

# RODOGE: Protocolo de disseminação de mensagens de alerta de acidentes com regras e controles de reenvio

Paulo H. Almeida<sup>1</sup>, Gilson Miranda Jr.<sup>2</sup>, Raphael B. Wincler<sup>1</sup>, Luiz H. A. Correia<sup>1</sup>

<sup>1</sup> Departamento de Ciência da Computação  
Universidade Federal de Lavras (UFLA) – Lavras – MG – Brazil

<sup>2</sup> Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte – MG – Brazil

paulohalmeidaa@gmail.com, {lcorreia, raphaelwb}@dcc.ufla.br,  
gilsonmiranda@dcc.ufmg.br

**Abstract.** *This paper proposes a protocol for the dissemination of alert messages with replication control called RODOGE (Rules to prevent Over-Dissemination Of messaGEs). The rules and functions applied by RODOGE limit the dissemination of messages by mobile devices and infrastructure, avoiding the broadcast storm problem. RODOGE has been tested and compared to other messaging protocols through simulations using OMNeT ++, SUMO and Veins with urban and road traffic. Results show that RODOGE reduces the amount of messages generated and received, avoids the broadcast storm and still maintains enough message coverage area for drivers to react to an event.*

**Resumo.** *Este artigo propõe um protocolo para disseminação de mensagens de alerta com controle de replicação chamado RODOGE (Rules to prevent Over-Dissemination Of messaGEs). As regras e funções aplicadas pelo RODOGE limitam a disseminação de mensagens por dispositivos móveis e de infraestrutura, evitando o problema de broadcast storm. O RODOGE foi testado e comparado com outros protocolos de disseminação de mensagens através de extensas simulações usando OMNeT ++, SUMO e Veins com tráfego urbano e rodoviário. Resultados mostram que o RODOGE reduz a quantidade de mensagens geradas e recebidas, evita o broadcast storm e ainda mantém uma área de cobertura de mensagens suficiente para que motoristas reajam a um evento.*

## 1. Introdução

Todos os anos são registrados milhares de acidentes automobilísticos em todo o mundo, muitos com vítimas fatais e feridos [WHO 2018]. Nos últimos anos, a indústria automobilística tem buscado aprimorar a segurança do motorista e mitigar o número de acidentes, instalando nos veículos dispositivos de segurança, como freios ABS, airbags, sistemas de controle de estabilidade e tração, sensores de ultrapassagem, etc. [Pögel and Wolf 2012].

Com a evolução das tecnologias de comunicação sem fio, a proposta de utilizar uma rede de comunicação entre veículos tem sido amplamente pesquisada pelo meio acadêmico e industrial. As redes veiculares, ou VANET (*Vehicular Ad hoc Network*), emergem como uma solução para a comunicação veicular, visando evitar situações de risco, auxiliar em caso de acidentes, prevenir colisões, controlar congestionamentos e

disponibilizar aplicações de entretenimento [Al-Sultan et al. 2014]. A troca de mensagens nas VANET pode ser entre veículos V2V (*Vehicle to Vehicle*), entre dispositivos de infraestrutura e veículos V2I (*Vehicle to Infrastructure*) ou por uma comunicação híbrida V2X. Nos veículos são instalados dispositivos de comunicação sem fio conhecidos como OBU (*On-Board Unit*) e nas infraestruturas o RSU (*Road Side Unit*). A arquitetura WAVE (*Wireless Access in Vehicular Environments*) tem direcionado o desenvolvimento das VANET, por meio de um conjunto de documentos que padronizam as regras de comunicação, segurança, tipos de aplicação e de controle de acesso ao meio [Group 2014].

As redes veiculares possuem características diferentes das demais redes, como topologia altamente dinâmica, desconexões frequentes, mobilidade restrita e locomoção previsível. As mensagens trocadas na VANET são fornecidas pelos veículos, sem necessariamente depender de intervenção humana, e podem conter dados para ajudar na detecção e prevenção de acidentes, alerta de congestionamento e outras informações relevantes para os motoristas. As mensagens de alerta devem ser entregues a todos os veículos de interesse devido à sua natureza de emergência, que causa um dos principais problemas nas VANET que é a inundação de mensagens na rede, chamada de *Broadcast Storm* (BS) [Tseng et al. 2001]. Como consequência, o BS pode ocasionar congestionamentos dos enlaces, perdas de mensagens, atrasos, contenções e colisões, e no pior caso inviabilizar a comunicação entre os dispositivos.

Na literatura são encontrados vários protocolos de disseminação de mensagens que visam solucionar o problema do BS, como o DBRS, AID, GEDDAI e ADDHV [Kim et al. 2008, Bakhouya et al. 2011, Villas et al. 2012, Meneguetta et al. 2014]. Apesar dos resultados apresentados para controle da disseminação de dados, esses protocolos não lidam com mais de um dispositivo de infraestrutura para a disseminação de mensagens de acidentes, nem com diferentes densidades de tráfego ou de cenários.

As redes 5G têm potencial para fornecer comunicações entre veículos do tipo D2D (*Device to Device*). Com uma área de cobertura mais ampla, baixa latência e altas taxas de dados, essas redes devem oferecer melhor suporte a dispositivos com alta mobilidade, como os veículos [Qi et al. 2018]. Espera-se que o 5G explore os benefícios da comunicação D2D, permitindo que os dispositivos se comuniquem diretamente com seus vizinhos na banda licenciada. Neste contexto, as Estações-Base podem ter diferentes níveis de envolvimento, desde o modo convencional (no qual o controle e os dados são passados pela Estação-Base), até deixar os dispositivos completamente responsáveis pelo estabelecimento da comunicação e troca de dados [Tehrani et al. 2014]. A tecnologia 5G fomentará a utilização de RSUs como parte da VANET para reduzir a replicação descontrolada de mensagens.

