

# Uma Solução Baseada em Infraestrutura Auxiliar para o Problema de Gerenciamento de Tráfego em VANETs

Massilon Lourenço, Thiago Gomides, Pedro H. Souza,  
Cristiano M. Silva e Daniel L. Guidoni

<sup>1</sup> Departamento de Computação – Universidade Federal de São João del-Rei (UFSJ)

{massilon, gomides, pedronhds, cristiano, guidoni}@ufs.br

**Abstract.** Expenses due to congestion problems in large urban centers reach billions of dollars worldwide. This is due to the time lost in traffic and fuel consumption caused mainly by traffic jams at peak hours. Several papers in the literature propose solutions to the problem of traffic management using the capacity of processing, storage and communication of the vehicular networks. Among the literature solutions, the infrastructured approach use the processing, storage and communication capabilities power of the infrastructure for the detection of traffic jams and suggestion of routes for the vehicles. This paper presents the GRIFO that, unlike the infrastructured approaches of the literature, the vehicles are responsible for checking the congestion and calculating new routes when necessary. Using only information of the vehicle's nearby roads provided by the auxiliary storage infrastructure, each vehicle checks the need to calculate a new route. The proposed work can distribute the flow of vehicles in the road network with the objective of reducing the average travel time compared to literature solutions.

**Resumo.** Os gastos devido aos problemas de congestionamentos em grandes centros urbanos chegam a bilhões de dólares em todo mundo. Isso se deve ao tempo perdido no trânsito e ao consumo de combustível causado, principalmente, pelos engarrafamentos em horários de pico. Vários trabalhos na literatura propõem soluções para o problema de gerenciamento de tráfego utilizando a capacidade de processamento, armazenamento e comunicação de redes veiculares. Dentre as soluções presentes na literatura, as abordagens infraestruturadas utilizam o poder de processamento e armazenamento de unidades de comunicação distribuídas ao longo da malha viária para a detecção de engarrafamentos e sugestão de rotas para os veículos. Este trabalho apresenta o GRIFO que, diferente das abordagens infraestruturadas da literatura, propõe que os veículos sejam responsáveis por verificar o congestionamento e calcular novas rotas, quando necessário. Utilizando apenas informações sobre as condições das vias próximas ao veículo providas pela infraestrutura auxiliar de comunicação, cada veículo verifica a necessidade de cálculo de nova rota. O trabalho proposto consegue distribuir o fluxo de veículos na malha viária com o objetivo de reduzir o tempo médio de viagem em comparação com algoritmos da literatura.

## 1. Introdução

O problema de congestionamento em grandes centros urbanos vem crescendo anualmente [Njus 2018, Ramalho 2018]. Dentre as 50 cidades mais congestionadas do mundo,

os EUA possuem 17 [Cookson 2018]. O exemplo de uma típica cidade com problemas de congestionamentos é Portland, que ocupa a posição de 12<sup>a</sup> cidade mais congestionada dos Estados Unidos e 28<sup>a</sup> do mundo [Pittman 2018]. Segundo um estudo realizado em 2017 [Cookson 2018], o tempo médio anual perdido em congestionamentos por motoristas na cidade de Portland cresceu de 47h para 50h. Além disso, o custo médio causado por engarrafamentos por motorista foi de 1.648,00 dólares ao ano e o custo anual total de congestionamentos chega a mais de 3.9 bilhões de dólares [Pittman 2018].

Com base no relatório do Ministério dos Transportes dos EUA [Inc 2005], congestionamentos são causados principalmente por três fatores: (i) eventos que influenciam o tráfego, como áreas de trabalho (concentração de escritórios), meteorologia, acidentes etc; (ii) demandas de tráfego, como eventos especiais (festas, shows, feriados) e flutuações esperadas do fluxo de veículos (horas de pico); e (iii) características físicas das malhas viárias, que abordam a quantidade e capacidades das vias, viadutos, sinais de trânsito, etc. A urbanização das cidades aumenta as áreas de trabalho bem como a demanda por mobilidade. Acidentes de trânsito e eventos meteorológicos podem ser imprevisíveis e afetam, instantaneamente, as condições de tráfego. Alterações na infraestrutura física da malha viária demandam recursos financeiros e tempo. Neste cenário, soluções tecnológicas, também conhecidas como Sistemas de Gerenciamento de Tráfego (SGT), apresentam como alternativa para a redução de congestionamentos em ambientes urbanos.

Um sistema de gerenciamento de tráfego urbano integra dados dos usuários e da malha viária com as tecnologias de processamento e comunicação com o objetivo de reduzir congestionamentos pela alteração das rotas dos veículos durante o seu percurso. Uma possível implementação desse sistema pode se dar através da disponibilização de comunicação sem fio nos veículos, formando uma rede *ad hoc* enquanto se movimentam. Esse tipo de rede é conhecido como VANET – *Vehicular Ad-hoc NETWORK* [Atallah et al. 2017]. Nas VANETs, os veículos podem se comunicar através da comunicação veículo-para-veículo (V2V) ou via infraestrutura auxiliar fixa (dispositivos depositados ao longo das vias conhecidos como RSUs – *Road-Side Units*) utilizando a comunicação V2I (veículo-para-infraestrutura) [Silva et al. 2017].

Vários trabalhos na literatura abordam o serviço de controle de congestionamento utilizando redes veiculares [Brennand et al. 2016, Pan et al. 2017, Li et al. 2018]. As soluções da literatura são frequentemente classificadas em centralizadas, totalmente distribuídas ou infraestruturadas. Soluções centralizadas utilizam servidores centralizados que recebem e armazenam informações sobre as vias da malha viária considerada. O servidor também possui o papel de detectar e sugerir rotas alternativas para os veículos com o objetivo de reduzir/mitigar congestionamentos. Soluções centralizadas podem utilizar algoritmos sofisticados para realizar o controle de congestionamento sem mover o congestionamento para outro ponto da rede, uma vez que o servidor possui uma visão global das vias e suas características. Entretanto, essas soluções demandam grande poder computacional do servidor e acesso contínuo à Internet pelos veículos [Pan et al. 2017], o que nem sempre é possível.

