

# GeOASDVN: Um protocolo geocast ciente de obstáculos baseado em Redes Veiculares Definidas por Software

Felipe Saraiva da Costa<sup>1</sup>, Roniel Soares de Sousa<sup>2</sup>, André C. B. Soares<sup>1</sup>,  
Antonio A. F. Loureiro<sup>2</sup>, Luiz Filipe M. Vieira<sup>2</sup>

<sup>1</sup> Departamento de Computação  
Universidade Federal do Piauí (UFPI)  
CEP 64.049-550 – Teresina – PI – Brasil

<sup>2</sup> Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais (UFMG)  
CEP 31270-901 – Belo Horizonte – MG – Brasil

felipesaraivac@outlook.com, andre.soares@ufpi.edu.br,  
{ronisds, loureiro, lfvieira}@dcc.ufmg.br

**Abstract.** *Vehicular Ad-hoc Networks (VANETs) represents a technology that allows the use of network services by drivers and passengers in vehicles. Due the high mobility of the vehicles, the VANETs topology is very dynamic and represents a challenge for the development of communication protocols. In this context, the Software Defined Vehicular Network (SDVN) paradigm emerges as a promising alternative that enables the creation of flexible and adaptive protocols. This paper presents a geocast dissemination protocol that uses SDVN information from the environment and vehicles to carry out the dissemination of messages in an intelligent way. The results obtained through simulations show that the protocol presents a more efficient behavior than the opponents, in both scenarios that consider buildings and scenarios that do not consider.*

**Resumo.** *As Redes Veiculares (VANETs) permitem a utilização de serviços de rede por motoristas e passageiros em veículos. Devido à alta mobilidade dos veículos, a topologia das VANETs é dinâmica e representa um desafio para o desenvolvimento de protocolos de comunicação. Neste contexto, o paradigma de Redes Veiculares Definidas por Software (SDVN) surge como uma alternativa promissora que permite a criação de protocolos flexíveis e adaptáveis. Este trabalho apresenta um protocolo de disseminação geocast que utiliza SDVN e informações do ambiente e dos veículos para otimizar a disseminação de mensagens. Os resultados obtidos através de simulações mostram que o protocolo apresenta um comportamento mais eficiente que os adversários, tanto em cenários que consideram construções (realístico) quanto em cenários que não consideram.*

## 1. Introdução

As Redes Veiculares, ou VANETs (*Vehicular Ad Hoc Networks*), são redes que permitem a troca de informações entre veículos (comunicação V2V - *Vehicle-to-Vehicle*) e entre veículos e infra-estruturas localizadas às margens das vias (comunicação V2I

- *Vehicle-to-Infrastructure*). O ambiente em que as VANETs estão inseridas possuem características particulares que representam desafios na comunicação entre veículos [Hartenstein and Laberteaux 2008].

Uma necessidade das VANETs é a disseminação de informações em regiões geograficamente definidas. A disseminação por região permite o envio de alertas de acidentes, informações sobre vias e serviços de assinatura por região. Por exemplo, uma aplicação de segurança no trânsito que precisa informar a ocorrência de um acidente grave próximo em um local movimentado. Tal aplicação precisa que as informações cheguem rapidamente, tanto às autoridades mais próximas, quanto aos motoristas da região. O nome dado para esse modo de disseminação é *geocast* [Maihofer 2004].

Na literatura existem propostas de protocolos de disseminação *geocast* que atendem diferentes cenários. Têm-se protocolos voltados para redes esparsas [Rahbar et al. 2010], rodovias [Bachir and Benslimane 2003], ou ambiente urbano [Kaiwartya et al. 2013], por exemplo. Propostas atuais são baseadas em protocolos já estabelecidos e atacam outros problemas, como redução no número de mensagens [Voicu et al. 2014] ou custo energético [Kumar et al. 2016].

Pelo fato da VANET ser formada por veículos, sua topologia muda com frequência e, uma aplicação que pode enfrentar problemas com as mudanças de topologia. Considerando tais problemas, estudos recentes propõem o uso do conceito de SDN (*Software Defined Networks*) no ambiente veicular [Macedo et al. 2015]. Nessas redes é adotada a nomenclatura de SDVN (*Software Defined Vehicular Network*) [He et al. 2016], que compreende as redes veiculares que utilizam da abordagem SDN. Arquiteturas SDVN foram propostas em [Ku et al. 2014, Kazmi et al. 2016, He et al. 2016] e são consideradas atrativas por serem redes adaptativas e programáveis, além de possuírem um controlador que mantém o estado da rede e dos veículos, facilitando assim a obtenção de informações como localização ou previsão de rota.

Este trabalho propõe um protocolo de disseminação *geocast* denominado GeOASDVN (*Geocast Obstacle-Aware protocol based on Software Defined Vehicular Networks*). O GeOASDVN utiliza informações do ambiente e dos veículos para realizar a disseminação de mensagens de forma eficiente. O protocolo proposto utiliza as tecnologias LTE e o WAVE para comunicação, além do uso do controlador e das informações do ambiente a que ele tem acesso para otimizar a troca de mensagens.

O restante deste artigo está organizado como segue. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 detalha o funcionamento do protocolo proposto. Os resultados do estudo de avaliação de desempenho são discutidos na Seção 4 e, por fim, a Seção 5 apresenta as conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

Nesta seção são apresentados trabalhos relacionados e uma visão geral sobre disseminação de mensagens.

Diferentes protocolos de disseminação *Geocast* foram propostos na literatura. O protocolo mais simples é o *Flooding* ou Inundação, no qual as mensagens são retransmitidas sem nenhum controle para os nós vizinhos, e tem uma alta taxa de entrega de pacotes [Maihofer 2004]. Apesar de atingir um grande percentual de veículos, a falta de controle

gera uma *Broadcast Storm* ou tempestade de *broadcast*, em que o alto número de pacotes acaba congestionando a rede.

