

Geo-SDVN: Um Protocolo *Geocast* para Redes Veiculares Definidas por *Software*

Roniel Soares de Sousa¹, Antonio A. F. Loureiro¹, Luiz Filipe M. Vieira¹,
André C. B. Soares², Felipe Saraiva da Costa²

¹ Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
CEP 31270-901 – Belo Horizonte – MG – Brasil

² Departamento de Computação
Universidade Federal do Piauí (UFPI)
CEP 64.049-550 – Teresina – PI – Brasil

{ronisds, loureiro, lfvieira}@dcc.ufmg.br,

andre.soares@ufpi.edu.br, felipesaraivac@outlook.com

Abstract. *Vehicular Networks represent an emerging technology that enables the use of network services by drivers and vehicle passengers. However, the heterogeneity of vehicular networks, which can use various wireless technologies, represents a challenge for the development of communication protocols. The Software Defined Vehicular Networks (SDVN) paradigm allows flexible design of these new protocols. This work proposes a new geocast protocol for Software Defined Vehicular Networks. Simulation results reveal that the proposed protocol exhibits superior results in terms of delivery rate and signaling overhead when compared to the Flooding algorithm.*

Resumo. *As Redes Veiculares representam uma tecnologia emergente que possibilita a utilização de serviços de rede por parte dos motoristas e passageiros de veículos. Porém, a heterogeneidade das redes veiculares, que podem utilizar diversas tecnologias de redes sem fio, representa um desafio para o desenvolvimento de protocolos de comunicação. O paradigma das Redes Veiculares Definidas por Software (SDVN) possibilita uma flexibilidade no desenvolvimento destes novos protocolos. Este artigo propõe um novo protocolo de geocast para Redes Veiculares Definidas por Software. Foi realizado um estudo de avaliação de desempenho no qual o protocolo proposto apresentou resultados superiores em termos de taxa de entrega e overhead de sinalização quando comparado com o algoritmo de inundação.*

1. Introdução

As Redes Veiculares (*Vehicular Ad-hoc Networks* - VANET) são sistemas de comunicação formados principalmente por veículos. Tais redes consistem em veículos automotores que se comunicam através de redes sem fio com outros veículos ou com infraestruturas localizadas à margem das vias. Essas redes são um dos componentes que formam os Sistemas Inteligentes de Transportes (*Intelligent Transportation Systems* – ITS) [dos S. Alves et al. 2009].

Dentre as características do ambiente veicular que acarretam em desafios para as VANETs estão: (i) a alta mobilidade dos nós, (ii) as mudanças constantes na topologia da rede, (iii) a existência de redes densas que ocasionam em altas taxas de perdas de pacotes por congestionamento na rede e (iv) a existência de redes esparsas que dificultam a disseminação de informações [dos S. Alves et al. 2009].

Recentemente, alguns autores sugerem e defendem a utilização de SDN (*Software Defined Networks*) no ambiente veicular. Em [He et al. 2016], é adotada a nomenclatura SDVN (*Software Defined Vehicular Network*) para as redes veiculares que utilizam a abordagem SDN. Além disso, também são propostas arquiteturas e serviços SDVN em [Kazmi et al. 2016], [Ku et al. 2014] e [He et al. 2016]. Redes Definidas por *Software* são atrativas por serem programáveis por meio de um controlador que mantém o estado da rede e podem se adaptar à mobilidade dos nós.

Uma importante classe de comunicação em redes veiculares é o *geocast*. *Geocast* é o nome dado à disseminação de mensagens para veículos dentro de uma região definida geograficamente [Maihofer 2004]. Nos últimos anos, diferentes protocolos *geocast* foram propostos para redes veiculares tradicionais [Bachir and Benslimane 2003] [Joshi et al. 2007] [Rahbar et al. 2010], sendo cada um otimizado para um cenário específico.

