

Um Método de Filtragem de Traços para Simulações Escaláveis de Redes Veiculares

Eric P. S. Oliveira e Miguel Elias M. Campista
GTA/PEE-COPPE/DEL-Poli - Universidade Federal do Rio de Janeiro (UFRJ)
Email: {ericpso,miguel}@gta.ufrj.br

Resumo—Traços com até milhares de nós são desejáveis em simulações de redes veiculares para que os resultados se aproximem da realidade. Porém, o uso desses traços em simuladores a eventos discretos como o ns-3 não escala. Este artigo propõe uma técnica de filtragem de traços baseada em algoritmo epidêmico. A ideia é reduzir o tamanho dos traços e assim aumentar a escalabilidade das simulações para estudos de VANETs (*Vehicular Ad Hoc Networks*). A filtragem proposta é feita de forma espaço-temporal com o intuito de incluir apenas veículos de interesse, aqueles que podem participar tanto da aplicação quanto do encaminhamento de pacotes. Simulações a processos estocásticos para redes veiculares conectadas ou não (*Delay and Disruption Tolerant Network - DTN*) demonstram que é possível obter resultados equivalentes a partir de filtragens do traço, as quais resultam em traços e tempos de simulação menores.

Index Terms—Filtragem de traço, Simulação de redes veiculares, Aumento da escalabilidade das simulações.

I. INTRODUÇÃO

A análise de protocolos de comunicação e aplicações de redes veiculares exige altos investimentos em infraestrutura, tornando praticamente inviável a realização de experimentos reais em larga escala [Fernandes e Ferreira, 2012], [Fernandes et al., 2012]. Devido a isso, as metodologias de análise mais comuns são a simulação com uso de traços sintéticos ou ainda experimentos em pequena escala [Uppoor et al., 2014]. Enquanto os experimentos em pequena escala continuam sofrendo com críticas sobre a relevância de casos particulares; as simulações oferecem maior grau de realismo usando traços gerados com base na movimentação habitual de veículos de cidades inteiras. Tais traços devem ser grandes o suficiente para capturar o comportamento tanto espacial, em termos de número de veículos e área da cidade; quanto temporal, em termos de duração do traço. Um exemplo de ferramenta amplamente utilizada para a geração de traços sintéticos é o Simulador de Mobilidade Urbana (*Simulation of Urban MOBility - SUMO* [Krajzewicz et al., 2012]), cujos traços são usados como entrada de simuladores a eventos discretos conhecidos como o ns-3 (*Network Simulator 3* [ns-3, 2018]) e o OMNeT++ (*Objective Modular Network Testbed* [OMNeT++, 2018]) na simulação de VANETs.

Um dos requisitos para que as simulações de redes veiculares sejam realistas é incluir um grande número de veículos. Porém, enquanto os traços sintéticos de movimentação podem ser gerados de forma eficiente por ferramentas ou até mesmo adquiridos a partir de repositórios, as simulações de redes veiculares encontram graves problemas de esca-

labilidade. Os simuladores de redes baseados em eventos discretos, como o próprio ns-3, alocam memória para todos os veículos desde o início da simulação, independente do momento de entrada ou de saída deste veículo do traço de mobilidade. Tal característica agrava um problema comum da implementação do canal sem fio que é a descoberta dos veículos receptores de um pacote. A descoberta é executada através de um laço de repetição que engloba todos os veículos da simulação. Como eventos de transmissão são numerosos, ainda mais com muitos veículos, a simulação enfrenta problemas de escalabilidade. Atualmente, na literatura, duas abordagens complementares atacam esse problema: a primeira visa aprimorar o funcionamento dos simuladores e a segunda visa reduzir o tamanho dos traços de mobilidade. Em relação aos simuladores, a ideia é desenvolver estruturas de dados mais eficientes, paralelizar o processamento ou reduzir o conjunto de veículos alcançáveis por transmissão de pacote [Fernandes e Ferreira, 2012]. Já a redução dos traços é uma etapa anterior à simulação, que visa remover os veículos que não interferem nos resultados pretendidos. A ferramenta `traceExporter.py` [SUMO Tools, 2018] do SUMO, p. ex., permite a redução do traço ao definir um tempo de início e de fim específico e uma área de interesse retangular menor que a área inteira coberta originalmente. A filtragem do `traceExporter.py`, porém, ainda resulta em traços grandes, principalmente quando as trajetórias dos veículos são longas. Dessa forma, estratégias mais sofisticadas para reduzir os traços de simulação sem impacto nos resultados finais ainda são necessárias, não só para traços do SUMO, mas para qualquer outro traço de mobilidade veicular.