Para diminuir esse problema, existem propostas que são baseadas em inundação, mas tais propostas utilizam algum controle com o objetivo de reduzir o número de mensagens geradas. Também existem estratégias baseadas em próximo salto (NHV - *Next Hop Vehicle*), em que os veículos que irão disseminar são escolhidos por uma estratégia ou heurística, como o IVG [Bachir and Benslimane 2003].

Outra diferenciação dentro dos protocolos de disseminação *geocast* é o cenário considerado em cada proposta. Algumas propostas visam a atuação em um cenário de rodovias, por exemplo o protocolo proposto por [Bachir and Benslimane 2003]. Enquanto outras propostas visam o cenário urbano [Kaiwartya et al. 2013], ou ambos [Joshi et al. 2007]. Alguns protocolos fazem o uso de RSUs (*Road Side Unit*), que são infraestruturas auxiliares localizadas às margens das vias [Li et al. 2017].

Propostas mais recentes focam no aperfeiçoamento de protocolos existentes por meio de técnicas como: redução no número de mensagens trocadas utilizando como parâmetro a relação sinal ruído [Voicu et al. 2014]; técnicas de controle de acesso ao meio [Omar et al. 2013]; e soluções com baixo custo energético [Kumar et al. 2016].

[Sousa et al. 2017] propõe um protocolo baseado em SDVN que utiliza comunicação LTE para tráfego de dados de controle, e WAVE para comunicação inter-veicular. O trabalho aqui proposto vem como objetivo de melhorar o protocolo proposto por [Sousa et al. 2017]. Pois em um estudo mostrou que em cenários realísticos, onde estão presentes construções, o protocolo sofre queda significativa de desempenho. Por ser o trabalho mais relevante para este estudo, seu funcionamento é explicado detalhadamente na Seção 3.

O protocolo IVG [Bachir and Benslimane 2003] foi escolhido para a avaliação de desempenho por ser o protocolo mais citado na literatura e por ser independente de RSUs. O propósito do IVG é informar veículos de uma região de risco, chamada de grupo *multicast*, sobre qualquer perigo em rodovias. Além do IVG, o protocolo aqui proposto também é comparado ao *Flooding*, pois apesar de gerar muitas mensagens, apresenta alta taxa de entrega.

### **3. GeOASDVN: Protocolo de disseminação *geocast* ciente de obstáculos**

Nesta seção é detalhado o funcionamento do protocolo GeOASDVN. O protocolo é baseado no protocolo Geo-SDVN, proposto por [Sousa et al. 2017]. A Seção 3.1 detalha o funcionamento do Geo-SDVN. Enquanto que a Seção 3.2 apresenta melhorias adicionadas no GeOASDVN tornando-o ciente de obstáculos.

#### **3.1. Geo-SDVN**

A Figura 1 descreve a arquitetura utilizada. A comunicação entre o controlador SDVN e os veículos é feita utilizando LTE, pois o protocolo não utiliza RSUs e cada veículo tem comunicação direta com o controlador SDVN. A comunicação entre veículos é feita utilizando a interface WAVE.

No protocolo, cada veículo representa um *switch* OpenFlow e possui uma *flow table*. A *flow table* possui dois campos de correspondência, o *Vehicle ID* e o *Geocast ID*.

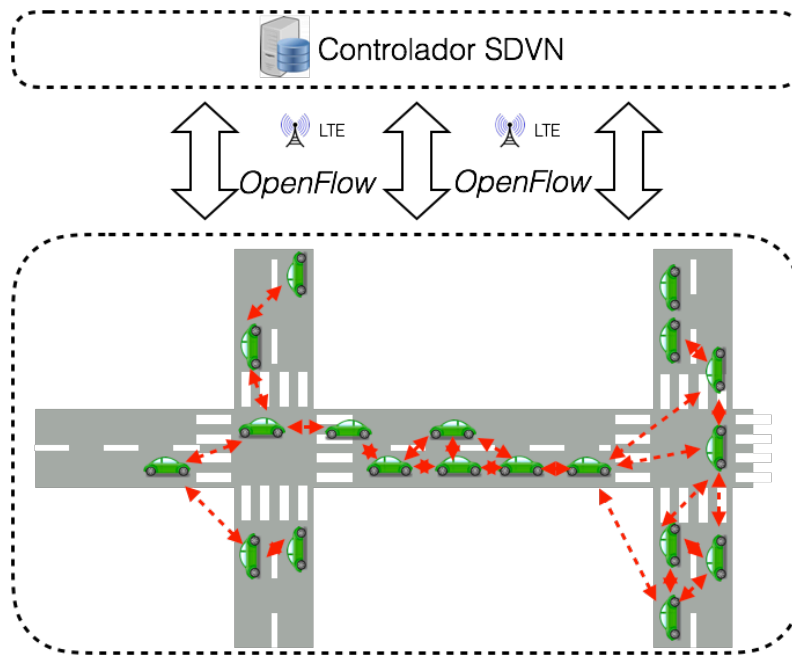


Figura 1. Arquitetura SDVN utilizada. Retirada de [Sousa et al. 2017].

O campo *Vehicle ID* identifica o veículo que gerou a mensagem que deve ser transmitida por meio do protocolo *geocast*. Já o campo *Geocast ID* é um identificador da mensagem. A sua utilização é da seguinte forma: sempre que uma aplicação deseja transmitir uma mensagem *geocast* para uma região *R*, será criado um identificador único *Geocast ID* para aquela mensagem. Se em algum momento uma dada aplicação deseja transmitir uma mensagem na mesma região *R*, será utilizado o mesmo identificador *Geocast ID* criado anteriormente.