Este trabalho propõe um protocolo de disseminação controlada de mensagens, denominado de RODOGE - (*Rules that Oppose overdone Dissemination Of messaGEs*), que utiliza um conjunto de regras e funções para controlar o reenvio da mensagens de acidentes. O RODOGE propõe a utilização de OBU e RSU para controlar o reenvio de mensagens e lida indistintamente com diferentes densidades de tráfego urbano e rodoviário. As regras para controle de reenvio de mensagens são baseadas em informações contidas na própria mensagem de alerta, como: tempo de vida, remetente, *backoff* e distância do remetente, e desambiguação de mensagens de acidentes. As principais contribuições do RODOGE em relação aos outros protocolos são:

- i. Considerar a RSU para mitigar o *broadcast storm*.
- ii. Lidar com diferentes cenários de tráfego urbano e rodoviário.
- iii. Realizar simulações com mais de um acidente nas vias, considerando duas ou mais mensagens de um mesmo acidente como apenas uma mensagem.
- iv. Considerar o tempo de reação do motorista diante da propagação de mensagens de alerta de acidentes.
- v. Ajustar dinamicamente o tempo de vida da mensagem de alerta de acordo com velocidade do veículo.
- vi. Ajustar dinamicamente o intervalo de envio das mensagens de acordo com a densidade da rede e da distância do emissor.

Para verificar a efetividade do RODOGE, foram realizadas simulações exaustivas nos ambientes OMNeT++, SUMO e Veins considerando diferentes tipos de tráfego em cenários urbanos e rodoviários. O RODOGE foi comparado com os protocolos clássicos de controle de disseminação de mensagens AID, DBRS e Flooding. Resultados mostraram que o RODOGE reduz a quantidade de mensagens geradas e recebidas, evitando o BS, e mantendo uma área de cobertura de mensagens suficiente para possibilitar o tempo de reação ao motorista.

Este artigo está organizado como descrito a seguir. A Seção 2 apresenta os principais protocolos para controle de BS. As funcionalidades e regras do protocolo RODOGE são descritas na seção 3. A Seção 4 apresenta as ferramentas, métricas e cenários utilizados para avaliar os protocolos de disseminação de mensagens. Os resultados e discussões dos protocolos avaliados são apresentados na Seção 5. Finalmente, a Seção 6 apresenta as conclusões da avaliação do RODOGE.

## 2. Trabalhos Relacionados

A aplicação ACORE (*Algorithm for Collision waRning Error correction*) analisa e envia mensagens de uma possível colisão com um veículo a frente [Neto et al. 2016]. Os veículos transmitem mensagens periodicamente com informações sobre sua posição geográfica e velocidade. Quando um veículo recebe uma nova mensagem, ele analisa a sua distância para o remetente e compara o resultado com a distância segura de frenagem. Essa distância é calculada considerando a eficiência do sistema de freio, peso do veículo, ângulo de inclinação da estrada, etc. [Chen et al. 2013]. Se a distância calculada entre os veículos for menor que a distância segura, um alerta de colisão é emitido para o motorista. A ACORE limita-se a emitir um alerta de possível colisão entre dois veículos, sem mecanismos para retransmitir alertas para outros veículos ou infraestruturas próximos.

Para reduzir as retransmissões de mensagens na rede, o protocolo ADD (*Data Dissemination Solution for Highly Dynamic Environments*) [Souza et al. 2014], seleciona veículos em zonas de preferências para realizar a disseminação de mensagens. Esta zona é definida por veículos mais propensos a disseminar a mensagem aos seus vizinhos, que são selecionados para realizar a retransmissão. Em casos de zona de preferência vazia, os veículos mais distantes do veículo de origem retransmitem o alerta de acidente.

Os autores [Feukeu and Zuva 2017] propuseram o protocolo DBSMA (*Dynamic Broadcast Storm Mitigation Algorithm*) para combater o problema do BS para mensagens de segurança, transmitidas em 10 MHz, do tipo CAM (Cooperative Awareness Message). O DBSMA considera uma distância de segurança baseada em uma regra de 3 segundos,

bem como uma velocidade móvel para computar um tempo de transmissão de mensagem de segurança correspondente. Apesar dos autores reivindicarem que o DBSMA consegue reduzir amplamente o BS, não são considerados na simulação no MATLAB, parâmetros de mobilidade, velocidade e a densidade da redes de veículos.

O protocolo AID (*Approach for Information Dissemination*) possui um controle para diminuir a quantidade de mensagens encaminhadas na rede [Bakhouya et al. 2011]. Esse controle é baseado na densidade de mensagens, que decide reenviar ou não a mensagem, de acordo com a quantidade de mensagens duplicadas recebidas em um determinado intervalo de tempo. O problema do AID é o particionamento de rede, quando o número de veículos na área de interesse não é suficiente para realizar retransmissões de mensagens.

O protocolo DBRS (*Distance Based Relay Selection*) é baseado na distância entre os veículos de origem e de destino [Kim et al. 2008]. O intervalo de tempo utilizado para retransmissão é inversamente proporcional à distância entre os veículos. Essa abordagem seleciona os veículos mais distantes como retransmissores de mensagens. Uma retransmissão de mensagem pode ser cancelada caso um veículo receba uma mensagem antes de realizar o seu reenvio e assim um novo intervalo de tempo é calculado com base na distância. Dessa forma, o novo intervalo calculado pode ser maior do que o necessário para fornecer uma cobertura adequada de mensagens para novos nós em uma região.

O protocolo utilizado como base para a disseminação de mensagens é conhecido como Flooding. A ideia deste protocolo é transmitir mensagens para todos os vizinhos, que armazenam as mensagens e as retransmitem. Este processo é repetido por todos os nós que recebem a mensagem. Embora seja a maneira mais rápida de disseminar uma mensagem em redes veiculares, o protocolo Flooding gera um alto número de retransmissões e, portanto, gera o problema de *broadcast storm* [Xeros et al. 2010].