Este trabalho propõe um método de filtragem espaço-temporal de traços de mobilidade com base em um algoritmo chamado de epidêmico e na definição de área e veículos de interesse, veículos rastreados e veículos infectados. Os veículos de interesse são fontes ou destinos de dados que não podem ser filtrados (retirados do traço) e cujas vizinhanças são de interesse à simulação. Já a área de interesse é a região geográfica relevante à aplicação ou ao protocolo simulado onde os veículos do traço final circulam. Os veículos rastreados são aqueles que possuem apenas suas entradas no traço final sem informação sobre seus vizinhos e os infectados são aqueles que, por fazerem o encaminhamento de pacotes, também se comportam como veículos de interesse (deseja-se saber possíveis interações com outros veículos), mesmo não sendo fonte ou destino de pacotes. A definição de veículos de interesse, rastreados e infectados tem por objetivo dar

suporte a uma filtragem que mantenha no traço apenas os veículos que interferiram direta ou indiretamente na rede. As definições também permitem que a filtragem englobe o maior número possível de casos de estudo, através de diferentes níveis de filtragem, incluindo casos onde não se conhece o veículo de destino ou de origem de um pacote. O método de filtragem deste trabalho tem por finalidade reduzir o tamanho e o número de nós de um traço de mobilidade, aumentando assim a escalabilidade da simulação. O único pré-requisito é que os traços de entrada sejam compostos por tuplas contendo o identificador do veículo e a sua posição a cada instante de tempo, o que garante o aspecto de generalidade da proposta. A filtragem proposta aborda ainda o problema de “saída e retorno” do traço que pode surgir em uma abordagem mais tradicional como a filtragem geográfica. A análise da proposta é feita através de simulações usando o ns-3, comparando o método de filtragem proposto com o usado pelo SUMO. Os resultados avaliam a redução do traço e o efeito da filtragem nos resultados da simulação. Os resultados mostram que a filtragem proposta consegue reduzir o traço sem afetar significativamente o resultado final da simulação realizada no ns-3. A filtragem também obteve sucesso em reduzir significativamente o tempo de simulação e contornar o problema de “saída e retorno” do traço de veículos em simulações ns-3 filtradas de forma tradicional.

A organização deste trabalho é a seguinte. A Seção II introduz as notações e definições usadas. Já a Seção III caracteriza os traços gerados pela filtragem padrão e os problemas que podem originar destes para simulações. O método de filtragem proposto é apresentado na Seção IV e os resultados obtidos a partir de seu uso são comentados na Seção V. Por fim, a Seção VI conclui este trabalho e apresenta direções futuras.

II. NOTAÇÕES E DEFINIÇÕES

Uma rede veicular é composta por nós móveis que, ao entrarem em contato em qualquer ponto da região simulada, podem trocar dados. Denota-se como \mathcal{V} o conjunto de nós da rede, t_0 e t_f os instantes de início e término da simulação e α a região simulada. Isso quer dizer que qualquer contato que ocorra em um ponto da região simulada possui um tempo de duração entre 0 e o tempo total de simulação, dependendo da trajetória e velocidade dos nós. Ainda, cada nó entra e sai da região simulada em momentos e pontos diferentes. Sendo assim, cada nó $v \in \mathcal{V}$ entra na simulação no instante de tempo t_e^v na posição (X_e^v, Y_e^v) e sai no instante t_s^v na posição (X_s^v, Y_s^v) , onde $t_0 \leq t_e^v \leq t_s^v \leq t_f$ e $(X_e^v, Y_e^v), (X_s^v, Y_s^v) \in \alpha$. No traço T , cada entrada é representada por uma tupla $\langle t, v, (X^v, Y^v) \rangle$, onde t é o instante de tempo, v é o identificador do nó e (X^v, Y^v) é a posição do nó. Dessa forma, o traço pode ser visto como uma lista de tuplas ordenadas em função do instante de tempo t . Caso se deseje listar apenas as tuplas referentes a um determinado instante de tempo t fixo, obtém-se a lista de tuplas $T(t) \subseteq T$. Ainda, cada nó v considera apenas como vizinho o nó que durante algum intervalo de tempo esteve em alcance de rádio mútuo. Logo, denota-se como

$\mathcal{N}(v) \subseteq \mathcal{V}$ o conjunto de nós vizinhos de v . As definições são utilizadas dependendo do nível de filtragem dos traços.