Por outro lado, soluções totalmente distribuídas utilizam somente a comunicação V2V entre veículos para descobrir as condições de tráfego em suas proximidades, o que pode demandar grande quantidade de mensagens ou mal funcionamento do sistema devido a desconexões [Atallah et al. 2017]. As soluções infraestruturadas utilizam a capaci-

dade de computação das RSUs para descentralizar o gerenciamento de tráfego e deixá-lo próximo aos veículos. Neste modelo, cada RSU é responsável por gerenciar o fluxo de veículos em uma determinada região do conjunto total de vias da malha viária considerada [Brennand et al. 2016].

Este trabalho propõem um sistema gerenciador de tráfego utilizando uma rede veicular infraestruturada chamado de GRIFO (GeRencIador de tráFego baseado em di-agrama vOronoi). O sistema proposto deposita um conjunto de RSUs nas vias e aplica o conceito de diagrama de voronoi, onde o centro de cada célula voronoi é a posição de uma RSU e o tamanho da célula define a região de responsabilidade da RSU em questão. Ao contrário dos trabalhos da literatura onde a RSU mantém as informações e calcula as rotas dos veículos, a função das RSUs no GRIFO é apenas de armazenar as informações sobre as vias. Durante o seu percurso, cada veículo monitora o tempo necessário para percorrer as vias. Sempre que o veículo inicia o percurso de uma nova via, ele envia uma mensagem para a RSU responsável pela sua região informando o tempo de percurso da via anterior. Quando necessário, o veículo requisita para a RSU responsável pela sua via atual a informação das vias da sua região. De posse das informações sobre o tempo de percurso das vias da sua região, o veículo verifica se existe um caminho mais adequado.

A solução proposta foi comparada com soluções da literatura utilizando a malha viária do centro da cidade de Portland, apresentando resultados promissores considerando quantidade de mensagens transmitidas e métricas de congestionamento (tempo de viagem, distância percorrida, velocidade média e tempo gasto no congestionamento).

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A seção 3 descreve a nova abordagem para o problema de controle de congestionamento. As seções 4 e 5 descrevem a metodologia de avaliação e os resultados de simulação. A seção 6 conclui o trabalho.

## 2. Trabalhos Relacionados

Dentre os algoritmos de controle de congestionamento que utilizam processamento centralizado, os mais relevantes para o trabalho proposto são DSP (*Dynamic Shortest Path*), RkSP (*Random k Shortest Paths*) e CO-OP (Uma Solução para a Detecção, Classificação e Minimização de Congestionamentos de Veículos utilizando Roteamento Cooperativo). DSP foi proposto em [Pan et al. 2017] e utiliza uma abordagem preditiva onde, periodicamente, as rotas dos veículos são alteradas com o objetivo de reduzir congestionamentos. Os veículos notificam o servidor central sobre o tempo de viagem das vias percorridas pelo veículo utilizando uma conexão com a Internet. Com base na informação global das vias, o DSP aplica um algoritmo de caminho mínimo (ponderamento das arestas com o tempo de viagem) para encontrar novas rotas. A desvantagem dessa solução é a alteração direta das rotas pelo caminho mais rápido, podendo ocasionar novos engarrafamentos em outras regiões. Para solucionar esse problema, os autores propõem o RkSP, onde ao invés de calcular apenas a rota mais rápida, o algoritmo calcula  $k$  rotas e aleatoriamente escolhe uma para o veículo. Os autores mostram que o RkSP possui resultados  $\approx 13\%$  menores em comparação com o DSP quando considera-se o tempo médio de viagem dos veículos.

O CO-OP [Souza et al. 2016] foi proposto como uma solução para prevenção e controle de congestionamentos utilizando roteamento cooperativo. Os autores propõem uma classificação do tráfego de veículos em uma determinada via em diferentes níveis

utilizando o *Highway Capacity Manual (HCM)* [Board]. O CO-OP utiliza o algoritmo de aprendizado de máquina *k*-NN (*K-Nearest Neighbor*) para classificar os níveis de congestionamentos em todos os cenários. Os veículos notificam o servidor central periodicamente sobre a sua posição e velocidade, e o cálculo da nova rota considera a velocidade média, densidade e velocidade máxima das vias. O termo cooperativo se refere à quantidade de veículos na via que é utilizada para evitar futuros congestionamentos. Uma possível desvantagem dessa abordagem é a necessidade de notificação periódica da velocidade dos veículos, o que aumenta a quantidade de mensagens trocadas entre veículos-servidor para o correto funcionamento do sistema.

Considerando as abordagens que utilizam apenas a comunicação V2V, podemos destacar as seguintes. Em [Doolan and Muntean 2013], os autores propõem o EcoTrec. O algoritmo utiliza informações de engarrafamento para reduzir o consumo de combustível dos veículos. Nessa abordagem, os veículos disseminam informações do seu trajeto e consumo de combustível. De posse dessas informações, cada veículo cria um mapa da malha viária onde o peso das arestas é o consumo de combustível da via correspondente. O EcoTrec busca novos caminhos que possuem os menores consumos de combustível e, segundo os autores, essa escolha tende a diminuir congestionamentos.

Utilizando também a comunicação V2V, os autores em [Meneguet et al. 2016] propõem o INCIDENT (INtelligent protocol of CongestIon DETection). O objetivo principal do INCIDENT é reduzir o tempo médio de viagem dos veículos. Possui, como núcleo, a criação de uma rede neural artificial para detectar congestionamentos. Veículos disseminam periodicamente informação sobre sua velocidade média e localização na via atual. Com base nessas características, é criada uma rede neural para verificar se uma determinada via com uma determinada velocidade média e quantidade de veículos está congestionada. A rede neural possui como saída um peso de 0 a 1, onde 1 representa uma via totalmente engarrafada. Cada veículo cria um mapa das vias e utiliza a saída da rede neural como pesos nas arestas. Ao detectar um congestionamento, o veículo calcula uma nova rota com o menor peso aplicando um algoritmo de caminho com custo mínimo.

Em [Brennan et al. 2016] é proposto o FOX – *Fast Offset XPath*, uma abordagem infraestruturada para o problema de gerenciamento de congestionamento em redes veiculares. Assim como no trabalho [Pan et al. 2017], o FOX também utiliza alterações periódicas nas rotas. Emprega o conceito de névoa para descentralizar o processamento e as informações do cenário, sendo ele dividido em sub-regiões onde cada RSU gerencia uma região (névoa). A deposição das RSUs é realizada de modo que todas as regiões sejam de mesmo tamanho e todo o cenário seja coberto por alguma RSU.