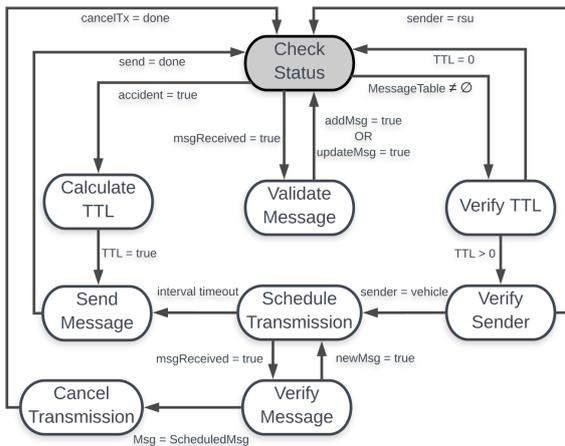
Apesar das soluções propostas diminuïrem a quantidade de mensagens geradas, os protocolos atuais não levam em conta dois ou mais acidentes ocorridos simultaneamente e com dois ou mais veículos acidentados gerando mensagens.

### 3. Protocolo RODOGE

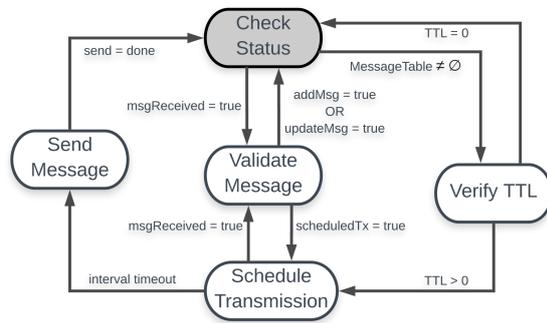
O protocolo RODOGE tem como objetivo principal reduzir e controlar o reenvio de mensagens em uma rede veicular, para evitar a conhecida *broadcast storm*. As mensagens de alerta de acidentes são retransmitidas para a rede somente quando um conjunto de regras é atendido. A aplicação dessas regras é baseada em informações trocadas pelos veículos, e para isso, o protocolo especifica mensagens com os seguintes campos:

`idMessage`: identificador único da mensagem, gerado pelo veículo origem;  
`messageOrigin`: número de identificação único do veículo origem;  
`position`: posição do GPS do veículo origem;  
`time`: momento em que a mensagem é criada;  
`lastPosition`: posição do GPS do veículo de último salto da mensagem;  
`type`: tipo de remetente da mensagem, veículo ou dispositivo de infraestrutura (RSU);  
`hops`: número de saltos que a mensagem percorreu.

É possível utilizar um campo na mensagem para que ela seja utilizada não apenas para alertas de colisões.



**Figura 1. Máquina de estados para funcionamento dos veículos.**



**Figura 2. Máquina de estados para funcionamento das RSUs.**

Alguns campos da mensagem são preenchidos na sua criação, com dados do veículo acidentado (origem), e são retransmitidos sem modificação pelos outros nós: *idMessage*, *messageOrigin*, *position* e *time*. Outros campos são atualizados pelo nó que replica a mensagem: *lastPosition*, *type*, *TTL* e *hops*.

As mensagens recebidas são armazenadas em uma tabela *hash*, que também armazena os seguintes dados de controle:

- timeReceive*: momento em que a mensagem foi recebida;
- latency*: diferença de tempo entre a transmissão e a recepção da mensagem;
- backoff*: valor usado no cálculo *backoff* da mensagem;
- distance*: distância entre remetente e receptor.

Para controlar o funcionamento do protocolo e a replicação de mensagens, um conjunto de regras determina a disseminação de mensagens:

**Regra #1:** As mensagens de alerta na tabela de cada nó possuem um número de identificação único e a posição geográfica de um acidente. Dessa forma, mensagens de alerta com o mesmo identificador e com posições geográficas próximas são tratadas como sendo de um mesmo acidente.

**Regra #2:** O tempo de vida da mensagem deve ser maior e inversamente proporcional à velocidade do veículo (Equação 2) para delimitar o tempo para os reenvios.

**Regra #3:** Na área de cobertura de um dispositivo de infraestrutura (RSU), os veículos não retransmitem mensagens.

**Regra #4:** O tempo de reenvio de mensagens dos veículos deve ser proporcional ao número de mensagens enviadas e recebidas.

**Regra #5:** O intervalo de retransmissão de mensagens é inversamente proporcional à distância entre o emissor e o receptor.

As regras definem políticas que as funções do protocolo devem aplicar. Um conjunto de funções aplica as regras e controla a operação do RODOGE. As máquinas de estado das Figuras 1 e 2 descrevem o funcionamento nos veículos e nas RSUs, respectivamente.

**Função #1: Check Status** - verifica as atividades e eventos nos elementos da rede. Nas RSUs esta função verifica se o nó recebeu uma nova mensagem ou se a tabela de mensagens não está vazia. No primeiro caso, a máquina segue para o estado *Validate Message*

para processar a mensagem. No segundo, a máquina segue para *Verify TTL*, verificando se alguma das mensagens na tabela deve ser transmitida naquele momento. Nos veículos, a função *Check Status* também verifica a ocorrência de um acidente com base no estado do *air bag* e no tempo em que o veículo permanece parado. Após a detecção de um acidente, a máquina segue para o estado *Calculate TTL*. Nos demais casos, as máquinas de estados dos veículos e das RSU operam de forma similar.

Ainda em *Check Status*, é calculado o tempo padrão para a retransmissão de mensagens, que considera a distância de reação do motorista e a distância de frenagem do veículo, que variam com a velocidade [Chen et al. 2013]. Na Equação 1, o tempo é calculado com base no limite de velocidade da via e na distância de reação, levando em consideração que o motorista deve receber o alerta com tempo suficiente para reagir.

$$TimeMsg = SafeDistance / Speed \quad (1)$$

**Função #2: Calculate TTL** - o tempo de vida da mensagem é calculado pela Equação 2. O TTL é baseado no tempo necessário para um veículo cruzar o diâmetro do alcance de comunicação dos nós (RSU ou veículos) com a sua velocidade máxima registrada (*maxSpeed*). Após esse cálculo, a mensagem é gerada e transmitida no estado *Send Message*.

$$TTL = 2 \times radius / maxSpeed \quad (2)$$

**Função #3: Validade Message** - implementa a Regra #1, que verifica se a mensagem recebida já está armazenada na tabela de mensagens do nó. Os números de identificação de mensagens e as coordenadas de origem são comparados para decidir se é um alerta duplicado ou novo. Na RSU, a transição para este estado pode vir de *Check Status*, que analisa a mensagem recebida, adiciona ou a atualiza, e retorna para *Check Status*. Caso a transição parta de *Schedule Transmission*, a função processa a mensagem recebida e retorna ao estado *Schedule Transmission* para terminar a espera do intervalo e prosseguir com a transmissão da mensagem. No veículo a mensagem recebida é adicionada caso seja uma mensagem nova, ou atualizada, caso seja mensagem de um acidente conhecido.