Este trabalho tem foco no problema de disseminação de informações em SDVN. Neste contexto, é proposto um novo protocolo *geocast* para Redes Veiculares Definidas por *Software*. O protocolo faz uso da tecnologia LTE (*Long Term Evolution*) para otimizar o roteamento, sendo utilizado principalmente no tráfego de mensagens de controle. Isso permite que mais veículos dentro da região desejada sejam alcançados.

O protocolo proposto utiliza-se das vantagens das redes SDVN para fornecer uma abordagem adaptativa. Ao utilizar o LTE, em conjunto com a tecnologia WAVE (*Wireless Access in Vehicular Environments*), para transmissão de mensagens de *geocast*, permite-se que mais veículos estejam ao alcance do remetente, ao mesmo tempo em que o número de mensagens enviadas é reduzido.

O restante deste artigo está organizado como segue. A Seção 2 apresenta alguns 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

Aplicações de ITS podem requerer que uma mensagem seja entregue apenas para veículos situados em uma determinada região [Zhang et al. 2014]. A transmissão realizada para uma área específica é chamada de *Geocast* ou *Geobroadcast*. Neste âmbito, diferentes soluções foram propostas na última década.

Por exemplo, em [Bachir and Benslimane 2003] foi proposto um protocolo *geocast* chamado IVG (*Inter-Vehicular Geocast*). O propósito do IVG é informar veículos de uma região de risco, chamada de grupo *multicast*, sobre qualquer perigo em rodovias. O protocolo envia mensagens alertando os demais veículos da região de risco no momento em que o veículo sofre um acidente. Os veículos descartam a mensagem caso já tenham recebido. Caso contrário, eles esperam um tempo para retransmiti-la. O veículo com

menor tempo é escolhido e retransmite a mensagem.

Já em [Joshi et al. 2007] foi proposto o protocolo DRG (*Directed Robust Geocast*), que é baseado em um esquema que usa distância como suporte para realizar inundação (*flooding*) direcionada e restrita. No DRG, o veículo que origina o *broadcast* envia a mensagem para todos os vizinhos e, o tempo para retransmissão é calculado com base na distância do receptor ao remetente. Quanto maior a distância, menor o tempo. O veículo mais distante então retransmite a mensagem e, os outros veículos recebem a mensagem e cancelam sua retransmissão.

Alguns trabalhos 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]; soluções com baixo custo energético [Kumar et al. 2016].

Como a maioria dos protocolos são otimizados para cenários específicos [He et al. 2016], alguns autores sugerem o uso de SDN em conjunto com as redes veiculares [He et al. 2016] [Ku et al. 2014] [Kazmi et al. 2016]. As redes veiculares que utilizam SDN vem sendo chamadas de SDVN e, têm como principal atrativo, a sua adaptabilidade, permitindo que a rede se adeque facilmente à mudança de topologia. Nesses trabalhos são propostas arquiteturas e serviços para SDVN.

Por ser um tema relativamente novo, existem poucos trabalhos abordando estratégias de *geocast* em SDVN. Em [Liu et al. 2015], foi proposta uma arquitetura SDVN e um protocolo *geocast* que funciona utilizando um conjunto de Unidades de Acostamento (*Roadside Units* - RSUs). Destaca-se que o conceito de SDVN é utilizado apenas nas comunicações realizadas entre as infraestruturas externas, e não entre os veículos. O protocolo proposto neste trabalho, apresentado na Seção 3, independe da existência de RSUs. Até o momento da escrita deste artigo não foram encontrados outros trabalhos que estudam a disseminação de informações em SDVN.

3. Protocolo *Geocast* para Redes Veiculares Definidas por *Software*

Nesta seção é apresentada a arquitetura SDVN utilizada, bem como os detalhes da implementação do protocolo *geocast* proposto tanto nos veículos quanto no controlador SDVN.

3.1. Visão Geral da Arquitetura

A Figura 1 descreve a arquitetura utilizada. Nessa arquitetura é utilizado um **controlador SDVN** centralizado. Os veículos utilizam a arquitetura WAVE para realizar a troca de dados entre eles. A comunicação entre o plano de dados e o plano de controle é realizada por meio da rede celular (padrão LTE), na qual os veículos trocam informações com o controlador utilizando o protocolo *OpenFlow* [McKeown et al. 2008].