Utilizando a comunicação V2V e V2I periodicamente, os veículos informam a sua velocidade média para a RSU. Esta, por sua vez, periodicamente verifica se existe congestionamento dentro de sua região e, em caso positivo, notifica o veículo com uma nova rota. A RSU utiliza o algoritmo *k-shortest paths* com a escolha aleatória seguindo a distribuição de *Boltzmann* para realizar o balanceamento do fluxo. Uma outra forma de balanceamento é a não utilização de rotas muito longas dado um valor de corte pré-definido. Um problema encontrado na criação de novas rotas pelo FOX é que o veículo apenas altera a sua rota dentro da região, sendo que a via de saída da região para uma outra região vizinha deve ser mantida. Diferentemente do GRIFO, onde cada veículo calcula a sua nova rota, o cálculo de novas rotas pelo FOX é realizado pela infraestrutura.

Dessa forma, deve-se calcular  $k$  rotas alternativas para se ter um balanceamento de carga. Além disso, o custo da RSU no GRIFO pode ser reduzido devido ao baixo poder computacional necessário para se manter apenas um banco de dados sobre informações das vias.

Com base nas soluções descritas e suas características, o trabalho proposto tem como objetivo superar os seguintes problemas encontrados nas soluções da literatura: (i) visão global da rede, onde um servidor é responsável por manter todas as informações; (ii) sobrecarga de comunicação para manter uma base de dados sobre as informações de tráfego e (iii) sobrecarga de processamento das soluções centralizadas e infraestruturadas.

### 3. Trabalho Proposto: GRIFO

GRIFO é um sistema gerenciador de tráfego que possui como objetivo melhorar o fluxo de veículos nas vias, reduzindo o congestionamento e minimizando o tempo de viagem. O sistema utiliza uma abordagem infraestruturada, onde RSUs são depositadas na malha viária. Essas RSUs tornam-se centros das células do diagrama de voronoi utilizado para a divisão do cenário em regiões. As RSUs atuam como base de dados onde os veículos enviam mensagens de atualização contendo informações sobre o tempo de viagem das vias ou mensagens de aquisição de informação sobre as vias. De posse das informações sobre as vias da sua região, os veículos calculam novas rotas quando necessário. A seguir apresenta-se o sistema em termos das estratégias usadas para a deposição de RSUs, criação do diagrama de voronoi, comunicação veicular e cálculo de novas rotas pelos veículos.

#### 3.1. Deposição de RSU e Diagrama de Voronoi

As RSUs são depositadas nas vias e a definição da quantidade de RSUs na malha viária é dada pelo projetista do sistema gerenciador de tráfego. Após a deposição das RSUs, é criado um diagrama de voronoi tendo as RSUs como centro das células de voronoi. Utiliza-se o diagrama de voronoi para dividir de maneira balanceada o tamanho de cada região dado uma deposição realizada. A figura 1 (rótulo A) ilustra uma deposição de RSUs (pontos verdes) em um trecho da malha viária da cidade de Portland – USA, bem como a divisão das células de voronoi. Nesse exemplo, 30 RSUs são depositadas de maneira aleatória considerando o cruzamento das vias (junções). Cada célula representa uma região, sendo a RSU o centro dessa região. A RSU mantém uma base de dados com as informações relacionadas ao tempo de travessia das vias da região. A base de dados contém tuplas  $\langle IdentificadorVia, TempoTravessia \rangle$ .

Durante seu percurso, cada veículo monitora o tempo de viagem das vias percorridas. Quando um veículo inicia o percurso da via com identificador  $Via_i$ , ele verifica o tempo de percurso  $t_{i-1}$  da  $Via_{i-1}$ , cria a tupla  $\langle Via_{i-1}, t_{i-1} \rangle$  e notifica a RSU responsável pela via em questão. Quando a RSU recebe a mensagem de notificação, ela sobrescreve o tempo de percurso da via  $Via_{i-1}$  em sua base de dados, mantendo apenas a última informação recebida. Foi observado que a utilização do último tempo de travessia verificado por um veículo possui um comportamento melhor em relação a utilização de uma janela de tempo de travessia (considerando os últimos valores). Dependendo do tamanho da janela de tempo, o sistema não é capaz de verificar rápidas variações ocasionadas pelas modificações das rotas dos veículos.

O projetista da rede deve fazer um estudo relacionando o custo/benefício da instalação e utilização das RSUs. Um pequeno número de RSUs leva a um pequeno número de células, porém com um tamanho individual de cada célula maior. Muitas RSUs proporcionam um grande número de células, porém com um tamanho individual menor. Se uma célula for muito grande, será necessário uma maior utilização da comunicação V2V, uma vez que a maioria dos veículos não conseguirão enviar mensagens direto para a RSU. Se o tamanho da célula for pequeno, a maioria dos veículos tende a utilizar a comunicação V2I diretamente com a RSU. Nesse trabalho utilizamos a deposição aleatória de RSUs na malha viária para considerar ambos os casos descritos, como pode ser observado na figura 1 (rótulo A).