---

#### Algorithm 1 Validade Message

---

```

1: function VALIDADEMESSAGE
2:   if idMsg! = message.id then
3:     calculateDistance
4:     if distance >= SafeDistance then
5:       addMsg
6:     else
7:       updateTable
8:   else
9:     updateTable

```

---

A comparação de mensagens é descrita no Algoritmo 1. A linha 2 verifica se a mensagem recebida já existe na tabela, e se existir, a linha 3 calcula a distância entre o nó de origem da mensagem recebida e o nó de origem da mensagem na tabela. Se a distância for maior que a distância segura (linha 4), a mensagem é considerada um novo alerta e é adicionada à tabela de mensagens (linha 5). Se as condições das linhas 2 e 4 forem falsas, a mensagem recebida será considerada uma duplicata e sua entrada será atualizada na tabela (linhas 7 e 9). Nos veículos, a função *updateTable* incrementa o valor de *backoff* e atualiza o tipo de remetente. Nas RSUs, apenas o *timeReceive* é atualizado.

**Função #4: Verify TTL** - implementa a Regra #2 e verifica o tempo de vida das mensagens (TTL) armazenadas na tabela e que estão sendo preparadas para retransmissão. O objetivo é evitar a retransmissão de mensagens com TTL expirado ou reenvio de alertas de eventos antigos. Se o TTL da mensagem for maior que 0, a máquina de estados do veículo segue para o estado *Verify Sender*, e nas RSU a máquina segue para o estado *Schedule Transmission*. Caso o TTL da mensagem seja 0, ela é removida da tabela.

**Função #5: Verify Sender** - é executada somente em veículos e implementa a Regra #3, que verifica o último remetente da mensagem que está sendo preparada para retransmissão. Quando um veículo recebe de uma RSU alguma mensagem que já esteja em sua tabela, a entrada desta mensagem é atualizada com o último remetente. Isso impede a replicação da mensagem pelo veículo enquanto ele estiver na área de cobertura da RSU.

---

**Algorithm 2** Verify Sender

---

```
1: function VERIFYSENDER
2:   if message.type = vehicle then
3:     ScheduleTransmission
4:   if message.type = RSU and currentTime > message.timeReceive + timeMsg then
5:     message.type = vehicle
6:     ScheduleTransmission
```

---

A função *Verify Sender* é descrita no Algoritmo 2. Na linha 2, é verificado se o remetente da mensagem é um veículo e, se for verdade, a máquina prossegue para o estado *Schedule Transmission*. Na linha 4 é verificado se o veículo perdeu o contato com a RSU, analisando se o tempo decorrido desde a recepção da última mensagem é maior que o intervalo de transmissão padrão, (*timeMsg*). Nesse caso, o veículo considera que saiu da área de cobertura da RSU, e retoma a retransmissão da mensagem. Na linha 5, o tipo de remetente da mensagem é atualizado e a máquina entra no estado *Schedule Transmission*.

**Função #6: Schedule Transmission** - escalona as transmissões de mensagens nos veículos e implementa as Regras #4 e #5, para que as retransmissões sejam baseadas no *backoff* da mensagem e na distância (ambos valores obtidos da tabela de mensagens).

A implementação da Regra #4 calcula o intervalo de retransmissão de mensagens baseado no valor de *backoff*. Toda vez que um veículo recebe uma mensagem duplicada, o valor de *backoff* é incrementado, sendo decrementado quando o veículo realiza seus reenvios com sucesso. Como o intervalo calculado é proporcional ao número de duplicatas recebidas, o protocolo prioriza a retransmissão por nós que receberam menos mensagens repetidas. Para implementar a Regra #5, é calculada a distância entre o remetente e o destinatário da mensagem. Os veículos distantes do remetente têm intervalos de retransmissão curtos e os veículos próximos têm os intervalos de retransmissão aumentados exponencialmente.

---

**Algorithm 3** Calculate Time

---

```
1: function CALCULATETIME
2:   timeBackoff = timeMsg * message.backoff
3:   distance = calculateDistance
4:   interval = timeBackoff * (radius/distance)
5:   CreateMsg
```

---

O Algoritmo 3 descreve o cálculo do intervalo de transmissão de uma mensagem (linhas 2 a 4) e cria a mensagem (linha 5) com informações obtidas da tabela e com o intervalo de retransmissão calculado. As regras #4 e #5 são válidas apenas para veículos. As RSUs

**Tabela 1. Parâmetros de Simulação.**

<b>Protocolos</b>	802.11p/WAVE
<b>Simuladores</b>	OMNeT++, SUMO, Veins
<b>Traces</b>	Grid e Cologne
<b>Tempo de Acidente</b>	50s
<b>Número de Acidentes</b>	2 veículos
<b>Momento do Acidente</b>	65s (Grid) e 600s (Cologne)
<b>Área de Cobertura</b>	250m
<b>Protocolos Comparados</b>	AID, DBRS, Flooding

tem prioridade na disseminação de alertas e seu intervalo de retransmissão é calculado pela Equação 1. Os veículos não retransmitem dentro da área da RSU, portanto não incrementam o número de retransmissões desnecessárias.

Os nós aguardam o tempo limite do intervalo no estado *Schedule Transmission* antes de efetivamente enviar a mensagem criada. Se durante este período de espera for recebida uma mensagem, o veículo e a RSU terão estados distintos para processar esta mensagem. O veículo prossegue para um estado *Verify Message* para processar a mensagem recebida. As RSUs apenas adicionam ou atualizam a tabela de mensagens com a mensagem recebida no estado *Validade Message* e, em seguida, retornam para aguardar o tempo limite do intervalo e enviar a mensagem programada.