Nos cenários em que a quantidade de veículos sobrecarrega o controlador centralizado, como pode acontecer em grandes cidades, poderá ser utilizada uma implementação distribuída do controlador. Para isso é necessário utilizar um componente que informa qual controlador os veículos em uma região devem utilizar. Já nos casos em que a comunicação com o controlador for interrompida, pode ser utilizado algum outro protocolo *geocast* V2V para disseminar as informações.

De forma geral, o protocolo *OpenFlow* funciona da seguinte maneira. Cada *switch OpenFlow* possui um tabela de fluxos (*flow table*) que armazena qual ação deve ser realizada com cada fluxo de dados conhecido. Cada entrada da *flow table* (*flow entry*) contém um conjunto de campos de correspondência, que identificam o fluxo, e um campo de ação, que informa a ação que deve ser realizada ao receber um pacote daquele fluxo. Quando um *switch* recebe um pacote, é verificado se existe uma entrada na *flow table* correspondente a aquele pacote. Caso exista, a ação correspondente ao fluxo é realizada. Caso contrário, o *switch* envia para o controlador uma mensagem do tipo *Table-miss*. O controlador então responde informando a ação necessária para aquele tipo de fluxo. Ao receber a mensagem de resposta, o *switch* realiza a ação informada e adiciona esta informação na sua *flow table*. Assim, quando for necessário encaminhar outro pacote correspondente ao mesmo fluxo, o *switch* já saberá qual ação deverá realizar.

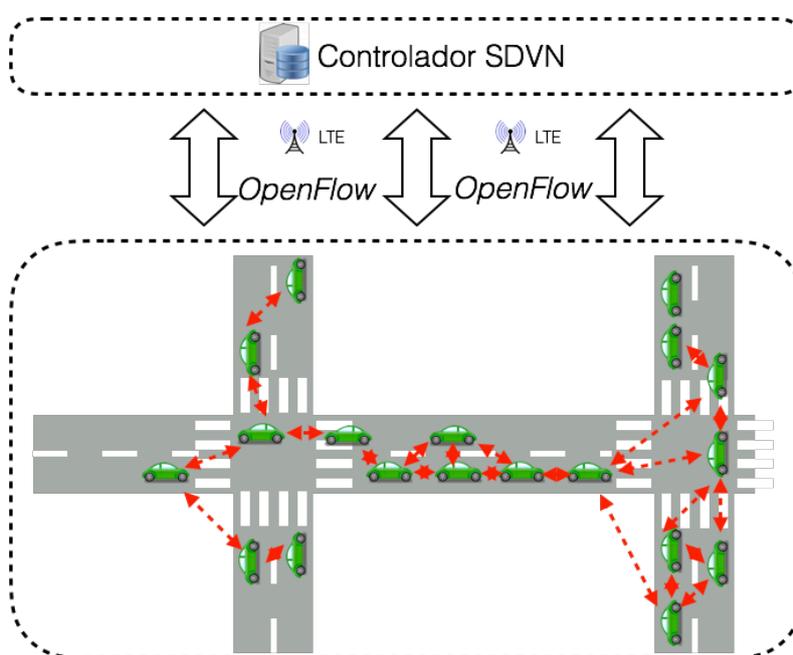


Figura 1. Arquitetura SDVN utilizada.

3.2. Implementação do Protocolo *Geocast* nos Veículos

Neste trabalho, cada veículo representa um *switch OpenFlow* e, com isso, possui uma *flow table*. A *flow table* possui dois campos de correspondência, o *Vehicle ID* e o *Geocast ID*. 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 desejar 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 a retransmitiram. Isso é necessário porque se algum dos vizinhos não retransmitir a mensagem, toda a disseminação pode ficar comprometida. Assim, o veículo que está encaminhando o pacote no momento esperará um tempo *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 enviará 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 esperará *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 pode esperar novamente por *Table-miss_Waiting_Time* segundos uma resposta. Esta espera acontece no máximo *Table-miss_Max_Waiting_Tries* vezes. Ou seja, o veículo esperará 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*, conterá 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*.