Definição 1 (Veículo de interesse). Veículo fonte ou destino de um pacote ou qualquer um que seja de interesse manter todas as entradas correspondentes no traço. Tais entradas são todas relacionadas tanto à posição do nó quanto às suas interações com nós vizinhos. Logo, para um veículo de interesse v_i , todas as entradas no traço referentes a v_i e sua vizinhança $\mathcal{N}(v_i)$ são mantidas no traço para cada instante de tempo. Caso haja mais de um veículo de interesse, $v_i, v'_i \in \mathcal{V}_i$, onde $\mathcal{V}_i \subset \mathcal{V}$ denota o conjunto de todos os veículos de interesse, todas as entradas referentes aos veículos em \mathcal{V}_i e suas vizinhanças são mantidas no traço.

Definição 2 (Veículo infectado). São todos aqueles que, após se tornarem vizinhos do nó de interesse v_i , desempenham o mesmo papel de um nó de interesse (como o caso de nós realizando encaminhamento de dados). Esses nós são chamados de veículos infectados para que seja possível diferenciá-los do veículo de interesse original. Portanto, considerando que v_s tenha se tornado vizinho de v_i , ele se torna um nó infectado. Ao encontrar outros nós vizinhos, p. ex., o nó v'_s , v_s se comporta como v_i , infectando o nó v'_s e assim recursivamente até um determinado limite superior de número de infecções ou número de saltos, chamado de ρ_{max} . Para monitorar o número na cadeia de infecções referentes a cada veículo infectado, é definido um conjunto \mathcal{J} contendo o número de saltos de infecção restantes de cada veículo infectado.

Semelhante ao encaminhamento epidêmico, chama-se de epidemia a infecção de veículos através de múltiplos saltos para a finalidade de filtragem do traço de mobilidade [Vahdat et al., 2000], [Spyropoulos et al., 2005], [Fall, 2003]. Denotando a união de todas as vizinhanças dos veículos infectados por $\mathcal{N}(\mathcal{V}_i)$ e considerando um veículo de interesse inicial v_i , o conjunto de nós infectados denotado por \mathcal{V}_s será igual ao conjunto de todos os veículos que em algum instante de tempo pertenceram a $\mathcal{N}(v_i)$ podendo ser expandido em seguida a qualquer veículo que tenha pertencido a $\mathcal{N}(v_i) \cup \mathcal{N}(\mathcal{V}_s)$ e assim sucessivamente contanto que não se atinja o limite de número de infecções. O conjunto de nós infectados pode ser estendido caso haja mais de um veículo de interesse. No caso dos veículos infectados, assim como em veículos de interesse, tanto as posições quanto as interações com os vizinhos são mantidas no traço.

Definição 3 (Veículo rastreado). Veículo que se deseja manter todas as entradas correspondentes no traço. Essa definição se aplica quando deseja-se saber a trajetória de um veículo que pode ser destino de um pacote ou mesmo interagir com um veículo de interesse ou infectado. Essa definição é importante para contornar o problema de “saída e retorno” do traço, como será explicado posteriormente. Considerando v_i como um veículo de interesse e v_s como um infectado, um veículo rastreado pode ser configurado como sendo todo aquele que, a partir do momento que entrar no conjunto de

vizinhos de v_i ($\mathcal{N}(v_i)$) ou v_s ($\mathcal{N}(v_s)$), passa a ter todas as suas entradas mantidas no traço. As entradas do veículo rastreado podem ser referentes ao instante de tempo anterior ou posterior ao instante de interação com v_i ou v_s . Isso é de interesse pois permite que veículos que interajam com o veículo de destino em um ponto futuro também tenham as suas posições anteriores rastreadas. Denotando a união de todas as vizinhanças dos veículos de interesse por $\mathcal{N}(\mathcal{V}_i)$, o conjunto de veículos rastreados \mathcal{V}_r seria formado por todos os veículos que, em algum momento do traço, pertenceram a $\mathcal{N}(\mathcal{V}_i)$ ou a $\mathcal{N}(\mathcal{V}_s)$. No caso do veículo rastreado, apenas as posições assumidas ao longo do traço são mantidas, ou seja, a vizinhança de um nó rastreado não é mantida.