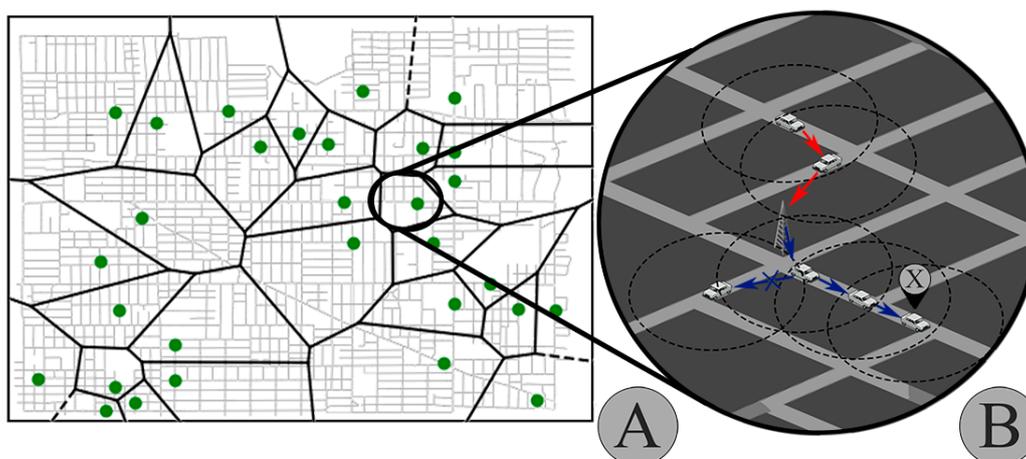


Figura 1. Ilustração de GRIFO

### 3.2. Comunicação entre Nós da Rede

O sistema utiliza comunicação V2V e V2I. A comunicação V2V é utilizada em dois casos: (i) quando o veículo utiliza veículos intermediários para que a mensagem alcance a RSU desejada ou (ii) quando a RSU responde a uma requisição sobre as informações das vias da sua região. O algoritmo 1 formaliza os processos de comunicação V2V e V2I. Quando uma mensagem é criada por um veículo, os seguintes campos são atualizados:  $msg.id = novoID$ , indicando o ID da mensagem;  $msg.posiçãoDestino = posiçãoRSUDestino$ , indicando a posição geográfica da RSU responsável pela via;  $msg.idDestino = idRSUDestino$ , indicando o ID da RSU de destino;  $msg.idOrigem = IDVeículo$ , indicando o id do veículo que está iniciando o processo de comunicação;  $msg.posiçãoOrigem = posiçãoVeículo$ , indicando a posição geográfica do veículo,  $msg.via_{id} = viaID$ , indicando o ID da via,  $msg.duração = duraçãoViaID$ , indicando o tempo de travessia da via em questão e  $msg.tipo = atualização | requisição$ , indicando se uma mensagem é de atualização ou requisição de informação. No caso de uma mensagem de requisição de informação sobre as vias, os campos  $msg.via$  e  $msg.duração$  são descartados pelo destino.

Quando uma mensagem é criada por RSUs, os seguintes campos são atualizados, onde algumas informações são obtidas a partir da mensagem recebida pela RSU:  $msg.id = novoID$ ;  $msg.posiçãoDestino = posiçãoVeículo$ , contendo a posição do veículo que realizou a requisição;  $msg.idDestino = idVeículo$ , contendo o ID do veículo que realizou

a requisição e  $msg.informaçãoVias$ , contendo as informações das vias sob responsabilidade da RSU e  $msg.tipo = resposta$ . Para que a comunicação funcione corretamente, é necessário que os veículos sejam equipados com um receptor GPS, possuam o mapa com a malha viária da cidade (com a localização das RSUs e as células de voronoi) e sejam equipados com uma unidade de processamento, armazenamento e comunicação sem fio.

O procedimento  $V2V(msg)$  no algoritmo 1 (linhas 1–13) ilustra a comunicação V2V entre veículos. Inicialmente, é verificado se a mensagem recebida já foi processada utilizando o seu  $ID$ . Se a mensagem ainda não foi recebida (linhas 2–7), o veículo verifica se ele está mais próximo do destino em relação ao veículo que enviou a mensagem para ele. Se sim, o veículo calcula um tempo de atraso de envio, onde o veículo que está mais próximo do destino possui um atraso de envio menor em comparação com veículos distantes do destino (linha 4). O  $alcanceMáximo$  indica o raio de comunicação e, nesse caso, o atraso no envio é um valor de 0 a 1. Após o cálculo do tempo, a mensagem é escalonada para o envio futuro. O procedimento da linha 7 tem como objetivo atualizar a lista de mensagens recebidas pelo veículo bem como armazenar as informações sobre as vias se a mensagem é do tipo resposta. Nesse caso, o veículo atualiza a sua base de dados local com as informações sobre as vias contidas na mensagem recebida junto com um  $timestamp$ . O objetivo do  $timestamp$  é contabilizar se a mensagem é recente ou não. A próxima seção descreve a utilização do  $timestamp$ .

Se a mensagem já foi recebida pelo veículo (linhas 8–12), o veículo verifica se ele está mais distante da RSU em comparação com o veículo que enviou a mensagem. Se sim, o veículo cancela o seu envio e, dessa forma, a mensagem caminhará no sentido da RSU. O algoritmo 1 (linhas 14–24) ilustra a comunicação entre veículos e infraestrutura (V2I). Se a mensagem é de atualização de tempo de percurso de uma determinada via (Linha 15), a RSU atualiza a sua base de dados (linha 17). Se a mensagem é de requisição de informação (linha 19), a RSU cria uma mensagem de resposta contendo as informações solicitadas e a envia para o veículo. Em ambos os casos é verificado se a mensagem já foi recebida ou enviada pela RSU. Após o envio da mensagem, o encaminhamento entre os veículos é feito utilizando o procedimento  $V2V(msg)$ .

A figura 1 (rótulo B) ilustra a comunicação no processo de atualização e resposta de informações sobre as vias. Seja os veículos de uma determinada célula do diagrama de voronoi e sua respectiva RSU. As setas em vermelho ilustram a comunicação quando um veículo envia uma informação de atualização de uma via que ele acabou de percorrer. A RSU, ilustrada pela torre central, atualiza a sua base de dados. As setas em azul ilustram as mensagens após a RSU receber uma mensagem de requisição pelo veículo  $\otimes$ . Observa-se que apenas os veículos mais próximos ao veículo que requisitou as informações repassam a mensagens. Como mencionado, os veículos que repassam as mensagens de resposta também atualizam a sua base de dados, uma vez que as informações sobre as vias também são de seu interesse.

### 3.3. Alteração de Rotas

Quando o veículo necessita encontrar uma rota alternativa, ele verifica em qual célula se encontra e envia uma mensagem de requisição de informação para a RSU responsável pela célula. Ao receber a resposta e de posse do mapa com a malha viária da cidade, o veículo cria um grafo ponderado  $G = (V, A)$ , onde  $V$  é o conjunto de vértices (cruzamentos)

---

**Algoritmo 1: Comunicação entre os nós**

---

```
1 Procedimento V2V(msg)
2   se !msgProcessada(msg.id) então
3     se distância(minhaPosição) < distância(msg.posição) então
4       atraso ←  $\frac{\text{diferença}(\text{minhaPosição}, \text{msg.posição})}{\text{alcanceMáximo}}$ ;
5       envia(msg, atraso);
6     fim
7     processaMensagem(msg);
8   senão
9     se distância(minhaPosição) > distância(msg.posição) então
10      cancelaEnvio(msg.id)
11    fim
12  fim
13 fim
14 Procedimento V2I(msg)
15   se msg.tipo = Atualização e !msgRecebida(msg) então
16     mensagemRecebida(msg.id, True);
17     atualizaTabela(msg.viaid, msg.duração);
18   fim
19   se msg.tipo = Requisição e !msgEnviada(msg) então
20     mensagemEnviada(msg.id, True);
21     novaMsg = BaseDados(msg), tipo = Resposta;
22     envia(novaMsg);
23   fim
24 fim
```