3.3. Controlador SDVN

O controlador SDVN é utilizado para informar aos veículos quais ações devem ser realizadas para cada fluxo de dados. A escolha das ações que cada veículo deve realizar para determinado fluxo de dados é feita da seguinte forma. Inicialmente, é necessário que o controlador possua informação global da topologia da rede veicular. Para isto, todos os veículos enviam *beacons* contendo suas posições geográficas obtidas por GPS ao controlador com uma determinada frequência (por exemplo, um *beacon* por segundo). O controlador mantém uma tabela *Vehicle_Table* com a posição de cada veículo. A Tabela 2 apresenta os campos presentes na *Vehicle_Table*.

<i>Vehicle ID</i>	<i>Position</i>	<i>Timestamp</i>
Veh_1	(15.3, 210.1, 0)	52.1
Veh_32	(258.0, 1330.1, 0)	52.5
Veh_12	(1530.8, 10.1, 0)	52.7
Veh_15	(72.1, 28.1, 0)	53.0

Tabela 2. Tabela *Vehicle_Table* armazenada no controlador com as posições dos veículos.

O campo *Vehicle ID* é o identificador único do veículo. O campo *Position* possui a posição presente no último *beacon* recebido do veículo representado por *Vehicle ID*. Já o *Timestamp* representa o momento em que foi recebido o último *beacon* do veículo *Vehicle ID*. Caso se passem *Vehicle_Entry_Duration* segundos desde o recebimento do último *beacon* 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. O *Guard_Range* é utilizado para minimizar os erros provocados por perdas de pacotes.

Sempre que o controlador recebe uma *Table-miss*, ele decidirá 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. Neste 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. Destaca-se que é realizada uma pequena modificação na heurística para que o veículo que criou a mensagem, a ser transmitida em *geocast*, sempre faça parte do MCDS calculado.

Os veículos escolhidos como pertencentes ao conjunto dominante conectado serão os veículos responsáveis por retransmitir a mensagem. Os demais veículos irão apenas descartar a mensagem.

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*. O fato de utilizar uma busca em profundidade no subgrafo formado pelos veículos contidos no MCDS faz com que o tamanho do conjunto *Next_Hop_IDs* de cada veículo seja pequeno.

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*.

Como discutido anteriormente, no momento em 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-lo, 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 detectará cada sub-região conectada dentro da região de interesse, e executará os mesmos passos utilizados quando a região é conectada para cada uma dessas sub-regiões conectadas. O controlador também escolherá um veículo de cada sub-região para agir como o veículo que originou a mensagem, ou seja, para começar 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 irá retransmiti-la para todos os *Region_Heads*, que, por sua vez, irão começar a disseminação da mensagem nas sub-regiões.

O controlador armazena em uma tabela todas as mensagens *Table-miss* recebi-

das nos últimos *Table-miss-Time* segundos. A cada *Update-Table-miss-Entry* segundos o controlador verificará 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.

4. Avaliação de Desempenho

Foi realizado um estudo de avaliação de desempenho por meio de simulações. O objetivo é verificar a eficácia do protocolo *geocast* em disseminar informações. As principais métricas de avaliação utilizadas são: (i) taxa de entrega de mensagens, (ii) *overhead* de sinalização e (iii) atraso na entrega de conteúdo.

4.1. Cenário

O desempenho do protocolo foi comparado com um algoritmo *geocast* simples de *Flooding* no qual todos os veículos na região de interesse retransmitem a mensagem uma vez. Apesar de não ser um protocolo de roteamento *geocast*, o algoritmo de *geocast Flooding* é útil para comparação com outros protocolos *geocast*. Isso se deve ao fato do mesmo ser utilizado como bloco de construção para diversos protocolos de roteamento [Maihofer 2004].