Definição 4 (*Área de interesse*). Define-se como área de interesse α^{v_i} a região geográfica no traço relevante à aplicação ou protocolo simulado. Portanto, a área de interesse deve incluir a região por onde os veículos de interesse v_i ($i \in \mathbb{N}$) se deslocam. Sendo assim, considerando o ponto de entrada de v_i no traço como sendo $(X_e^{v_i}, Y_e^{v_i})$ e o de saída $(X_s^{v_i}, Y_s^{v_i})$. A área de interesse é uma figura geométrica que abrange todos os pontos da trajetória de v_i entre o ponto de entrada e de saída do traço de mobilidade. Independente da figura geométrica, a área de interesse α^{v_i} delimita os pontos de interesse que devem ser mantidos no traço. Tal região também pode considerar mais de um veículo de interesse, sendo assim, estendida.

A Figura 1 mostra o veículo de interesse v_i e o veículo rastreado v_r . A área de interesse α^{v_i} é definida a partir das posições de entrada $(X_e^{v_i}, Y_e^{v_i})$ e de saída $(X_s^{v_i}, Y_s^{v_i})$ de v_i . O veículo v_r , após entrar na vizinhança de v_i torna-se um veículo rastreado e, portanto, possui todas as suas posições armazenadas no traço, mesmo em caso de saída da área de interesse α^{v_i} . Já a Figura 2 apresenta a mesma área de interesse, porém com a participação dos veículos infectados. Um veículo v_i infecta o nó v_s que, por sua vez, infecta posteriormente o nó v'_s . A infecção ocorre quando o nó se torna vizinho de outro já infectado. Diferente do veículo rastreado, os veículos infectados e de interesse possuem também as vizinhanças mantidas no traço. As vizinhanças dos nós estão representadas na figura por círculos concêntricos aos nós. A aplicação dos diferentes veículos definidos na filtragem dos traços de mobilidade será vista na Seção IV.

III. DESAFIOS NO USO DE TRAÇOS REAIS

As simulações de redes sem fio que utilizam traços em escala real esbarram em dois grandes desafios: consumo de memória e tempo de processamento. Primeiro, todos os nós do traço de mobilidade são instanciados na simulação desde o início, independente dos seus respectivos instante de tempo de entrada e saída. Por exemplo, um nó v que entre em um instante de tempo $t_e^v \gg t_0$ e que saia em um instante de tempo $t_s^v \ll t_f$ é alocado em memória desde o início da simulação. Segundo, a implementação típica do canal sem fio faz com que, a cada evento de transmissão de pacote, um laço de repetição envolvendo todos os nós da simulação seja

executado. Como o número de nós é grande, tanto a memória quanto o tempo de processamento crescem rapidamente, tornando a escalabilidade um problema grave de simuladores de redes como o ns-2 e o ns-3 [Fernandes et al., 2012].

O processo de filtragem é anterior ao da simulação de rede. Após a geração do traço de mobilidade, a filtragem pode ser usada para aumentar a escalabilidade da simulação. As abordagens típicas para filtragem de traços de mobilidade são baseadas em cortes temporais ou espaciais da área de interesse. O corte espacial é feito em uma área retangular e o temporal é feito diminuindo o intervalo inicial e final do traço original. Dessa forma, o traço gerado poderia começar e terminar nos instantes t_0^{sim} e t_f^{sim} , onde $t_0 \leq t_0^{sim} \leq t_f^{sim} \leq t_f$, e poderia descrever a área de interesse retangular $\alpha^{sim} \subseteq \alpha$. Apesar de melhorar a escalabilidade da simulação, essa filtragem ainda é muito simples, especialmente quando inclui veículos com trajetórias não retilíneas. Esse tipo de trajetória leva a áreas retangulares maiores que, conseqüentemente, resulta em traços de mobilidade com um número maior de entradas referentes a veículos que não participam da simulação. As Figuras 1(a) e 2(a) ilustram o problema. Em ambas as figuras, os pontos $(X_s^{v_i}, Y_s^{v_i})$ não possuem nenhuma interação envolvendo o veículo de interesse, tornando-os irrelevantes em $t = t_0$. Além desse problema, o método de filtragem típico também desconsidera veículos que tenham saído e retornado à área de interesse, problema chamado neste trabalho de “saída e retorno” de veículo. Dado que os arquivos de mobilidade do ns-3 requerem que a trajetória de cada nó seja contínua (um nó move-se sempre em linha reta entre duas posições consecutivas na simulação), um nó que apresente parte de sua trajetória fora do traço é simulado como se tivesse se movido do ponto de saída da região de interesse α^{sim} diretamente ao seu ponto de retorno. Esse problema muda a trajetória do veículo e, assim, as interações com outros nós. Como a trajetória considerada é em linha reta, as interações não necessariamente são as mesmas comparadas às interações experimentadas ao longo da trajetória real. Para evitar esse problema, os métodos de filtragem excluem os veículos que retornam à área de interesse α^{sim} . Essa solução, porém, pode causar outro tipo de inconsistência, já que remove veículos que potencialmente participam da rede. Esse problema está ilustrado nas Figuras 1(b) e 2(b), onde, respectivamente, os veículos v_r e v_s saem e depois retornam à área de interesse, interagindo com o de interesse ao retornarem.