**Função #7: Send Message** - configura o rádio e transmite as mensagens agendadas. Após a transmissão da mensagem, a máquina de estado retorna ao estado *Check Status*.

**Função #8: Verify Message** - presente apenas nos veículos, processa uma mensagem recebida no intervalo de tempo de espera para retransmissão em *Schedule Transmission*. Se a mensagem recebida for um novo alerta, ela é adicionada à tabela de mensagens. Se a mensagem recebida se referir ao mesmo acidente da mensagem agendada para transmissão, a transmissão agendada é cancelada, seguindo para o estado *Cancel transmission*.

**Função #9: Cancel Transmission** - presente apenas nos veículos, é utilizada para cancelar uma transmissão de mensagem programada, retornando ao estado *Check Status*.

#### **4. Metodologia**

Nos experimentos foram utilizados dois cenários distintos e com dois veículos acidentados em um mesmo acidente. A quantidade de veículos foi variada em experimentos com 33 repetições, sendo que os resultados finais são a médias de repetições com intervalo de confiança de 95%. O primeiro cenário é um Grid, composto por blocos 4x4 com lados de 250m cada. As rotas dos veículos foram geradas aleatoriamente e a quantidade de veículos foi variada em 25, 50, 100, 150 e 200. O acidente ocorre 65 segundos após o início da simulação e tem 50 segundos de duração. A velocidade máxima dos veículos é de 50 km/h e quatro RSUs são colocadas para fornecer cobertura no mapa.

O segundo cenário utiliza o mapa da cidade de *Cologne*, com dados reais de fluxo e rotas, obtidos a partir da movimentação de 5.000 táxis. Para as simulações foram implementados acidentes, 600 segundos após o início da simulação, em três locais distintos de de uma autopista de alta velocidade, que correspondem a regiões com quantidade baixa, média e alta de veículos (estimação empírica). São utilizadas quatro RSUs, alocadas nas proximidades do acidente, para interação com os veículos da região. Em todas as simulações, o alcance do rádio foi definido como 250 m com base nos resultados empíricos de [Barcelos et al. 2014]. As simulações foram realizadas no OMNeT++

versão 5.1.1, no framework Veins versão 4.6 e os eventos relacionados à mobilidade implementados no SUMO, versão 0.30. A Tabela 1 mostra os parâmetros de simulação.

Para avaliar o RODOGE e compará-lo com outras soluções, são utilizadas as seguintes métricas: número de mensagens retransmitidas, número de mensagens recebidas, distância de recebimento, número de saltos de uma mensagem, porcentagem de remetentes e porcentagem da área de cobertura.

## 5. Resultados e Discussões

Neste artigo foi simulada a disseminação de mensagens de alerta de acidente por dois veículos envolvidos em um mesmo evento. Foram utilizados dois cenários, uma Grid e um *trace* de *Cologne* para comparar o RODOGE aos protocolos clássicos de disseminação de mensagens Flooding, AID e DBRS, que possuem código disponível ou passíveis de implementação. Para melhor visualização, os resultados do protocolo Flooding foram omitidos nos resultados da quantidade de mensagens geradas e do número de mensagens recebidas, já que seus valores estavam longe dos outros protocolos.

### 5.1. Cenário Grid

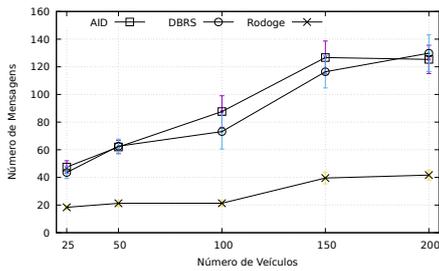
A Figura 3 apresenta os resultados da média de mensagens geradas por cada protocolo, pela variação da quantidade de veículos (25, 50, 100, 150 e 200). O RODOGE obteve melhores resultados que outros protocolos em todos os casos, gerando quase um terço das mensagens geradas pelos demais protocolos, mantendo a mesma cobertura de disseminação de alertas. A redução do número de mensagens geradas pelo RODOGE deve-se às regras de reenvio de mensagens, como a regra de verificação do remetente, que adia retransmissões de veículos quando estão em áreas cobertas por uma RSU, e a comparação de mensagens para identificar mensagens duplicadas, que faz com que o protocolo aumente seu intervalo de transmissão.

Os resultados mostrados na Figura 4 são da quantidade média de mensagens geradas por nó, calculadas com base no número total de mensagens geradas pelo número de veículos e RSUs que geraram mensagens nas simulações. Em todos os cenários, variando o número de veículos, o protocolo RODOGE gerou menor número de mensagens por nó, essa diferença é por causa das regras de limitação de retransmissão do RODOGE.

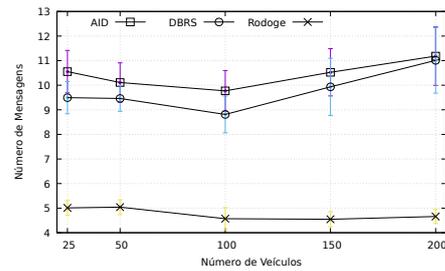
Os resultados da média das mensagens recebidas por execução (ou redundância de mensagens) são mostrados na Figura 5. O RODOGE obteve os melhores resultados, com menos de 1.000 mensagens em todos os cenários. A maior quantidade de mensagens recebidas pelos AID e DBRS é consequência da maior quantidade de mensagens geradas, mostrado na Figura 3. Os controles de retransmissão do RODOGE reduzem a quantidade de mensagens geradas, consequentemente reduzem a quantidade de mensagens recebidas.

A Figura 6 apresenta os resultados da quantidade média de mensagens recebidas por nó, calculada pelo número de veículos e RSUs que receberam mensagens. Em todas as variações do número de veículos, o RODOGE recebeu um número médio de mensagens por nó menor que os demais protocolos. O RODOGE gera menos mensagens devido à restrição da propagação de mensagens por veículos distantes da região do acidente.