As ferramentas de simulação utilizadas foram o simulador de eventos discretos OMNeT++ 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.

Por meio do simulador de tráfego SUMO foi gerado um cenário do tipo *Manhattan Grid* 10×10 sob uma região de $5,0625 \text{ km}^2$. Cada segmento de via possui 250 metros de extensão e duas faixas em cada sentido. O limite de velocidade dos veículos é de aproximadamente 50 km/h. Esse tipo de mapa é comumente utilizado para avaliar protocolos de redes veiculares em ambientes urbanos [de Sousa and Soares 2015] [Milojevic and Rakocevic 2014] [Araújo et al. 2014] [Garip et al. 2015].

Para criar a demanda de tráfego necessária no mapa, foram geradas 3000 rotas aleatórias de tamanho mínimo de 1 km. As rotas foram criadas escolhendo pares aleatórios de vias origem e destino. Para cada par, foi escolhido como rota a sequência de vias com o menor caminho que ligam esse par.

A duração de cada simulação foi de 120 segundos. Foram implementadas instâncias deste cenário com quatro demandas de tráfego diferentes. Nessas instâncias, a quantidade de veículos foi mantida constante em 50, 100, 150 e 200, respectivamente. Cada instância foi replicada 33 vezes. 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 sumariza os principais parâmetros adotados nas simulações.

Parâmetro	Valor
Duração	120 segundos
Demanda de tráfego	50, 100, 150 e 200 veículos simultâneos
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 <i>bytes</i>
Tamanho dos dados de um <i>beacon</i>	17 <i>bytes</i>

Tabela 3. Parâmetros utilizados nas simulações do cenário *Manhattan Grid*.

Durante todas as simulações, um veículo foi escolhido para transmitir uma mensagem a cada 5 segundos. A região de destino da mensagem é uma região fixa com raio igual a 1500 metros. Toda a rota do veículo transmissor está contida nesta região. A Tabela 4 apresenta os valores das variáveis do protocolo *geocast* proposto utilizados nas simulações.

Parâmetro	Valor
<i>Flow-Entry_Lifetime</i>	120 segundos
<i>Max_Waiting_Time</i>	0.1 segundos
<i>Max_Tries</i>	3
<i>Table-miss_Waiting_Time</i>	0.1 segundos
<i>Table-miss_Max_Waiting_Tries</i>	3
<i>Table-miss_Max_Tries</i>	3
<i>Vehicle_Entry_Duration</i>	5 segundos
<i>Guard_Range</i>	20 metros

Tabela 4. Parâmetros utilizados das variáveis protocolo *geocast* proposto nas simulações.

4.2. Taxa de Entrega de Mensagens

A Figura 2 apresenta a taxa de entrega de mensagens. A taxa de entrega de mensagens refere-se à porcentagem de veículos presentes na região de interesse no momento do envio da mensagem que a receberam.

Observa-se que em todos os cenários avaliados o protocolo proposto apresenta uma taxa de entrega próxima a 100%. Isso ocorre devido às transmissões realizadas pelo controlador nos casos em que não é possível realizá-la utilizando a comunicação V2V. As perdas de pacotes, que ocorrem sobretudo em consequência das colisões, fazem com que a taxa de entrega não seja igual a 100%. Já o algoritmo *Flooding* apresenta uma taxa de entrega baixa nos cenários com baixa densidade de veículos e uma taxa alta nos cenários com maior densidade de veículos. Este fenômeno ocorre porque em cenários com poucos veículos, a rede formada pela comunicação V2V não é conectada, já em cenários com muitos veículos é formada uma rede conectada.

4.3. *Overhead* de Sinalização

A Figura 3 apresenta a quantidade média de pacotes enviados pelos veículos utilizando a comunicação V2V em cada um dos cenários avaliados.

