoc networks ( vanets ) using vanetmobisim. In: . [S.l.: s.n.], 2013.

KIM, T.-H. et al. An effective data dissemination in vehicular ad-hoc network. In: VAZÃO, T.; FREIRE, M. M.; CHONG, I. (Ed.). **Information Networking. Towards Ubiquitous Networking and Services**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 295–304. ISBN 978-3-540-89524-4.

KORKMAZ, G. et al. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In: **Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks**. New York, NY, USA: ACM, 2004. (VANET '04), p. 76–85. ISBN 1-58113-922-5. Disponível em: <<http://doi.acm.org/10.1145/1023875.1023887>>.

KUROSE, J.; ROSS, K. **Redes de computadores e a Internet: uma abordagem top-down**. ADDISON WESLEY BRA, 2007. ISBN 9788588639188. Disponível em: <<https://books.google.com.br/books?id=i5WwAAAACAAJ>>.

LI, F.; WANG, Y. Routing in vehicular ad hoc networks: A survey. **IEEE Vehicular Technology Magazine**, v. 2, n. 2, p. 12–22, June 2007. ISSN 1556-6072.

MEJRI, M. N.; BEN-OTHMAN, J.; HAMDI, M. Survey on vanet security challenges and possible cryptographic solutions. **Vehicular Communications**, v. 1, n. 2, p. 53 – 66, 2014. ISSN 2214-2096. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2214209614000187>>.

MENEGUETTE, R. I. et al. Autonomic data dissemination in highway vehicular ad hoc networks with diverse traffic conditions. In: **2014 IEEE Symposium on Computers and Communications (ISCC)**. [S.l.: s.n.], 2014. p. 1–6. ISSN 1530-1346.

NAKORN, N. N.; ROJVIBOONCHAI, K. Deca: Density-aware reliable broadcasting in vehicular ad hoc networks. In: **ECTI-CON2010: The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology**. [S.l.: s.n.], 2010. p. 598–602.

NAUMOV, V.; BAUMANN, R.; GROSS, T. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In: **Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing**. New York, NY, USA: ACM, 2006. (MobiHoc '06), p. 108–119. ISBN 1-59593-368-9. Disponível em: <<http://doi.acm.org/10.1145/1132905.1132918>>.

NETO, J. B. P. et al. An error correction algorithm for forward collision warning applications. In: **2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2016. p. 1926–1931. ISSN 2153-0017.

OMNET++. **OMNeT++**. 2018. Disponível em: <<http://omnet-tutorial.com/>>. Acesso em: 02/10/2018.

PIGNÉ, Y.; DANOY, G.; BOUVRY, P. A vehicular mobility model based on real traffic counting data. In: \_\_\_\_\_. **Communication Technologies for Vehicles: Third International Workshop, Nets4Cars/Nets4Trains 2011, Oberpfaffenhofen, Germany, March 23-24, 2011. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 131–142. ISBN 978-3-642-19786-4.

PIORKOWSKI, M.; SARAFIJANOVIC-DJUKIC, N.; GROSSGLAUSER, M. **CRAWDAD data set epfl/mobility (v. 2009-02-24)**. 2009. Disponível em: <<http://crawdad.cs.dartmouth.edu/epfl/mobility>>. Acesso em: 02/10/2018.

QI, W. et al. Sdn-enabled social-aware clustering in 5g-vanet systems. **IEEE Access**, v. 6, p. 28213–28224, 2018. ISSN 2169-3536.

SICHITIU, M. L.; KIHLE, M. Inter-vehicle communication systems: a survey. **IEEE Communications Surveys Tutorials**, v. 10, n. 2, p. 88–105, Second 2008. ISSN 1553-877X.

SOMMER, C. **Veins - The open source vehicular network simulation framework**. 2016. Disponível em: <<http://veins.car2x.org/>>. Acesso em: 02/10/2018.

SOUZA, A. M.; MAIA, G.; VILLAS, L. A. Add: A data dissemination solution for highly dynamic highway environments. In: **2014 IEEE 13th International Symposium on Network Computing and Applications**. [S.l.: s.n.], 2014. p. 17–23.

SUMO. **Simulation of Urban MObility (SUMO)**. 2016. Disponível em: <[http://sumo.dlr.de/wiki/Mainz\\_Page](http://sumo.dlr.de/wiki/Mainz_Page)>. Acesso em: 02/10/2018.

- TEHRANI, M.; UYSAL, M.; YANIKOMEROGLU, H. Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions. **IEEE Communications Magazine**, v. 52, n. 5, p. 86–92, 2014. ISSN 01636804.
- TONGUZ, O. K.; WISITPONGPHAN, N.; BAI, F. Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. **IEEE Wireless Communications**, v. 17, 2010.
- TOSTES, A. I. J. et al. From data to knowledge: city-wide traffic flows analysis and prediction using bing maps. In: ACM. **Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing**. [S.l.], 2013. p. 12.
- TSENG, Y.-C.; NI, S.-Y.; SHIH, E.-Y. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. In: **Proceedings 21st International Conference on Distributed Computing Systems**. [S.l.: s.n.], 2001. p. 481–488.
- UNIVERSITY, S. J. T. **Suvnet-trace data**. 2007. Disponível em: <<http://wirelesslab.sjtu.edu.cn/download.html>>. Acesso em: 02/10/2018.
- VARSCHEN, C.; WAGNER, P. Mikroskopische modellierung der personenverkehrsnachfrage auf basis von zeitverwendungstagebüchern. In: BECKMANN, K. J. (Ed.). **AMUS 2006 (7. Aachener Kolloquium "Mobilität und Stadt")**. [S.l.]: Institut für Stadtbauwesen und Stadtverkehr, RWTH Aachen, 2006. (Stadt Region Land, v. 81), p. 63–69.
- VEGNI, A. M.; LITTLE, T. D. C. A message propagation model for hybrid vehicular communication protocols. In: **Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on**. [S.l.: s.n.], 2010. p. 382–386.
- VILLAS, L. A. et al. Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks. **Computer Networks**, v. 75, p. 381 – 394, 2014. ISSN 1389-1286.
- VILLAS, L. A. et al. An efficient and robust data dissemination protocol for vehicular ad hoc networks. In: **Proceedings of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks**. New York, NY, USA: ACM, 2012. (PE-WASUN '12), p. 39–46. ISBN 978-1-4503-1621-7. Disponível em: <<http://doi.acm.org/10.1145/2387027.2387035>>.
- WHO. **World Health Organization (WHO)**. 2018. Disponível em: <<http://www.who.int/en/news-room/fact-sheets/detail/road-traffic-injuries>>. Acesso em: 02/10/2018.
- WISITPONGPHAN, N. et al. Broadcast storm mitigation techniques in vehicular ad hoc networks. **IEEE Wireless Communications**, v. 14, n. 6, p. 84–94, December 2007. ISSN 1536-1284.
- XEROS, A. et al. Adaptive probabilistic flooding for information hovering in vanets. In: **2010 IEEE Vehicular Networking Conference**. [S.l.: s.n.], 2010. p. 239–246. ISSN 2157-9865.
- ZAFOUNE, Y.; KANAWATI, R.; MOKHTARI, A. Mobile agents localization in ad hoc networks: a comparative study of centralized and distributed approaches. In: **2007 ITI 5th International Conference on Information and Communications Technology**. [S.l.: s.n.], 2007. p. 269–275. ISSN 2329-6364.
- ZEADALLY, S. et al. Vehicular ad hoc networks (vanets): status, results, and challenges. **Telecommunication Systems**, v. 50, n. 4, p. 217–241, Aug 2012. ISSN 1572-9451.