---

de toda a malha viária, enquanto  $A$  é o conjunto de arestas (vias). Os pesos das arestas do grafo  $G$  representam o tempo de travessia das vias e são atualizados utilizando as informações recebidas da RSU (ou seja, tempo de travessia das vias pertencentes à célula que o veículo se encontra). As vias que estão fora da célula de cobertura da RSU são atualizadas com o tempo de travessia esperado para fluxo livre, ou seja, considerando a velocidade máxima permitida e o tamanho da via. Utilizando o  $ID$  da via atual e da via de destino, o veículo executa um algoritmo de caminho mínimo para encontrar uma nova rota com o menor tempo de viagem. Se o tempo encontrado for menor do que o tempo estimado da rota atual, o veículo inicia o novo percurso.

Utilizando como base os algoritmos da literatura sobre gerenciamento de tráfego em redes veiculares [Brennan et al. 2016, Pan et al. 2017], os veículos recebem notificações de caminhos alternativos em intervalos de tempo pré-definidos. Ou seja, a cada  $u$  unidades de tempo os veículos podem calcular novas rotas. Diferentemente dessas abordagens em que o intervalo de tempo  $u$  de todos os veículos da rede se esgotam no mesmo tempo, na abordagem proposta cada veículo possui seu intervalo individual. Isso é feito para que, quando o veículo encaminhe uma mensagem de resposta para algum veículo, ele atualize seu grafo  $G$  com as informações recebidas. Se o veículo verificar que já se passaram  $u/2$  unidade de tempo em relação à sua última verificação de novas rotas, uma nova rota pode ser calculada. Essa estratégia tem como objetivo reduzir a quantidade de requisições realizadas pelo sistema de gerenciamento de tráfego e, ao mesmo tempo, reduzir a utilização do meio sem fio para que outras aplicações possam ser executadas sem interferência de acesso ao canal sem fio. Após o cálculo da nova rota, um temporizador é novamente iniciado com o tempo  $u$ .

O processo de alteração de rota é ilustrado na figura 2, onde cada célula de voroni possui uma RSU e uma cor associada para delimitar seu escopo. O veículo que está na célula verde solicita à RSU (verde) as informações do tempo de travessia das vias da

célula e todas as arestas da célula verde são atualizadas no grafo  $G$ , enquanto as arestas das regiões azul, vermelha e amarela são atualizadas com o tempo de travessia com tráfego livre. Utilizando as informações recebidas, o veículo é capaz de encontrar uma rota com menor tempo de viagem (rota verde) em comparação com a rota anterior (marrom).

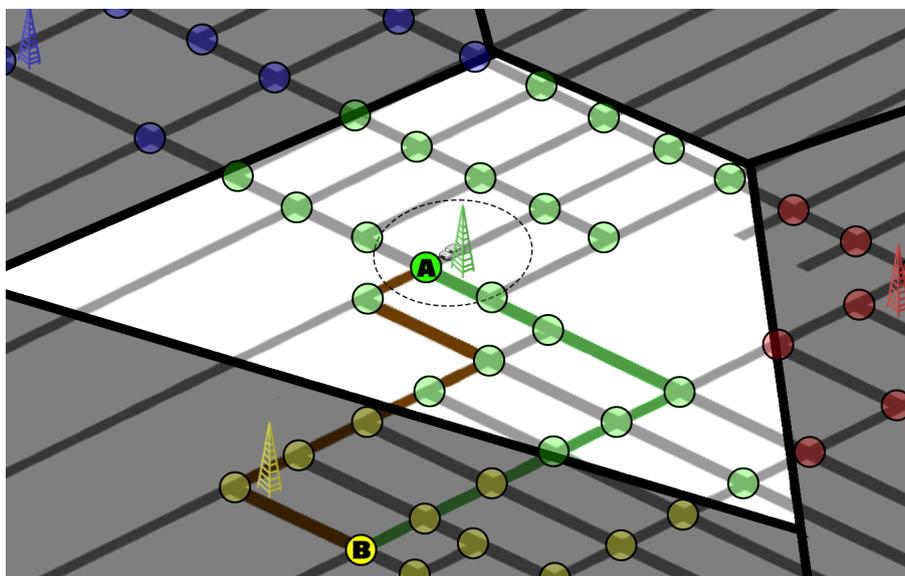


Figura 2. Ilustração do GRIFO

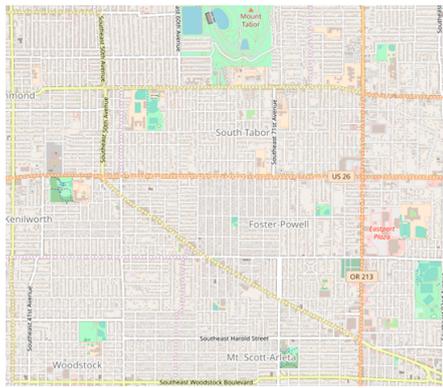
#### 4. Metodologia de Avaliação

A avaliação das soluções foi realizada utilizando um conjunto de simuladores específicos para redes veiculares. Para modelar a mobilidade dos veículos, foi utilizado o SUMO (*Simulation of Urban Mobility*) em sua versão 0.30.0. Utilizamos o simulador OMNET++ versão 5.2.1 juntamente com o *framework* Veins [Sommer et al. 2011] para simular a comunicação 802.11p entre os veículos e infraestrutura. O cenário de simulação escolhido foi uma área de 17km<sup>2</sup> da região metropolitana da cidade de Portland – USA extraída utilizando a ferramenta OpenStreetMaps<sup>1</sup>. As topologias do cenário do OpenStreetMaps e do SUMO são ilustradas na figura 3.