A *flow table* possui ainda um campo *Timestamp*, que armazena o momento em que a *flow-entry* foi inserida na *flow table*. O campo *Timestamp* é utilizado para remover uma *flow-entry* quando esta passar um tempo determinado (*Flow-Entry Lifetime*) sem ser atualizada.

Por fim, a *flow table* possui dois campos que identificam a ação a ser realizada com os pacotes do fluxo. São eles o campo *Actions* e o campo *Next\_Hop\_IDs*. O campo *Actions* possui um dos seguintes valores:

- **DROP**: O veículo deve descartar o pacote.
- **SEND\_TO\_CONTROLLER**: O veículo deve encaminhar o pacote de dados para o controlador utilizando a rede LTE.
- **BROADCAST**: O veículo deve encaminhar o pacote para os seus veículos vizinhos utilizando *broadcast one hop*.
- **SEND\_TO\_CONTROLLER+BROADCAST**: O veículo deve realizar as ações *SEND\_TO\_CONTROLLER* e *BROADCAST*.

Já o campo *Next\_Hop\_IDs* informa quais veículos vizinhos também irão retransmitir a mensagem. Esse campo é utilizado para que o veículo verifique se todos os vizinhos que devem retransmitir a mensagem de fato retransmitiram. Assim, o veículo que está encaminhando o pacote no momento aguarda um tempo igual a *Max.Waiting.Time*.

Se ao final deste tempo ele não receber as retransmissões de todos os *Next\_Hop\_IDs*, o veículo então transmite novamente a mensagem. Isto é realizado no máximo *Max\_Tries* vezes. A Tabela 1 exemplifica uma *flow table* com quatro *flow-entries*, uma para cada ação possível.

<i>Vehicle ID</i>	<i>Geocast ID</i>	<i>Timestamp</i>	<i>Actions</i>	<i>NextHopIDs</i>
Veh_1	5	22.5	DROP	Empty
Veh_32	1	14.1	SEND_TO_CONTROLLER	Empty
Veh_12	2	52.8	BROADCAST	Veh_1, Veh_15
Veh_15	15	5.0	SEND_TO_CONTROLLER+BROADCAST	Veh_1, Veh_32

**Tabela 1. Exemplo de uma *flow table*.**

Como discutido anteriormente, quando um veículo não possui uma entrada na *flow table* correspondente ao pacote que deseja transmitir, este envia ao controlador uma mensagem do tipo *Table-miss*. Uma mensagem *Table-miss* possui os seguintes campos:

- ***Vehicle ID***: Identificador do veículo que criou a mensagem.
- ***Geocast ID***: Identificador *geocast* da mensagem, que é criado como discutido anteriormente.
- ***Region Center***: Coordenadas que identificam o centro da região de interesse da mensagem.
- ***Region Radius***: Raio da região de interesse da mensagem.

Ao enviar uma *Table-miss*, o veículo espera *Table-miss\_Waiting\_Time* segundos. Ao final deste tempo, o veículo verifica se foi recebida uma mensagem de resposta *Table-miss\_Response* do controlador. Caso positivo, a ação presente na resposta é realizada. Caso contrário, o veículo espera novamente por *Table-miss\_Waiting\_Time* segundos uma resposta. Esta espera ocorre no máximo *Table-miss\_Max\_Waiting\_Tries* vezes. Ou seja, o veículo espera a resposta por no máximo *Table-miss\_Waiting\_Time* × *Table-miss\_Max\_Waiting\_Tries* segundos. Se ao final deste tempo não for obtida uma resposta, o veículo envia novamente uma mensagem *Table-miss* ao controlador. Acontecem no máximo *Table-miss\_Max\_Tries* tentativas de envio da *Table-miss* ao controlador. Se não for obtida a resposta do controlador, o veículo simplesmente descarta o pacote.

A mensagem de resposta do controlador, chamada de *Table-miss\_Response*, contém todos os campos presentes na mensagem *Table-miss*, mais os campos abaixo:

- ***Actions***: Ação que deve ser realizada (*DROP*, *BROADCAST*, etc).
- ***Next\_Hop\_IDs***: Lista de identificadores dos próximos saltos, como discutido anteriormente. Essa lista é vazia caso a ação não seja do tipo *BROADCAST* ou *SEND\_TO\_CONTROLLER+BROADCAST*.

## Controlador SDVN

O controlador SDVN é quem decide a ação a ser tomada por cada veículo ao receber um pacote enviado por outro veículo. A escolha das ações que cada veículo deve realizar para determinado fluxo de dados é feita da seguinte forma: todos os veículos enviam *beacons* periodicamente (por exemplo, um *beacon* por segundo), e cada *beacon* contém a posição geográfica do veículo emissor, obtida por GPS. Sendo esse envio necessário para que o controlador tenha acesso à informação global da topologia da rede e posição dos

<i>Vehicle ID</i>	<i>Position</i>	<i>Timestamp</i>
Veh.5	(80.5, 20.1, 0)	52.5
Veh.6	(20.0, 700.1, 0)	52.1
Veh.21	(850.0, 15.7, 0)	53.7
Veh.75	(50.1, 10.0, 0)	54.0