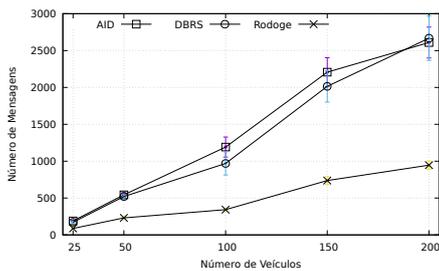
A Figura 7 apresenta a distância média de recebimento dos veículos, ou a distância de propagação da mensagem. Neste cenário, a velocidade máxima dos veículos foi de 50 km/h, e a *SafeDistance* (Equação 1) foi de 30,6m. Os resultados obtidos para todos os



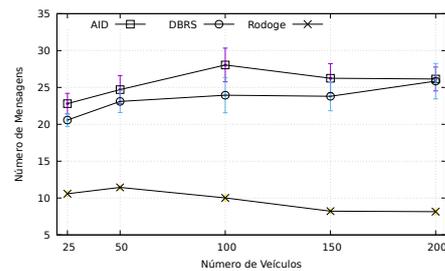
**Figura 3. Média de mensagens geradas por execução.**



**Figura 4. Média de mensagens geradas por nó.**



**Figura 5. Média de mensagens recebidas por execução.**



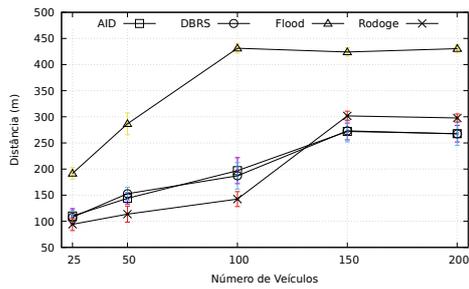
**Figura 6. Média de mensagens recebidas por nó.**

protocolos, considerando a variação do número de veículos, foram superiores à distância segura. O protocolo Flooding obteve melhores resultados, apresentando distâncias acima da distância segura, devido à grande quantidade de mensagens enviadas. Para 25 veículos, AID, DBRS e RODOGE obtiveram resultados estatisticamente semelhantes, enquanto para 50 e 100 veículos, AID e DBRS foram superiores ao RODOGE. Para 100 e 200 veículos, o RODOGE aumentou a distância de recebimento, alcançando resultados estatisticamente equivalentes ao DBRS e ao AID. Apesar do menor número de mensagens geradas pelo RODOGE, a distância de recebimento foi sempre acima da *SafeDistance*.

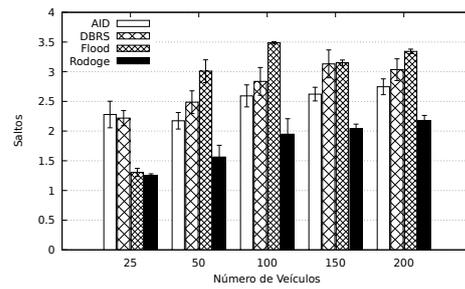
Para verificar a entrega de mensagens de alerta para veículos com o menor número de intermediários, o número médio de saltos é mostrado na Figura 8. No cenário com 25 veículos, os protocolos Flooding e RODOGE obtiveram resultados estatisticamente semelhantes e melhores em comparação a outros protocolos. Para os demais cenários, de 50 a 200 veículos, o RODOGE obteve o menor número de saltos das mensagens, isso se deve à regra que prioriza as retransmissões pela RSU.

Na Figura 9 é mostrada a porcentagem de nós que retransmitiram a mensagem de alerta. Em todas as simulações, o Flooding obteve a maior porcentagem de remetentes, já que não há controles de retransmissão. Os resultados dos protocolos AID e DBRS foram estatisticamente semelhantes. O RODOGE obteve os melhores resultados, justificado pela aplicação da Regra #3, que limita o encaminhamento de mensagens por veículos dentro da área de cobertura da RSU. Além disso, as Regras #4 e #5 impedem que um veículo retransmita uma mensagem se na sua região houver outros veículos replicando a mensagem, e se estiver próximo do remetente.

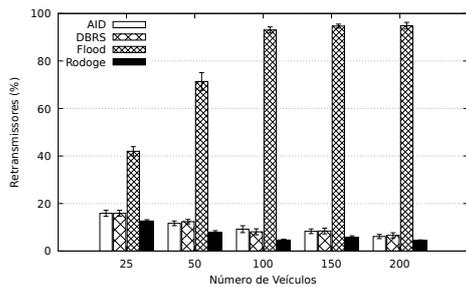
Uma das métricas mais importantes é a porcentagem de veículos que recebem a mensagem de alerta de acidente em distância segura. É importante observar que nem todos os veículos da rede devem receber alertas. A Figura 10 mostra os resultados, o Flooding obteve os maiores percentuais, devido ao grande número de mensagens que o permite



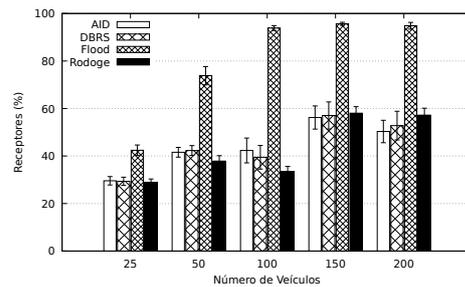
**Figura 7. Distância média de recebimento.**



**Figura 8. Média de saltos da mensagem.**



**Figura 9. Porcentagem média de retransmissões.**



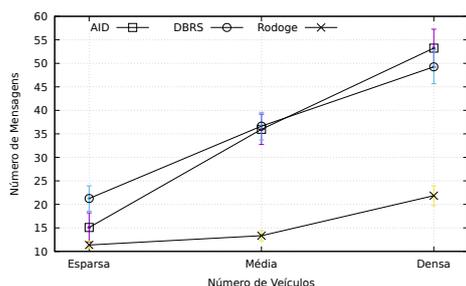
**Figura 10. Média da área de cobertura.**

alcançar uma maior quantidade de veículos. RODOGE, AID e DBRS obtiveram resultados estatisticamente semelhantes e menores que o Flooding, devido aos seus controles de reenvio de mensagens. Além de reduzir o número de mensagens retransmitidas, o RODOGE mantém a porcentagem de veículos que recebem a mensagem de alerta com uma distância segura semelhante aos outros protocolos que geram mais mensagens.