Verifica-se que o algoritmo *Flooding* apresenta um crescimento linear acentuado no *overhead* de sinalização com o aumento da demanda de tráfego. Este comportamento acontece porque no algoritmo *Flooding* todos os veículos retransmitem a mensagem uma

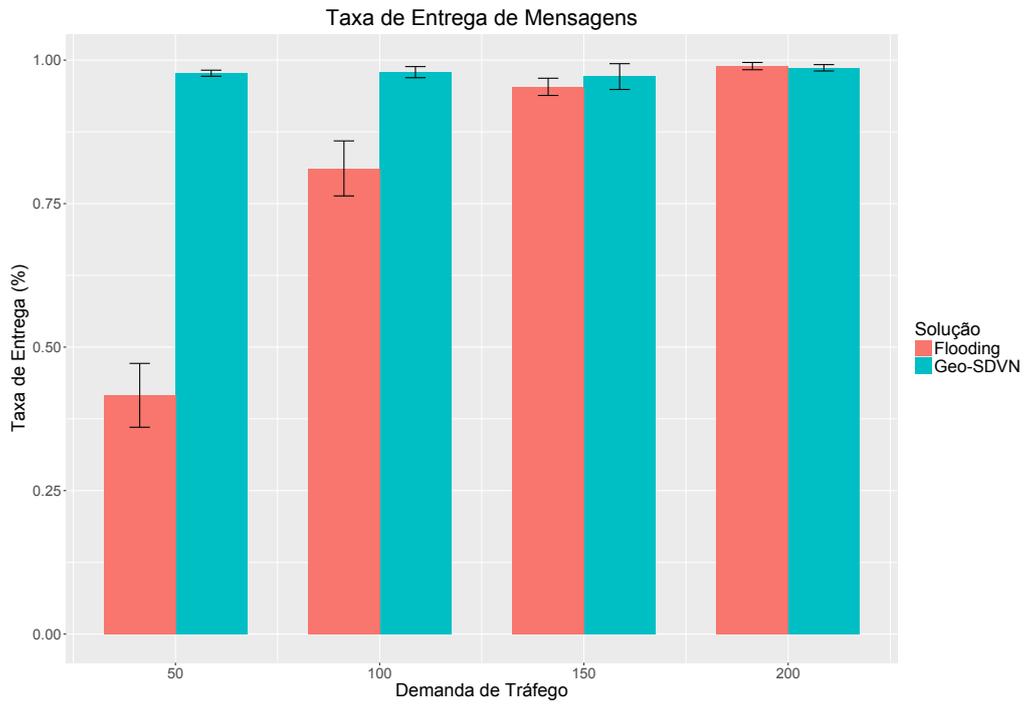


Figura 2. Taxa de entrega de mensagens.