**Tabela 2. Tabela *Vehicles\_Table* armazenada no controlador com as posições dos veículos.**

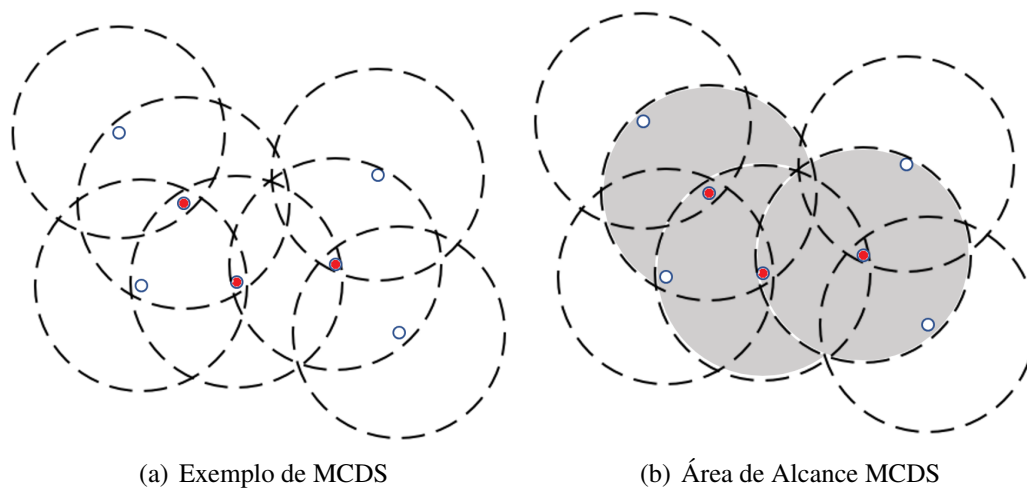
veículos. O controlador mantém uma tabela *Vehicle\_Table* com a posição de cada veículo. A Tabela 2 apresenta os campos presentes na *Vehicles\_Table*.

O campo *Vehicle ID* é o identificador único do veículo. O campo *Position* possui a posição presente no último *beacon* recebido do respectivo veículo. Já o *Timestamp* representa o momento em que foi recebido o último *beacon* do veículo *Vehicle ID*. Ao final de *Vehicle\_Entry\_Duration* segundos sem recebimento de *beacons* de um determinado veículo, a sua entrada é removida da *Vehicle\_Table*.

A partir da *Vehicle\_Table*, o controlador constrói uma matriz de adjacência correspondente ao estado atual de conexão dos veículos na rede veicular. Para isso, o controlador possui a informação da potência de transmissão dos veículos. A partir da potência de transmissão dos veículos, o controlador calcula qual o valor aproximado do raio de comunicação *Transmission\_Range* de todos os veículos, em metros. O controlador assume que dois veículos podem comunicar um com o outro se a distância entre eles for menor ou igual a *Transmission\_Range* menos um valor de segurança igual a *Guard\_Range* metros.

Sempre que o controlador recebe uma *Table-miss*, ele decide as ações que todos os veículos da região de interesse devem realizar da seguinte forma: primeiro é verificado se a região de interesse da mensagem referente ao *Table-miss* recebido está conectada. Ou seja, é verificado se é possível alcançar todos os veículos a partir do veículo que criou a mensagem, utilizando apenas a comunicação V2V. Caso positivo, os seguintes passos são executados. O primeiro é calcular o *Minimum Connected Dominating Set - MCDS* (Conjunto Dominante Conectado Mínimo) do grafo formado pelos veículos presentes na região. Porém, sabe-se que o MCDS é um problema *NP-Hard* tanto para grafos arbitrários não direcionados quanto para grafos do tipo *Unit Disk* [Clark et al. 1990] [Guha and Khuller 1998]. Assim, faz-se necessária a utilização de heurísticas para calcular soluções aproximadas de instâncias do problema MCDS em tempo factível. No trabalho, foi utilizada a heurística proposta em [Butenko et al. 2004]. Essa heurística foi utilizada por ser simples de ser implementada e apresentar bons resultados. A heurística foi alterada para que o veículo que criou a mensagem, a ser transmitida em *geocast*, sempre faça parte do MCDS calculado.

Os veículos presentes no conjunto dominante retransmitem a mensagem, enquanto os demais veículos descartam-na. Isso evita a ocorrência de uma tempestade de *broadcast*. A Figura 2(a) ilustra um MCDS, em que os veículos marcados em vermelho são os selecionados para retransmitir a mensagem; os veículos não marcados vão descartar a mensagem; e as linhas pontilhadas representam o alcance de transmissão de cada veículo. Enquanto a Figura 2(b) ilustra a área de alcance dos veículos do MCDS calculado. Percebe-se que todos os veículos podem ser alcançados por algum dos veículos em vermelho.



**Figura 2. Exemplo de MCDS e área de alcance.**

O controlador calcula os *Next\_Hop\_IDs* de cada veículo que deverá retransmitir a mensagem da seguinte forma. Cria-se um subgrafo formado apenas pelos veículos que deverão retransmitir a mensagem, ou seja, aqueles contidos no MCDS. Após isso, é realizada uma busca em profundidade a partir do veículo que criou a mensagem. Durante a busca em profundidade, quando visita-se um determinado nó, todos os seus vizinhos que ainda não foram visitados são adicionados à sua lista de *Next\_Hop\_IDs*.

Assim, após calcular o MCDS aproximado e as listas de *Next\_Hop\_IDs*, o controlador enviará uma mensagem do tipo *Table-miss\_Response* para cada veículo presente na região de interesse. Os veículos que estão no conjunto dominante receberão uma *Table-miss\_Response* com a ação *BROADCAST* e seus respectivos *Next\_Hop\_IDs*, e os demais veículos receberão uma *Table-miss\_Response* com a ação *DROP*.

Assim que o veículo que originou a mensagem verificar que acabou o tempo correspondente ao *Table-miss\_Waiting\_Time*, este executará a ação presente na *Table-miss\_Response* recebida. Ao receberem o pacote enviado pelo veículo que originou a mensagem, os demais veículos contidos no MCDS irão retransmiti-la, pois estes já terão recebido mensagens *Table-miss\_Response* do controlador.