IV. FILTRAGEM PROPOSTA

A filtragem proposta complementa as filtragens espaço-temporais existentes, incluindo novas opções para a redução do traço de mobilidade. Além disso, a filtragem proposta lida com problemas decorrentes da utilização da filtragem típica, como é o caso do problema da “saída e retorno” mencionado. A ideia chave da filtragem proposta baseia-se na noção de que existe um conjunto de veículos cujas interações são relevantes à simulação. Sendo assim, o corte do traço que se deseja é aquele que mantém após a filtragem apenas os veículos que participam de interações relevantes. Para tal, os conceitos

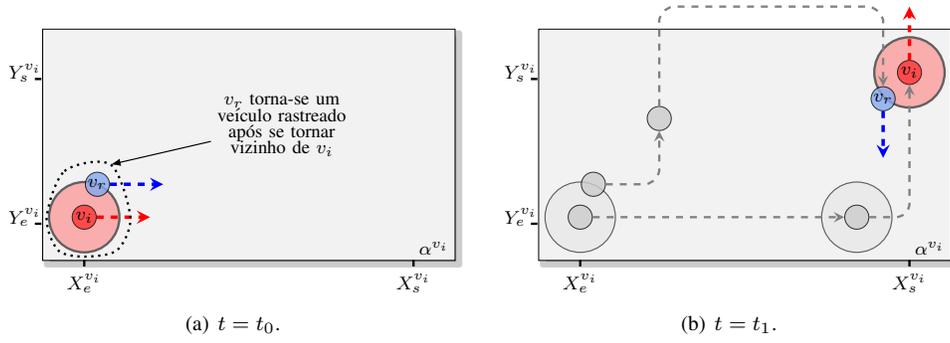


Figura 1. (a) O veículo v_r torna-se um veículo rastreado. (b) Após se distanciar de v_i , v_r se reaproxima em momento posterior.

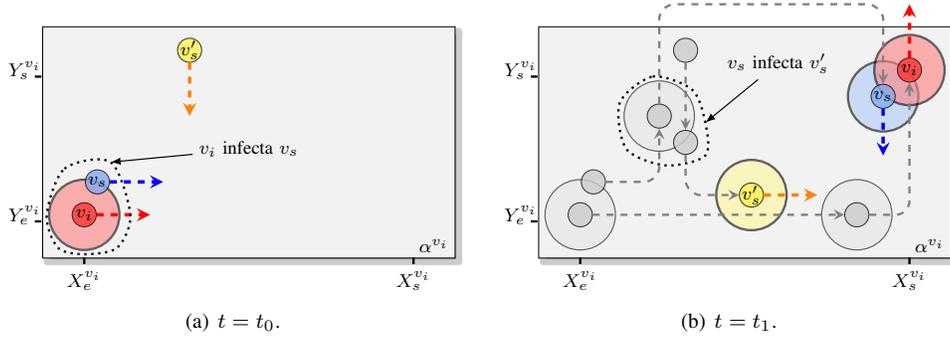


Figura 2. (a) O veículo v_s é infectado por v_i . (b) v_s e v'_s também se comportam como v_i , sendo v'_s infectado por v_s ao se encontrarem entre t_0 e t_1 .

expostos na Seção II são usados para criar diversas opções de filtragem. Tais opções podem ser usadas conjuntamente para a redução do traço final de acordo com o caso de uso ou a simulação desejada, por exemplo, se é uma rede DTN ou rede veicular de múltiplos saltos. Note que assim como na filtragem padrão, todas as opções de filtragem propostas recebem como entrada um traço de mobilidade e oferecem como produto um traço filtrado. Todas as opções requerem apenas que o traço de mobilidade possua entradas em formato de tuplas contendo a posição e a identificação do veículo a cada instante de tempo. Esse pré-requisito confere generalidade à proposta.