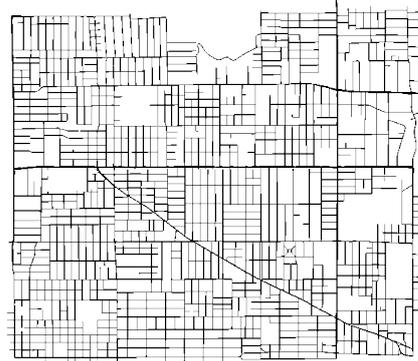
Para realizar a comparação com diferentes condições de tráfego, são utilizadas diferentes densidades de veículos, variando-se de 50–350 veículos por km<sup>2</sup> em intervalos de 50, ou seja, de 850 a 5.950 veículos considerando-se 17km<sup>2</sup>. Uma rede com 850 veículos representa um cenário sem congestionamentos, enquanto 5.950 veículos representa uma rede muito congestionada. As rotas iniciais de cada veículo são geradas aleatoriamente como um par origem-destino segundo o menor caminho entre eles. O raio de comunicação dos veículos e da RSU é de 250m e a largura de banda é de 18Mbps. Cada cenário foi executado 33 vezes utilizando diferentes sementes de geração de números aleatórios para se obter um intervalo de confiança de 95%.

A solução proposta GRIFO foi comparada com o DSP, RkSP, FOX e OVMT (*Original Vehicle Mobility Trace*, que representa a trajetória inicial dos veículos sem controle de congestionamento). Para todas as soluções avaliadas foi definido um intervalo de verificação de rotas alternativas de 60 segundos. Foi definido  $k=3$  para a quantidade

<sup>1</sup><http://www.openstreetmap.org/>



(a) Mapa do OpenStreetMaps



(b) Mapa do SUMO

Figura 3. Área de 17km<sup>2</sup> da cidade de Portland – USA

de caminhos alternativos das soluções RkSP e FOX. Foram depositadas 30 RSUs aleatoriamente no mapa para a execução das soluções FOX e GRIFO. A figura 1 ilustra um exemplo de deposição de 30 RSUs na cidade de Portland, vale notar que para o GRIFO cada semente utiliza uma deposição diferente.

O Controle de congestionamento das soluções foi avaliado considerando as seguintes métricas: **mensagens transmitidas**: quantidade de mensagens para realizar o controle do congestionamento; **tempo médio de viagem**: tempo médio gasto pelos veículos para saírem de suas origens e chegarem a seus respectivos destinos; **distância percorrida**: média das distâncias percorridas pelos veículos; **velocidade média**: velocidade média durante os percursos dos veículos; e **tempo de conclusão por fração de veículos**: tempo médio de viagem considerando as seguintes quantidade de veículos que chegam ao destino: 20%–[0 a 20%], 40%–]20 a 40%], 60%–]40 a 60%], 80%–]60 a 80%] e 100%–]80 a 100%] da quantidade total de veículos.

## 5. Resultados e Discussão

Uma questão importante na solução proposta é verificar o tamanho das mensagens transmitidas entre veículos e RSUs, uma vez que a RSU responde a uma solicitação com uma mensagem contendo informações de todas as vias de sua célula de voronoi. Uma mensagem de requisição contém 8 campos (apresentados na seção 3.2) e, considerando 4 bytes para cada campo, a mensagem possui uma carga útil de 32 bytes. Convertendo o mapa de Portland (figura 3(b)) no grafo  $G=(V,A)$  (apresentado na seção 3.3), temos um grafo com 7.680 arestas (vias). É importante ressaltar que uma rua com o mesmo nome é representada por múltiplas arestas (vias) no grafo  $G$ , como pode ser observado na figura 2 (rua com várias junções). Considerando que cada aresta possui um ID de 4 bytes e um peso de 4 bytes, o grafo total resultante ocupa 61,44 Kbytes. Considerando 30 células e cada uma contendo 256 vias, a mensagem de resposta de uma requisição terá 12 bytes referentes aos três campos e, em média, mais 2,04 Kbytes de carga útil.

A figura 4 apresenta a avaliação da quantidade de requisições geradas e mensagens transmitidas considerando a comparação do GRIFO com o FOX. A solução FOX apresenta maior similaridade com o trabalho proposto e também utiliza infraestrutura auxiliar, diferentemente das soluções DSP e RkSP que são baseadas em servidores centrais. Pode-se perceber que, considerando a quantidade de requisições (figura 4(a)) ou

a quantidade de mensagens transmitidas (figura 4(b)), para o funcionamento completo do sistema, a solução proposta utiliza menos recursos da rede. Considerando 3.400 veículos, a solução GRIFO necessita de apenas 10% do total de requisições realizadas pela solução FOX. Quando a quantidade de veículos na rede é maior (5.950 veículos), a solução proposta reduz em 96% o número de requisições realizadas. Isso acontece porque na solução proposta, quando um veículo repassa uma mensagem de resposta da RSU contendo informações das vias, ele atualiza a sua base de dados e essa informação pode ser utilizada para o cálculo de novas rotas, não necessitando realizar nova requisição. Esse reaproveitamento não poderia ser utilizado pelo FOX pois, diferentemente da solução proposta, a resposta da infraestrutura é endereçada a um veículo contendo a rota que ele deve seguir. Considerando o gráfico de mensagens transmitidas e 3.400 veículos, o GRIFO reduz em 87% a quantidade de mensagens, enquanto a redução chega a 91% ao considerar um cenário mais congestionado (5.950 veículos).