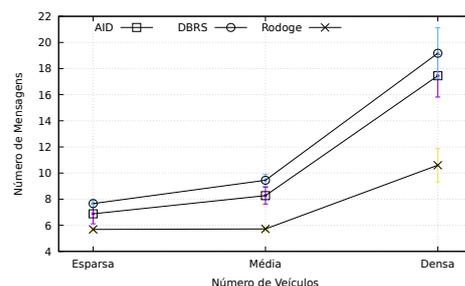
## 5.2. Cenário Cologne

A Figura 11 apresenta os resultados da média de mensagens geradas por cada protocolo, considerando a quantidade de veículos na rede (Esparsa, Média e Densa). Na rede Esparsa, os protocolos DBRS e RODOGE obtiveram resultados similares e melhores, ligeira vantagem para o RODOGE. Nas demais variações o RODOGE obteve os melhores resultados, gerando cerca da metade da quantidade de mensagens geradas pelos outros protocolos. Essa redução está relacionada às regras para reenvio de mensagens na área da RSU, à regra que compara as mensagens (para identificar os dois acidentes deste cenário) e também às regras que incrementam o intervalo de transmissão da mensagem de acordo com sua densidade e distância do remetente. A média de mensagens geradas é calculada com base na quantidade de veículos e RSUs que geraram mensagens. A Figura 12, mostra que o protocolo RODOGE obteve os melhores resultados em todas as variações, gerando uma quantidade menor de mensagens por nó, justificado pelas regras que limitam o reenvio.

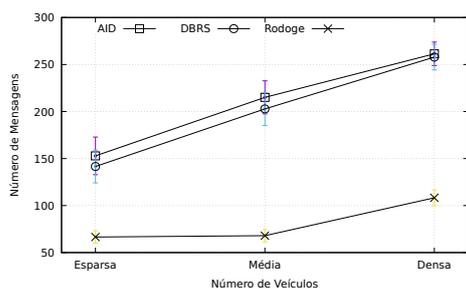
Os resultados da média das mensagens recebidas por execução (redundância de mensagens) para a variação da quantidade de veículos (Esparsa, Média e Densa), são mostrados na Figura 13. O RODOGE obteve os melhores resultados, com média inferior a 120 mensagens recebidas em todos os cenários. Os protocolos AID e DBRS aumentaram a quantidade de mensagens recebidas, justificado pela maior quantidade de mensagens geradas. Os controles de reenvio do RODOGE reduzem a quantidade de mensagens geradas, e com isso, a quantidade de mensagens recebidas, como mostrado na Figura 14.



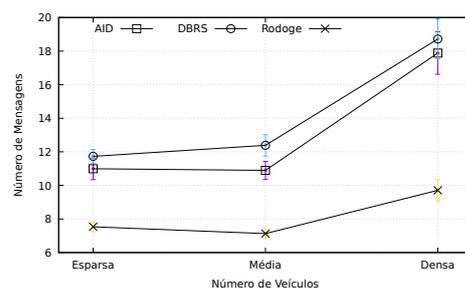
**Figura 11. Média de mensagens geradas por execução.**



**Figura 12. Média de mensagens geradas por nó.**



**Figura 13. Média de mensagens recebidas por execução.**



**Figura 14. Média de mensagens recebidas por nó.**

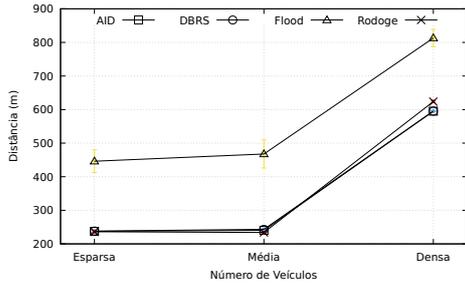
No cenário de *Cologne*, foi considerada a velocidade máxima 110 km/h e uma distância segura de 91,6 m. A Figura 15 apresenta a distância média de recebimento dos veículos, ou a distância de propagação da mensagem. Os resultados de todos os protocolos, em todas as variações de veículos, foram superiores à distância segura. O Flooding superou todos os outros protocolos, com distâncias de recebimento maiores devido à grande quantidade de mensagens reenviadas. Apesar do menor número de mensagens geradas pelo RODOGE, a distância de recebimento foi sempre acima da distância de segurança.

Para verificar a entrega de mensagens de alerta para veículos com o menor número de intermediários, o número médio de saltos é mostrado na Figura 16. O protocolo Flooding obteve os piores resultados em todas as variações de veículos com suas mensagens possuindo o maior valor médio de número de saltos. Em todos os cenários, os protocolos AID, DBRS e RODOGE obtiveram resultados estatisticamente similares, com o RODOGE possuindo um valor levemente menor. O RODOGE manteve uma constância nos resultados, justificada pela centralização de reenvio pela RSU ao seguir a regra de verificação do remetente da mensagem.

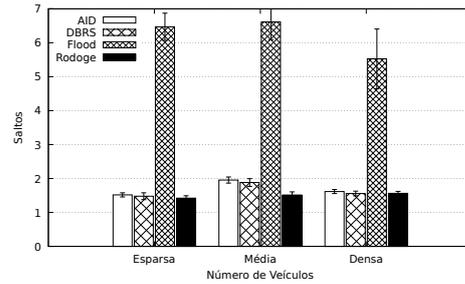
A média de remetentes de mensagens nas simulações pode ser observada na Figura 17. Em todas as variações de veículos o protocolo Flooding obteve a maior porcentagem de remetentes, pois não há controles de reenvio de mensagens. Os resultados dos protocolos AID e DBRS foram estatisticamente similares em todas as variações de veículos. O protocolo RODOGE obteve os melhores resultados em todos os casos, com uma média de remetentes sempre menor que os demais protocolos. Tais resultados estão relacionadas as regras de reenvios utilizadas pelo RODOGE, como a Regra #3 que limita o reenvio de mensagens pelos veículos dentro da área de cobertura da RSU e as Regras #4 e #5 que alternam e aumentam os tempos de retransmissão.