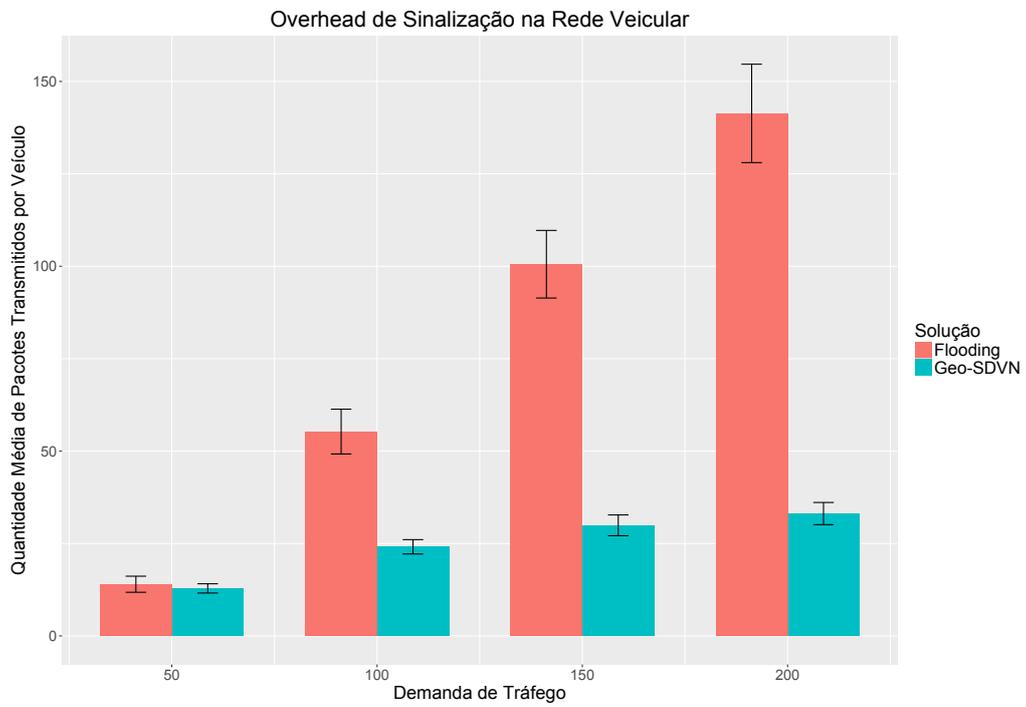


Figura 3. Overhead de sinalização.

vez, criando mensagens redundantes, o que resulta em uma tempestade de *broadcast*. Por outro lado, o protocolo proposto apresenta um baixo *overhead* de sinalização em todos os cenários. O principal motivo para este comportamento é o fato de apenas um subconjunto dos veículos da região de interesse serem escolhidos como nós retransmissores. Tal subconjunto é escolhido pelo controlador utilizado a heurística para calcular o conjunto dominante conectado “mínimo” de forma aproximada.

4.4. Atraso na Entrega de Mensagens

Por fim, a Figura 4 exibe o atraso médio na entrega das mensagens para os veículos da região de interesse.

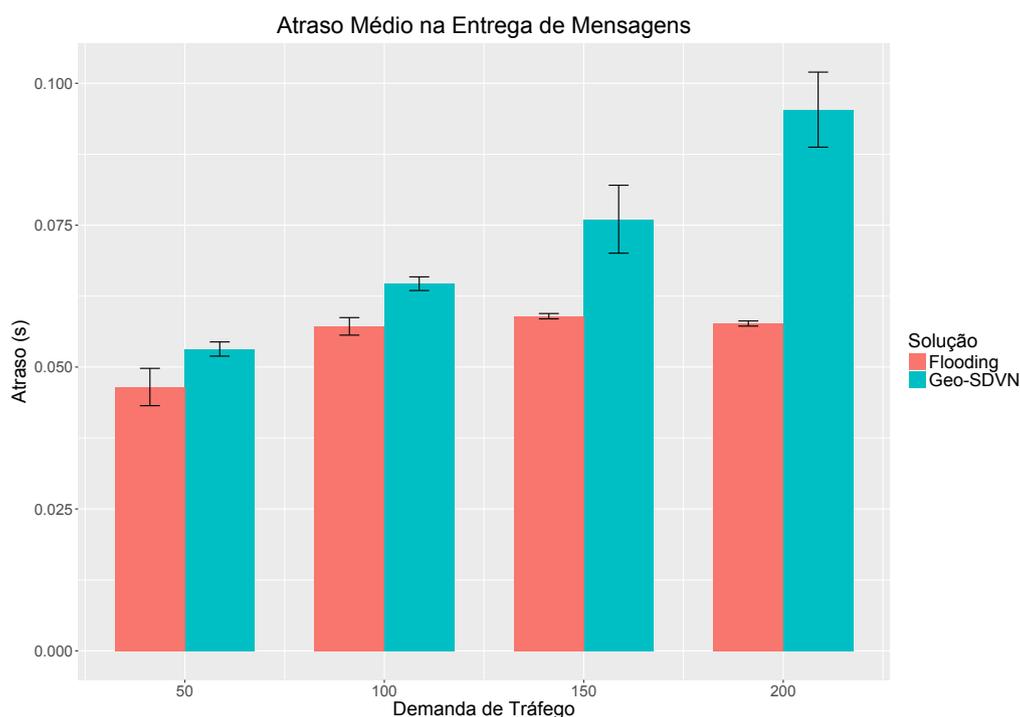


Figura 4. Atraso na entrega de mensagens.