A. Opções de filtragem

Este trabalho adota como estratégia de filtragem a manutenção dos nós que de fato interagem com a simulação. Dessa forma, como a interação de um nó depende do caso de uso, define-se diferentes opções de filtragem com diferentes níveis de redução do traço de mobilidade.

O Algoritmo 1 descreve a estrutura geral da filtragem, onde classifica-se se a entrada referente a um veículo deve ser mantida ou retirada do traço de mobilidade da simulação. O traço deve manter apenas os veículos classificados ou como de interesse e respectivos vizinhos, ou como infectados e respectivos vizinhos ou como rastreados. Para tal, pelo menos uma das funções definidas para cada método de filtragem (Is_VI, Is_RV e Is_II), descritos a seguir, deve retornar verdadeiro. Note que dependendo do caso de uso, nem todos os tipos de veículos serão necessários. Caso algum tipo de

veículo não seja usado, basta retirar a sua função de filtragem do teste condicional da linha 5, através, p. ex., de uma *flag*.

1) **Filtragem por veículos de interesse:** Sendo as interações dos veículos de interesse (vide Definição 1) pertinentes ao objetivo da simulação, quaisquer veículos em suas vizinhanças são mantidos pela filtragem. Sendo assim, a filtragem é feita de forma espaço-temporal mantendo apenas os veículos de interesse e seus vizinhos a cada instante de tempo. Essa opção de filtragem é análoga a uma filtragem espacial para cada instante de tempo do traço de mobilidade, onde a área de interesse é equivalente ao raio de cobertura de cada veículo de interesse. Já a filtragem temporal é feita considerando os instantes iniciais e finais do traço de mobilidade como sendo iguais ao instante de entrada $t_e^{v_i}$ do primeiro veículo e o instante de saída $t_s^{v_i}$ do último veículo, dentre todos $v_i \in \mathcal{V}_i$ da simulação. No espaço, a área de interesse se torna igual à união de toda a área geográfica delimitada pelo raio de alcance de cada v_i ao longo do tempo. A diferença dessa filtragem em comparação à espaço-temporal típica é que a filtragem por veículos de interesse evita o desperdício de recursos com nós que não participam das interações entre os nós de interesse. A função Is_VI para filtragem de veículos de interesse e respectivas vizinhanças está descrita no Algoritmo 2.

2) **Filtragem por rastreamento de veículos:** O problema da “saída e retorno” de veículos da área de interesse, descrito na Seção III, é contornado nesta opção. Para tanto, os veículos rastreados definidos na Seção II são utilizados. Assim, após tornar-se vizinho de um veículo de interesse v_i , o veículo v_r

Algoritmo 1 Algoritmo da estrutura geral da filtragem.

Entrada: $\mathcal{V}_i, \mathcal{V}_s, \mathcal{V}_r, \rho_{max}, T$ **Resultado:** T filtrado

```
1:  $t = t_0; \mathcal{V}_s = \{\}; \mathcal{V}_r = \{\}; \mathcal{J} = \{\}$ 
2: enquanto  $t \neq t_f$  faça
3:    $T(t) \leftarrow \text{PEGA\_TRAÇO}(T, t)$ 
4:   para todo  $v$  em  $T(t)$ 
5:     se  $\text{IS\_VI}(v, \mathcal{V}_i, T(t))$  ou  $\text{IS\_RV}(v, \mathcal{V}_i, \&\mathcal{V}_r, \&\mathcal{V}_s, T(t))$  ou  $\text{IS\_II}(v, \mathcal{V}_i, \&\mathcal{V}_s, T(t), \&\mathcal{J}, \rho_{max})$  então
6:       senão
7:          $T(t) \leftarrow T(t) - \text{entrada de } v$ 
8:        $T \leftarrow \text{ATUALIZA\_TRAÇO}(T, T(t), t)$ 
9:        $t \leftarrow$  próximo instante de tempo
10: retorne  $T$ 
```