A Figura 18 apresenta os resultados da média de veículos cobertos pela mensagem

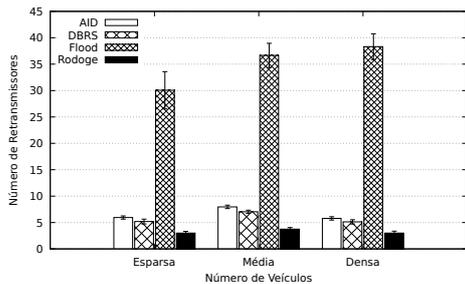
de alerta de acidente. Em todas as simulações e variações na quantidade de veículos, o protocolo Flooding obteve maiores médias com um maior número de veículos cobertos pela mensagem, devido ao grande número de mensagens que reenvia e assim alcança nós mais rapidamente. Em todos os cenários, os protocolos AID e DBRS obtiveram resultados estatisticamente similares. Apesar do RODOGE reduzir a quantidade de mensagens reenviadas, a quantidade de veículos que recebem a mensagem de alerta em distância segura é ligeiramente inferior à dos protocolos que reenviam mais mensagens.



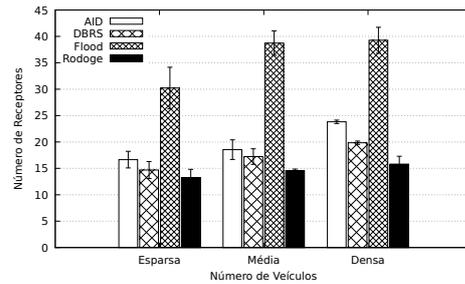
**Figura 15. Distância média de recebimento.**



**Figura 16. Média de saltos da mensagem.**



**Figura 17. Média de remetentes.**



**Figura 18. Média da área de cobertura.**

## 6. Conclusões e Trabalhos Futuros

O protocolo RODOGE reduziu o número de mensagens retransmitidas devido aos controles impostos pelas cinco regras implementadas e ainda manteve a área de cobertura de alerta de acidentes e a distância de recebimento maiores que a distância segura. Os resultados mostraram que o RODOGE retransmite menos mensagens quando comparado aos protocolos clássicos AID, DBRS e Flooding, especialmente em cenários mais densos. Além disso, mesmo reduzindo a quantidade de retransmissões, o RODOGE é capaz de fornecer uma área de cobertura equivalente aos outros protocolos. Como trabalhos futuros, pretendemos realizar outras simulações considerando a interligação de vários dispositivos de infraestrutura, executar simulações com um maior número de acidentes e utilizar outros *traces* reais de mobilidade de veículos.

## Referências

- Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., and Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. *Journal of Network and Computer Applications*, 37:380 – 392.
- Bakhouya, M., Gaber, J., and Lorenz, P. (2011). An adaptive approach for information dissemination in vehicular ad hoc networks. *Journal of Network and Computer Applications*, 34(6):1971 – 1978. Control and Optimization over Wireless Networks.

- Barcelos, V. P., Amarante, T. C., Drury, C. D., and Correia, L. H. A. (2014). Vehicle monitoring system using ieee 802.11p device and android application. In *2014 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7.
- Chen, Y., Shen, K., and Wang, S. (2013). Forward collision warning system considering both time-to-collision and safety braking distance. In *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, pages 972–977.
- Feukeu, E. A. and Zuva, T. (2017). Overcoming broadcast storm problem in a vehicular network. In *2017 13th SITIS*, pages 402–407.
- Group, I. . W. (2014). IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. *IEEE Std 1609.0-2013*, pages 1–78.
- Kim, T.-H., Hong, W.-K., Kim, H.-C., and Lee, Y.-D. (2008). An effective data dissemination in vehicular ad-hoc network. In *Information Networking. Towards Ubiquitous Networking and Services*, pages 295–304. Springer Berlin Heidelberg.
- Meneguette, R. I., Maia, G., Madeira, E. R. M., Loureiro, A. A. F., and Villas, L. A. (2014). Autonomic data dissemination in highway vehicular ad hoc networks with diverse traffic conditions. In *2014 IEEE ISCC*, pages 1–6.
- Neto, J. B. P., Gomes, L. C., Castanho, E. M., Campista, M. E. M., Costa, L. H. M. K., and Ribeiro, P. C. M. (2016). An error correction algorithm for forward collision warning applications. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1926–1931.
- Pögel, T. and Wolf, L. (2012). Prediction of 3g network characteristics for adaptive vehicular connectivity maps (poster). In *2012 IEEE VNC*, pages 121–128.
- Qi, W., Song, Q., Wang, X., Guo, L., and Ning, Z. (2018). Sdn-enabled social-aware clustering in 5g-vanet systems. *IEEE Access*, 6:28213–28224.
- Souza, A. M., Maia, G., and Villas, L. A. (2014). Add: A data dissemination solution for highly dynamic highway environments. In *2014 IEEE 13th International Symposium on Network Computing and Applications*, pages 17–23.
- Tehrani, M., Uysal, M., and Yanikomeroglu, H. (2014). Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions. *IEEE Communications Magazine*, 52(5):86–92.
- Tseng, Y.-C., Ni, S.-Y., and Shih, E.-Y. (2001). Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. In *Proceedings 21st International Conference on Distributed Computing Systems*, pages 481–488.
- Villas, L. A., Ramos, H. S., Boukerche, A., Guidoni, D. L., Araujo, R. B., and Loureiro, A. A. (2012). 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*, pages 39–46, NY, USA. ACM.
- WHO (2018). World Health Organization - Road Traffic Injuries. <http://www.who.int/en/news-room/fact-sheets/detail/road-traffic-injuries>. Acessado em 10/2018.
- Xeros, A., Lestas, M., Andreou, M., and Pitsillides, A. (2010). Adaptive probabilistic flooding for information hovering in vanets. In *2010 IEEE Vehicular Networking Conference*, pages 239–246.