Com isso, o controlador consegue tomar as decisões quando a região de interesse da mensagem está conectada. Em cenários onde a densidade de veículos da região de interesse é baixa, pode acontecer da região em questão não ser conectada. Neste caso, o controlador detecta cada sub-região conectada dentro da região de interesse, e executa os mesmos passos utilizados quando a região é conectada para cada uma dessas sub-regiões conectadas. O controlador também escolhe um veículo de cada sub-região para agir como o veículo que originou a mensagem, e iniciar a disseminação da mensagem na sua sub-região conectada. Este veículo é chamado de *Region\_Head*. Já o veículo que originou a mensagem receberá uma *Table-miss\_Response* contendo a ação *BROADCAST+SEND\_TO\_CONTROLLER*, diferente do que acontece no cenário em que a região de interesse é conectada. Isso se faz necessário para que sempre que o veículo decidir enviar uma mensagem na região, este transmita a mensagem para seus veículos vizinhos por meio do *BROADCAST* e também para o controlador por meio da rede LTE. Ao receber a

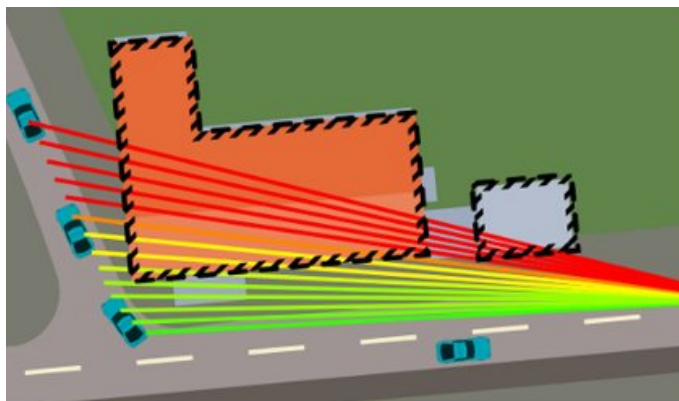
mensagem, o controlador retransmite a mensagem para todos os *Region\_Heads*, que, por sua vez, realizam a disseminação da mensagem nas sub-regiões.

O controlador armazena em uma tabela todas as mensagens *Table-miss* recebidas nos últimos *Table-miss\_Time* segundos. A cada *Update\_Table-miss\_Entry* segundos o controlador verifica se precisa atualizar as ações de algum dos veículos que estão na região referente àquela mensagem *Table-miss*. Qualquer modificação em alguma das ações criadas anteriormente faz com que o controlador envie uma *Table-miss\_Response* ao veículo correspondente informando a modificação. Essas modificações ocorrem devido à mobilidade dos veículos que faz com que a matriz de adjacência altere constantemente.

### 3.2. GeOASDVN: Geocast Obstacle-Aware protocol based on Software Defined Vehicular Networks

Como detalhado na seção 3.1, antes da disseminação do *geocast* para a região de interesse, o controlador primeiro calcula o MCDS. No Geo-SDVN, o cálculo do MCDS é feito tomando a distância dos veículos, pois o controlador sabe a posição de cada veículo e tem como base a potência média de transmissão para definir se dois veículos são mutuamente alcançáveis no raio de transmissão um do outro. Por fim, o MCDS é atualizado sempre que o controlador julgar necessário.

No entanto, em um cenário onde são considerados as construções existentes, o protocolo acaba tendo seu desempenho afetado por elas. A Figura 3 mostra o efeito de deterioração do sinal causado pela existência de construções. Na imagem é possível observar que quando não existem construções entre dois veículos, o sinal sofre pouca interferência. No entanto, o sinal sofre maior impacto à medida que ele tem que atravessar os obstáculos para alcançar seu destino.



**Figura 3. Deterioração do sinal causada pela existência de construções. Retirada de [Sommer et al. 2011]**

A Figura 4 apresenta um exemplo que pode ocorrer em um ambiente real. Nela existem dois veículos que estão a cerca de 70m um do outro. Tomando a distância como parâmetro, o protocolo assume que a comunicação entre os veículos pode ser realizada de forma eficiente. No entanto, a existência de prédios pode degradar o sinal de tal forma que uma mensagem não consiga ser entregue. No caso de um dos veículos ser um ponto de conexão entre dois conjuntos maiores de veículos, a interferência pode ocasionar a não entrega de uma mensagem para vários veículos dentro de uma região crítica.





Figura 4. Recorte do mapa de Bolonha - Itália. Obtido através do OpenStreetMap

Levando em consideração os problemas mencionados, o GeOASDVN insere no controlador a possibilidade de calcular o MCDS de forma que não esteja condicionado apenas à distância entre os veículos ou potência de transmissão, mas também os dados disponíveis sobre o ambiente em que os veículos se encontram.

Os dados sobre construções e outros obstáculos mapeados são obtidos através do OpenStreetMap, que é uma plataforma de dados aberto que mantém mapas e dados relevantes sobre eles. Através da API (Application Programming Interface) fornecida pela plataforma, é possível obter dados que foram mapeados de várias cidades pelo mundo.

O Algoritmo 1 detalha como é feito o cálculo do MCDS, no qual os dados de entrada são o conjunto de veículos, a distância máxima considerada para alcance ( $\sigma$ ), e um limiar que define o máximo de interferência a ser aceito para considerar dois veículos mutuamente alcançáveis ( $\phi$ ).

**Entrada:**  $V = \{v_1, v_2, \dots, v_n\}, \sigma, \phi$

**Saída:** Conjunto Conectado Dominante Mínimo

**início**

$listaAdjacencia \leftarrow []$

**para**  $i \leftarrow 0$  **até**  $n - 1$  **faça**

**para**  $j \leftarrow i + 1$  **até**  $n$  **faça**

**se**  $distancia(v_i, v_j) < \sigma$  **e**  $interferencia(v_i, v_j) < \phi$  **então**

$listaAdjacencia.adiciona(v_i, v_j)$

**fim**

**fim**

**fim**

**retorna**  $calculaMCDS(listaAdjacencia)$

**fim**

#### 4. Avaliação de Desempenho

Esta seção detalha o estudo de avaliação de desempenho realizado com o objetivo de verificar a eficácia do protocolo *geocast* em disseminar informações. Na seção de resultados são apresentados também dados referentes ao desempenho do IVG e do *Flooding*.