▷ Enquanto o traço não chega ao fim em t_f
▷ Filtra as entradas do traço T referentes ao instante t
▷ $T(t)$ permanece inalterado
▷ Atualiza $T(t)$ em T
▷ Retorna T filtrado

passa a ter todas as suas entradas no traço, mesmo que este saia da vizinhança de um veículo de interesse ou da área de interesse. Essas entradas são as posições ocupadas pelo veículo ao longo de sua trajetória. Esta opção de filtragem contorna o problema da “saída e retorno” de veículos, resultante da ausência de suporte a teleporte do ns-3, sem a adição de nós extras na simulação. A Figura 1 ilustra o uso desta opção de filtragem. Note que apenas as posições ocupadas pelo nó rastreado v_r são mantidas no traço. Em oposição, o nó de interesse v_i tem, além de suas posições, a sua vizinhança mantida no traço, assim como na opção de filtragem por veículos de interesse. A função IS_RV que identifica e atualiza os veículos rastreados é descrita no Algoritmo 3. Os veículos rastreados vão sendo adicionados ao conjunto \mathcal{V}_r , já que todos devem ser mantidos no traço final filtrado. Por isso mesmo, o conjunto de \mathcal{V}_r é passado por referência para a função IS_RV . Se um veículo rastreado não se encontra no instante de tempo sendo analisado $T(t)$, conclui-se que este chegou a seu objetivo e, portanto, pode ser retirado de \mathcal{V}_r para liberar memória (realizado nas linhas 7 a 9 do Algoritmo 3). Caso contrário, \mathcal{V}_r iria continuar aumentando conforme mais períodos de tempo fossem analisados. A liberação de memória é realizada de forma análoga para o conjunto de veículos infectados \mathcal{V}_s (linhas 7 a 9 do Algoritmo 4).

3) **Filtragem por infecção de interesse:** A partir dos identificadores dos veículos fonte de dados, o uso da filtragem por veículos de interesse leva em conta todas as transmissões de um salto a partir destes que possam ocorrer durante a simulação. Conhecendo os veículos destinos de pacotes, e configurando estes como veículos rastreados, é possível levar em conta todas as transmissões de até dois saltos com a filtragem por rastreamento de veículos. Entretanto, transmissões de dados envolvendo um número maior de saltos ou envolvendo veículos de destino desconhecidos requerem uma nova opção de filtragem além das introduzidas anteriormente. Isso ocorre pois os nós que realizam encaminhamento nem sempre se tornam vizinhos dos nós de origem ou destino.

Um algoritmo epidêmico é usado para considerar encaminhamentos por múltiplos saltos quando não se sabe o destino final das transmissões. Para isso, considera-se que veículos que entrem na vizinhança de um veículo de interesse podem infectar outros veículos, se comportando, assim, como veículos de interesse. A infecção é limitada por um número máximo

de saltos (ρ_{max}) a partir do veículo de interesse original pois, caso contrário, a filtragem envolveria muitos nós, reduzindo sua eficácia. O número de saltos restantes de infecção a partir de cada veículo infectado v_s é monitorado por um ρ_{v_s} correspondente e armazenado no conjunto de saltos \mathcal{J} . A ideia desta filtragem é garantir que qualquer veículo destinatário de uma transmissão originada no veículo de interesse esteja presente no traço final. A Figura 2 ilustra esta opção de filtragem. A Figura 2(a) mostra o veículo v_s se comportando como um veículo de interesse v_i , após entrar na vizinhança de v_i ($v_s \in \mathcal{N}(v_i)$). Já a Figura 2(b) mostra que o veículo v'_s também se comporta como um veículo de interesse, após o contato com v_s . Consequentemente, tanto as posições quanto as vizinhanças dos nós v_i, v_s e v'_s são mantidas no traço. A função IS_II que identifica os veículos infectados e controla a propagação da infecção de interesse é descrita no Algoritmo 4. Os veículos infectados vão sendo adicionados ao conjunto \mathcal{V}_s , já que todos devem ser mantidos no traço final filtrado. Por isso mesmo, o conjunto de \mathcal{V}_s é passado por referência para a função IS_II .