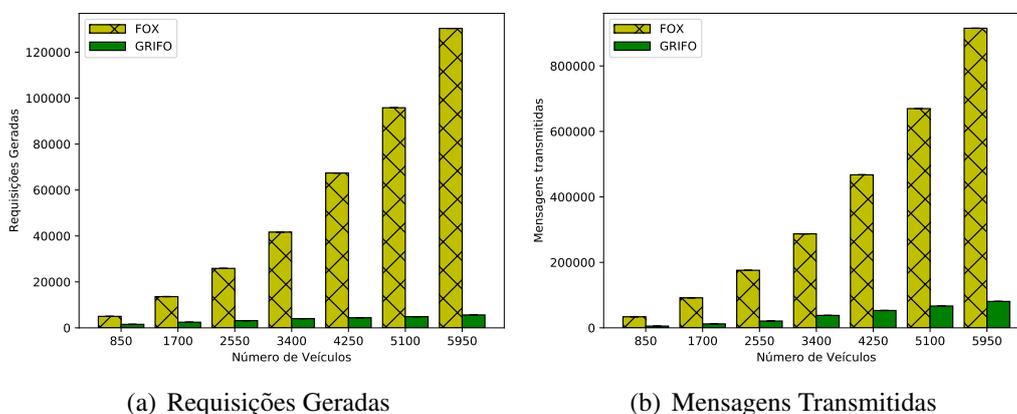


Figura 4. Avaliação do número de mensagens

A figura 5 apresenta os resultados considerando as métricas de congestionamento. De maneira geral, pode-se perceber que as soluções centralizadas DSP e RkSP apresentam resultados melhores em comparação com o FOX. Entretanto, a solução proposta GRIFO apresenta resultados melhores ou similares em comparação com o RkSP para as métricas avaliadas. Quando consideramos 3.400 veículos, a solução OVMT, que não modifica as rotas dos veículos, possui um tempo médio de viagem de 20 minutos, sendo que 14 minutos foram gastos em congestionamento, possuindo uma velocidade média de 9km/h, entretanto, percorrendo a menor distância média por viagem (3km). É importante observar que mesmo com o aumento da quantidade de veículos, a distância percorrida pelo OVMT se mantém a mesma, porém, aumentando o tempo perdido no congestionamento e, por consequência, uma maior média no tempo de viagem. As soluções FOX, RkSP, DSP e GRIFO possuem uma redução do tempo médio de viagem de 26%, 43%, 45%, 47% respectivamente considerando 3.400 veículos (figura 5(a)). Entretanto, quando a quantidade de veículos é 5.950, a solução proposta GRIFO reduz em 13%, 32%, 40% e 68% em comparação com RkSP, DSP, FOX e OVMT, respectivamente.

De maneira geral, a distância média percorrida (figura 5(b)) pelas soluções avaliadas é maior em comparação com o OVMT. Isso é esperado, uma vez que rotas mais rápidas não são necessariamente mais curtas. Entretanto, a solução GRIFO possui as menores distâncias entre as soluções avaliadas, sendo 33%, 14% e 7% inferiores em

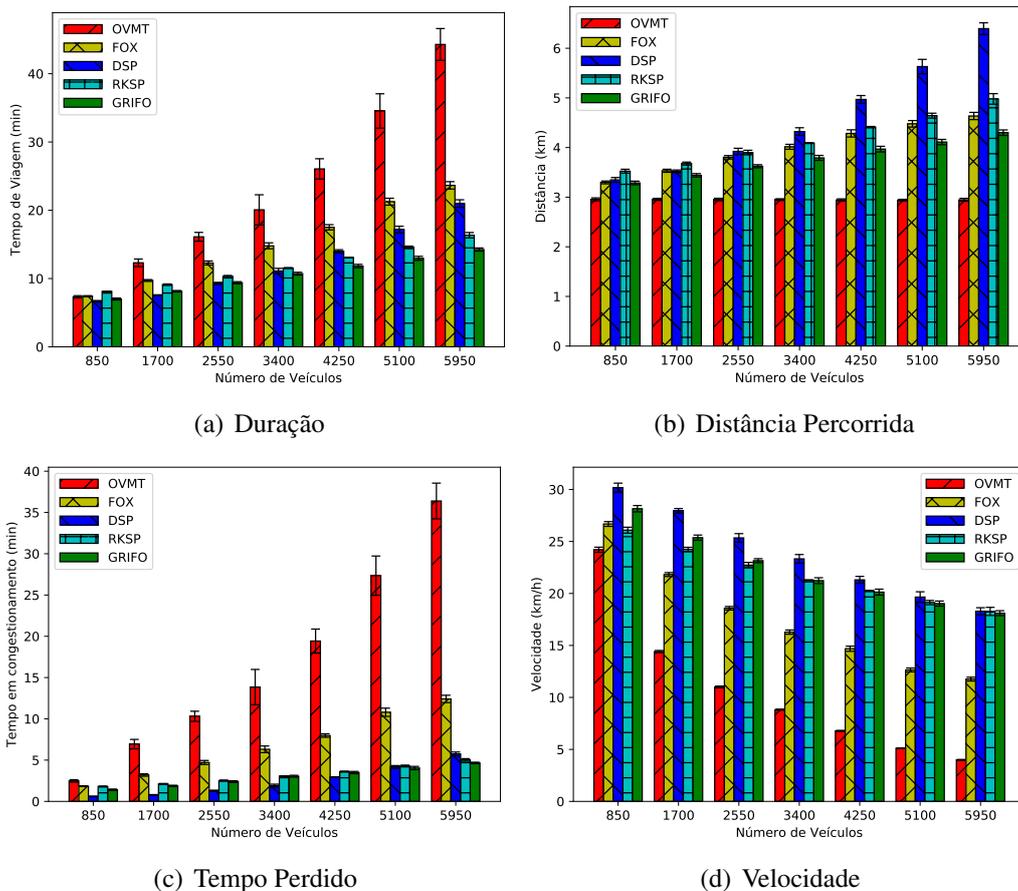


Figura 5. Avaliação das soluções.

comparação com o DSP, RkSP e FOX considerando 5.950 veículos. Como as soluções FOX e RkSP utilizam  $k$  rotas alternativas, o congestionamento não é movido para outra localização e menores distâncias são percorridas em comparação com DSP. Considerando o tempo perdido (figura 5(c)), as soluções DSP, RkSP e GRIFO apresentam resultados similares com o aumento do congestionamento, possuindo resultados inferiores ao OVMT e FOX. O mesmo comportamento é observado na velocidade média dos veículos (figura 5(d)). Com mais veículos, a velocidade média de DSP, RkSP e GRIFO são similares e, em média, 56% superior ao FOX e 360% superior ao OVMT.