#### 4.1. Cenário

As ferramentas de simulação utilizadas foram o simulador de eventos discretos OM-NeT++ 4.6 [Varga and Hornig 2008] em conjunto com o simulador de tráfego e mobilidade urbana SUMO 0.25.0 [Krajzewicz et al. 2012]. O *framework* Veins LTE 1.3 [Hagenauer et al. 2014], que é utilizado no estudo de redes veiculares heterogêneas e possui suporte para o padrão IEEE 802.11p, foi utilizado para integrar esses dois simuladores.

O cenário de simulação está disponível na página do projeto Bologna Ringway Dataset, e consiste em um mapa da cidade de Bologna na Itália. O cenário completo disponibiliza um *trace* que corresponde ao tráfego de um dia típico da cidade, considerando um intervalo entre 8h e 9h da manhã e contém 22000 veículos.



Figura 5. Mapa do cenário *Bologna Ringway* [Bedogni et al. 2015].

A duração de cada simulação foi de 240 segundos e replicada 30 vezes. O tempo de simulação considerado é o dobro do tempo de simulação utilizado por [de Sousa et al. 2018], e o número de replicações é o mesmo. Todos os gráficos possuem intervalos com nível de confiança de 95%.

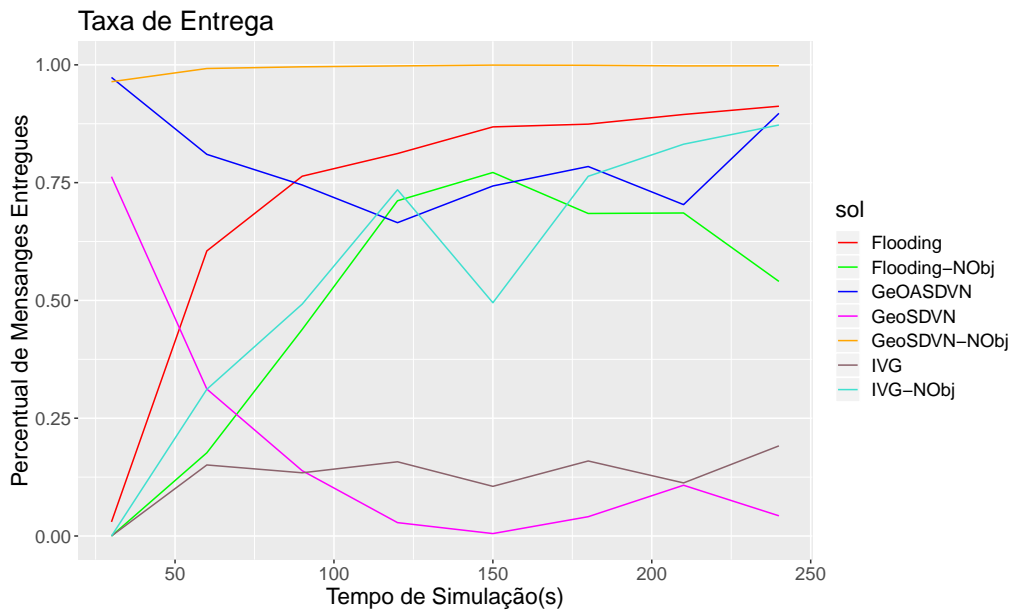
A potência de transmissão dos veículos foi de 7,1 miliwatts (mW) para a comunicação V2V, o que se traduz em um raio de comunicação de aproximadamente 300 metros. Já a comunicação LTE foi configurada de forma que um veículo tenha alcance da torre de transmissão a partir de qualquer posição do mapa. A Tabela 3 resume os principais parâmetros adotados nas simulações.

Parâmetro	Valor
Duração	240 segundos
Potência de transmissão V2V	7,1 miliwatts (300 metros de alcance)
Frequência de transmissão de <i>beacons</i> para o controlador SDVN	1 Hz
Velocidade máxima dos veículos	50 km/h
Tamanho do cabeçalho dos pacotes	17 bytes
Tamanho dos dados de um <i>beacon</i>	17 bytes

Tabela 3. Parâmetros utilizados nas simulações do cenário *Bologna Ringway*.

#### 4.2. Resultados

Para esta avaliação, o comportamento dos protocolos foi analisado considerando dois cenários: no primeiro cenário, não é considerada a existência de obstáculos (NObj), enquanto que no segundo é avaliado o impacto das construções no desempenho dos protocolos. O protocolo aqui proposto não foi simulado em cenário sem obstáculos por apresentar