Observa-se que, em todos os cenários, o protocolo proposto apresenta um atraso superior ao algoritmo de *Flooding*. Dois fatores contribuem para este fenômeno. Primeiro, a comunicação LTE possui um atraso maior que o da comunicação V2V [XiaoFang et al.]. Segundo, os mecanismos de recuperação utilizados pelo protocolo proposto quando são detectadas perdas de pacotes ocasionam em um aumento no atraso da entrega de mensagens.

5. Conclusões

Neste trabalho foi proposto um protocolo *geocast* para Redes Veiculares Definidas por *Software*. A arquitetura utilizada pelo protocolo dispensa a existência de unidades de acostamento, fazendo o uso apenas da comunicação V2V e da comunicação entre veículo e controlador por meio da rede LTE.

Foram avaliadas a taxa de entrega, o *overhead* de sinalização e o atraso médio na entrega de mensagens. Os resultados foram comparados com um algoritmo de *Flooding*.

Verificou-se que o protocolo proposto possui uma maior taxa de entrega de mensagens e um menor *overhead* de sinalização V2V nos cenários avaliados. O atraso na entrega foi ligeiramente superior ao do algoritmo *Flooding*, principalmente devido à comunicação LTE.

Como trabalhos futuros, pretende-se analisar o desempenho do protocolo em outros cenários e analisar o *overhead* de sinalização do controlador SDVN. Será ainda verificada a melhor configuração de valores para as variáveis do protocolo. Também pretende-se comparar o protocolo com outros protocolos *geocast* propostos para redes veiculares tradicionais.

Referências

- Araújo, G. B., Tostes, A. I. J., de L. P. Duarte-Figueiredo, F., and Loureiro, A. A. F. (2014). Um protocolo de identificação e minimização de congestionamentos de tráfego para redes veiculares. In *XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 207 – 220, Florianópolis.
- 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.
- 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. and Soares, A. C. B. (2015). Estimativa e sinalização de congestionamentos de tráfego através de redes veiculares V2V. In *XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Vitória.
- dos S. Alves, R., do V. Campell, I., and de S. Couto, R. (2009). Redes veiculares: Princípios, aplicações e desafios. In *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC 2009*, chapter 5, pages 199–254. Sociedade Brasileira de Computação.
- Garip, M. T., Gursoy, M. E., Reiher, P., and Gerla, M. (2015). Scalable reactive vehicle-to-vehicle congestion avoidance mechanism. In *12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 943–948.
- 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.
- 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.

- 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.
- Liu, Y. C., Chen, C., and Chakraborty, S. (2015). A software defined network architecture for geobroadcast in vanets. In *2015 IEEE International Conference on Communications (ICC)*, pages 6559–6564.
- Maihofer, C. (2004). A survey of geocast routing protocols. *IEEE Communications Surveys Tutorials*, 6(2):32–42.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- Milojevic, M. and Rakocevic, V. (2014). Distributed road traffic congestion quantification using cooperative vanets. In *13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pages 203–210.
- 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.
- OPEN NETWORK FOUNDATION (2013). Software-defined network (sdn) definition - open network foundation. <https://www.opennetworking.org/sdn-resources/sdn-definition>. acesso em: 26 de outubro de 2016.
- Rahbar, H., Naik, K., and Nayak, A. (2010). Dtsq: 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.
- 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.
- XiaoFang, L., Chi-Yuan, C., and Rouzbeh, R. Simulation comparison between lte and wi-fi in networks.

Zhang, H., Wang, R., and Larsson, T. (2014). Simulation of region-based geocast routing protocols. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 723–730.