A figura 6 mostra o tempo médio de viagem encontrado pelas abordagens considerando diferentes quantidades de veículos que chegaram ao destino final. Considerando 850 veículos, o tempo médio de viagem dos 20% primeiros veículos que chegaram ao destino é de 3,8 minutos para todas as soluções. Entretanto, quando observamos os últimos 20% dos veículos que chegaram ao destino (rótulo 100% no eixo-x), percebe-se que o GRIFO, DSP e FOX possuem, em média, um tempo de viagem de 19 minutos, contra 27 minutos para o RkSP e 30 minutos para o OVMT. Como o cenário possui pouco congestionamento, o RkSP não consegue bons resultados para os maiores tempos de viagens. Ao aumentarmos a quantidade de veículos, o FOX possui o pior resultado em comparação com as demais soluções que realizam o re-roteamento de veículos. Considerando os 20% veículos finais na figura 6(b), DSP e GRIFO possuem um tempo médio de viagem de 32 minutos, onde o RkSP e FOX possuem um tempo médio de 41 e 47 minutos, respectiva-

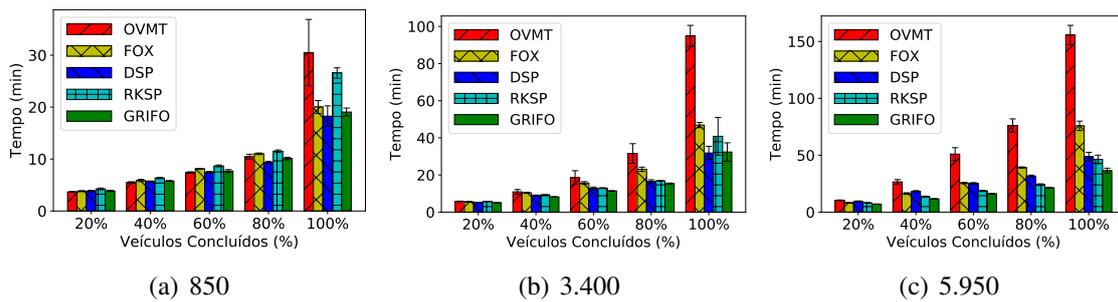


Figura 6. Comportamento das soluções para diferentes quantidade de veículos.

mente. Essa diferença se torna ainda maior quando avaliamos 5.950 veículos e considerando os últimos 20% dos veículos que chegaram ao destino, nesse caso, o OVMT possui um tempo médio de viagem de 156 minutos. As soluções de controle de congestionamento conseguem reduzir o tempo médio de viagem dos veículos para 76, 49, 46 e 36 minutos considerando o FOX, DSP, RkSP e GRIFO, respectivamente.

## 6. Conclusão e Trabalhos Futuros

Neste trabalho foi proposto o GRIFO, um sistema de controle de congestionamento que utiliza uma infraestrutura auxiliar para armazenar informações de tráfego. Após a deposição das RSUs, a região de responsabilidade de cada RSU é determinada considerando um diagrama de voronoi, onde as RSUs são o centro das células de voronoi. No sistema proposto, o veículo requisita informação sobre tráfego para a RSU e o próprio veículo calcula uma nova rota quando necessário. Dessa forma, a solução proposta consegue reduzir em 91% a quantidade de mensagens transmitidas para realizar o gerenciamento do tráfego. Quando a rede possui muitos congestionamentos, a solução proposta reduz o tempo de viagem em 70% comparado com a solução que não altera as rotas dos veículos. Além disso, a solução proposta reduz o tempo de viagem em mais de 20% em comparação com soluções da literatura que realizam o gerenciamento de tráfego. Como trabalhos futuros, pretende-se propor heurísticas para o problema de deposição de RSU que considerem o fluxo de veículos na célula com o objetivo de distribuição da sobrecarga. Além disso, serão investigadas novas soluções baseadas em comunicação *geocast* para a comunicação V2V.

## Agradecimntos

Os autores gostariam de agradecer a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) sob concessão No.: APQ-03120-17 e a Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) sob concessão No.: 150545/2018-5. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## Referências

- Atallah, R., Khabbaz, M., and Assi, C. (2017). Multihop v2i communications: A feasibility study, modeling, and performance analysis. *IEEE Transactions on Vehicular Technology*, 66(3):2801–2810.
- Board, T. R. Hcm 2010 - highway capacity manual. national research council.
- Brennand, C. A., da Cunha, F. D., Maia, G., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2016). Fox: A traffic management system of computer-based vehicles fog. In

- Computers and Communication (ISCC), 2016 IEEE Symposium on*, pages 982–987. IEEE.
- Cookson, G. (2018). Inrix global traffic scorecard. [https://www.dmagazine.com/wp-content/uploads/2018/02/INRIX\\_2017\\_Traffic\\_Scorecard\\_Final\\_2.pdf](https://www.dmagazine.com/wp-content/uploads/2018/02/INRIX_2017_Traffic_Scorecard_Final_2.pdf).
- Doolan, R. and Muntean, G. (2013). Vanet-enabled eco-friendly road characteristics-aware routing for vehicular traffic. In *IEEE 77th Vehicular Technology Conference (VTC Spring)*, pages 1–5.
- Inc, C. S. (2005). Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. Technical report, US Federal Highway Administration.
- Li, G., He, B., and Du, A. (2018). A traffic congestion aware vehicle-to-vehicle communication framework based on voronoi diagram and information granularity. *Peer-to-Peer Networking and Applications*, 11(1):124–138.
- Meneguette, R. I., Filho, G. P. R., Guidoni, D. L., Pessin, G., Villas, L. A., and Ueyama, J. (2016). Increasing intelligence in inter-vehicle communications to reduce traffic congestions: Experiments in urban and highway environments. *PLOS ONE*, 11(8):1–25.
- Njus, E. (2018). Portland-area traffic congestion is still getting worse, report says. [https://www.oregonlive.com/commuting/index.ssf/2018/02/portland-area\\_traffic\\_congesti.html](https://www.oregonlive.com/commuting/index.ssf/2018/02/portland-area_traffic_congesti.html).
- Pan, J., Popa, I. S., and Borcea, C. (2017). Divert: A distributed vehicular traffic re-routing system for congestion avoidance. *IEEE Transactions on Mobile Computing*, 16(1):58–72.
- Pittman, T. (2018). Portland traffic congestion among worst in u.s., inrix reports. <https://www.kgw.com/article/traffic/portland-traffic-congestion-among-worst-in-us-inrix-reports/283-515262476>.
- Ramalho, G. (2018). Brasil perde r\$ 267 bilhões por ano com congestionamentos. <https://g1.globo.com/globonews/noticia/2018/08/07/brasil-perde-r-267-bi-por-ano-com-congestionamentos.ghtml>.
- Silva, C. M., Masini, B. M., Ferrari, G., and Thibault, I. (2017). A Survey on Infrastructure-Based Vehicular Networks. *Mobile Information Systems*, 2017, Article ID 6123868:28–56.
- Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15.
- Souza, A., Guidoni, D. L., Botega, L., and Villas, L. (2016). Co-op: Uma solução para a detecção, classificação e minimização de congestionamentos de veículos utilizando roteamento cooperativo. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016)*, pages 1–14. SBC.