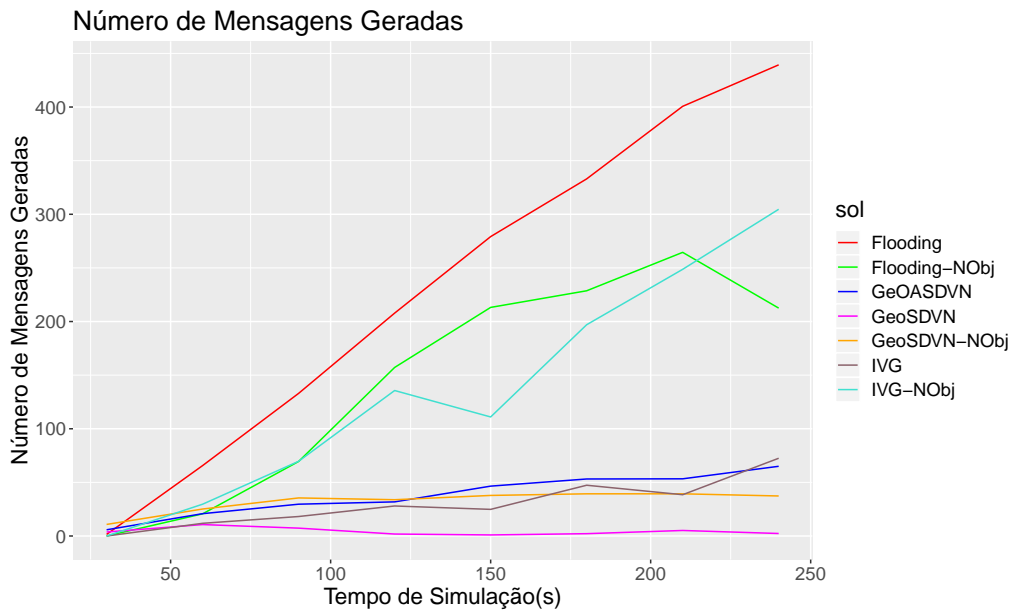


**Figura 6. Eficiência de entrega dentro da região de interesse.**

comportamento semelhante ao Geo-SDVN. As métricas de avaliação utilizadas são: (i) taxa de entrega de mensagens, (ii) *overhead* de sinalização e (iii) atraso na entrega de conteúdo.

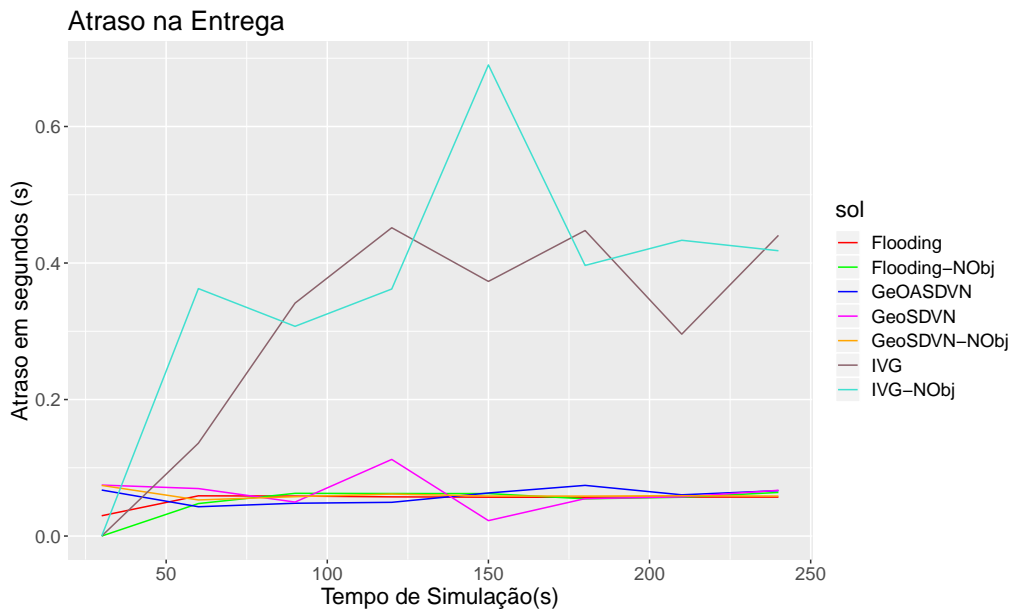
A Figura 6 apresenta o impacto de construções no desempenho do protocolo, no qual o Geo-SDVN, em cenário desconsiderando construções, apresenta taxa de entrega de quase 100%, enquanto que ao considerar construções, a taxa de entrega permanece abaixo de 10% maior parte do tempo de simulação, pois as construções tem influência na forma em que os CMDS são construídos. O Flooding não sofre impacto negativo significativo no cenário com obstáculos devido ao funcionamento do protocolo, no qual todos os veículos repassam uma mensagem recebida. O IVG apresenta uma redução significativa de desempenho entre os cenários. Por fim, o GeOASDVN apresenta taxa de entrega entre 75% e 99% durante maior parte da simulação. Por estar considerando um cenário mais próximo ao real que o cenário sem obstáculos, considera-se que apenas o Flooding e o GeOASDVN apresentam desempenho aceitável.

Na Figura 7, é mostrado o número total de mensagens enviadas durante cada disseminação *geocast*. Nessa caso são consideradas mensagens geradas e retransmitidas, e os *beacons* não são considerados por serem mensagens de controle que não precisam de tempo significativo para serem processados. De acordo com o gráfico, o *Flooding* possui maior número de mensagens enviadas, o que é esperado, considerando que no protocolo os veículos retransmitem sempre que recebem uma mensagem que não tenham recebido. O Geo-SDVN, assim como o IVG, apresenta um baixo número de mensagens geradas no cenário com obstáculos, mas isso ocorre devido aos veículos retransmissores não estarem efetivamente recebendo as mensagens. Esse comportamento do Geo-SDVN ocorre porque ao se construir o CMDS, não são levados em conta fatores que dificultam a comunicação. Já o comportamento do GeOASDVN se assemelha ao comportamento do Geo-SDVN em cenário sem construções, apresentando uma leve alta no total de mensagens enviadas, pois são escolhidos mais retransmissores para adaptar-se ao cenário.



**Figura 7. Número total de mensagens geradas/retransmitidas.**

Por fim, a Figura 8 mostra o atraso na entrega de mensagens. Isto é, o tempo médio entre o início do *geocast* até o momento em que foi recebida pelos veículos. Em ambos os cenários o IVG apresentou maior atraso, enquanto que os demais protocolos não apresentam diferença significativa de tempo. O comportamento do IVG é ocasionado pela espera que um veículo deve realizar antes de retransmitir uma mensagem.



**Figura 8. Atraso na entrega das mensagens.**

## 5. Considerações Finais

Neste trabalho foi proposto um novo protocolo *geocast* para Redes Veiculares Definidas por Software. A arquitetura utilizada independe de infraestrutura nas vias e resolve

um problema existente no seu antecessor, que por não considerar o ambiente em que os veículos se encontram ao escolher os retransmissores, acabava tendo uma queda de desempenho significativa.

O protocolo foi avaliado em termos de Taxa de Entrega, número de mensagens geradas e atraso na entrega, apresentando ótimos resultados quando comparado aos adversários. Os resultados também mostram que o protocolo apresenta boa taxa de entrega, enquanto gera um baixo número de mensagens. Por usar a tecnologia LTE, o protocolo pode utilizar uma infraestrutura que já está disponível.

## Referências

- Bachir, A. and Benslimane, A. (2003). A multicast protocol in ad hoc networks inter-vehicle geocast. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, volume 4, pages 2456–2460 vol.4.
- Bedogni, L., Gramaglia, M., Vesco, A., Fiore, M., Härri, J., and Ferrero, F. (2015). The bologna ringway dataset: Improving road network conversion in sumo and validating urban mobility via navigation services. *IEEE Transactions on Vehicular Technology*, 64(12):5464–5476.
- BolognaRingwayDataset (2016). [http://www.cs.unibo.it/projects/bolognaringway/files/bolognaringway\\_1.0.tar.gz](http://www.cs.unibo.it/projects/bolognaringway/files/bolognaringway_1.0.tar.gz).
- Butenko, S., Cheng, X., Oliveira, C. A., and Pardalos, P. M. (2004). *A New Heuristic for the Minimum Connected Dominating Set Problem on Ad Hoc Wireless Networks*, pages 61–73. Springer US, Boston, MA.
- Clark, B. N., Colbourn, C. J., and Johnson, D. S. (1990). Unit disk graphs. *Discrete Mathematics*, 86(1–3):165 – 177.
- de Sousa, R. S., da Costa, F. S., Soares, A. C. B., Vieira, L. F. M., and Loureiro, A. A. F. (2018). Geo-sdvn: A geocast protocol for software defined vehicular networks. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6.
- Guha, S. and Khuller, S. (1998). Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387.
- Hagenauer, F., Dressler, F., and Sommer, C. (2014). Poster: A simulator for heterogeneous vehicular networks. In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 185–186.
- Hartenstein, H. and Laberteaux, L. P. (2008). A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, 46(6):164–171.
- He, Z., Cao, J., and Liu, X. (2016). Sdvn: enabling rapid network innovation for heterogeneous vehicular communication. *IEEE Network*, 30(4):10–15.
- Joshi, H. P., Sichitiu, M. L., and Kihl, M. (2007). Distributed robust geocast multicast routing for inter-vehicle communication. In *Proceedings of WEIRD Workshop on WiMax, Wireless and Mobility*, pages 9–21.
- Kaiwartya, O., Kumar, S., and Kasana, R. (2013). Traffic light based time stable geocast (t-tsg) routing for urban vanets. In *2013 Sixth International Conference on Contemporary Computing (IC3)*, pages 113–117.

- Kazmi, A., Khan, M. A., and Akram, M. U. (2016). Devanet: Decentralized software-defined vanet architecture. In *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, pages 42–47.
- Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.
- Ku, I., Lu, Y., Gerla, M., Gomes, R. L., Ongaro, F., and Cerqueira, E. (2014). Towards software-defined vanet: Architecture and services. In *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pages 103–110.
- Kumar, A., Kumar, S., and Kumar, V. (2016). A novel energy efficient geocast routing algorithm for mobile ad hoc networks. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 2926–2929.
- Li, P., Zhang, T., Huang, C., Chen, X., and Fu, B. (2017). Rsu-assisted geocast in vehicular ad hoc networks. *IEEE Wireless Communications*, 24(1):53–59.
- Macedo, D. F., Guedes, D., Vieira, L. F., Vieira, M. A., and Nogueira, M. (2015). Programmable networks—from software-defined radio to software-defined networking. *IEEE communications surveys & tutorials*, 17(2):1102–1125.
- Maihofer, C. (2004). A survey of geocast routing protocols. *IEEE Communications Surveys Tutorials*, 6(2):32–42.
- Omar, H. A., Zhuang, W., and Li, L. (2013). Vemac: A tdma-based mac protocol for reliable broadcast in vanets. *IEEE Transactions on Mobile Computing*, 12(9):1724–1736.
- OpenStreetMap (2015). *OpenStreetMap*. <http://www.openstreetmap.org>.
- Rahbar, H., Naik, K., and Nayak, A. (2010). Dtsng: Dynamic time-stable geocast routing in vehicular ad hoc networks. In *2010 The 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–7.
- Sommer, C., Eckhoff, D., German, R., and Dressler, F. (2011). A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments. In *8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2011)*, pages 84–90, Bardonecchia, Italy. IEEE.
- Sousa, R. S., Loureiro, A. A. F., Vieira, L. F. M., Soares, A. C. B., and da Costa, F. S. (2017). Geo-sdvn: Um protocolo geocast para redes veiculares definidas por software. In *Anais XXXV do Simpósio Brasileiro de Redes de Computadores*, pages 487–500.
- Varga, A. and Hornig, R. (2008). An overview of the omnet++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Simutools '08, pages 60:1–60:10, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- Voicu, R. C., Abbasi, H. I., Fang, H., Kihei, B., Copeland, J. A., and Chang, Y. (2014). Fast and reliable broadcasting in vanets using snr with ack decoupling. In *2014 IEEE International Conference on Communications (ICC)*, pages 574–579.