Algoritmo 2 Função para filtragem de veículos de interesse e respectivas vizinhanças.

```
1: função  $\text{IS\_VI}(v, \mathcal{V}_i, T(t))$ 
2:   se  $v \in \mathcal{V}_i$  ou  $v \in \mathcal{N}(\mathcal{V}_i)$  então
3:     retorne verdade
4:   senão
5:     retorne falso
```

Algoritmo 3 Função de identificação e atualização dos veículos rastreados.

```
1: função  $\text{IS\_RV}(v, \mathcal{V}_i, \&\mathcal{V}_r, \&\mathcal{V}_s, T(t))$ 
2:   se  $v \in \mathcal{V}_r$  então
3:     retorne verdade
4:   se  $v \in \mathcal{N}(\mathcal{V}_i)$  ou  $v \in \mathcal{N}(\mathcal{V}_s)$  então
5:      $\mathcal{V}_r \leftarrow \mathcal{V}_r + \{v\}$ 
6:     retorne verdade
7:    $\forall v \in \mathcal{V}_r$  faça
8:     se  $v \notin T(t)$  então
9:        $\mathcal{V}_r \leftarrow \mathcal{V}_r - \{v\}$ 
10:  retorne falso
```

▷ Atualiza-se \mathcal{V}_r
▷ Libera memória

4) **Filtragem por infecção de interesse retro-temporal:** A opção de Filtragem com infecção de interesse visa considerar todos os nós do traço que podem ser destinos de pacotes enviados por caminhos com múltiplos saltos, mesmo quando

Algoritmo 4 Função de identificação e atualização de infecção de interesse.

```
1: função IS_II( $v, \mathcal{V}_i, \&\mathcal{V}_s, T(t), \&\mathcal{J}, \rho_{max}$ )
2:   se  $v \in \mathcal{N}(\mathcal{V}_i)$  então
3:      $\rho_v \leftarrow \rho_{max} - 1$ 
4:      $\mathcal{J} \leftarrow \mathcal{J} + \{\rho_v\}$ 
5:      $\mathcal{V}_s \leftarrow \mathcal{V}_s + \{v\}$ 
6:     retorne verdade
7:    $\forall v \in \mathcal{V}_s$  faça
8:     se  $v \notin T(t)$  então
9:        $\mathcal{V}_s \leftarrow \mathcal{V}_s - \{v\}$ 
10:    se  $v \in \mathcal{N}(\mathcal{V}_s)$  então
11:       $\rho_v = 0$ 
12:      para todo  $v_s \in \mathcal{V}_s$  tal que  $v \in \mathcal{N}(\mathcal{V}_s)$  faça
13:        se  $GET\_SALTOS(v_s, \mathcal{J}) > \rho_v$  então
14:           $\rho_v \leftarrow GET\_SALTOS(v_s, \mathcal{J}) - 1$ 
15:        se  $\rho_v \neq 0$  então
16:           $\mathcal{J} \leftarrow \mathcal{J} + \{\rho_v\}$ 
17:           $\mathcal{V}_s \leftarrow \mathcal{V}_s + \{v\}$ 
18:        retorne verdadeiro
19:    senão
20:      retorne falso
```

▷ Atualiza-se o número de saltos restantes referentes a v
▷ Adiciona-se v ao conjunto de de veículos infectados

▷ Libera memória

▷ Recebe o número de saltos de infecção restantes referentes a v_s
▷ Atualiza-se o maior valor já obtido

▷ Atualiza-se o número de saltos restantes referentes a v
▷ Atualiza-se o conjunto de veículos infectados

os identificadores dos destinos são desconhecidos. De forma análoga, quando os nós de destino são conhecidos, mas os de origem não são, torna-se necessário fazer uma filtragem onde a infecção de interesse ocorra para trás no tempo. Os instantes de tempo do arquivo de traço são analisados no sentido inverso (retro-temporal) e a infecção de interesse é tratada de forma similar à infecção de interesse apresentada. O rastreamento de veículos é tratado da mesma forma: o arquivo de traço é analisado no sentido inverso e veículos v_r que entrem na vizinhança de um veículo de interesse v_i passam a ser rastreados ao longo do traço. De forma análoga ao Algoritmo 1, este algoritmo analisa o arquivo de traço no sentido inverso do tempo aplicando as mesmas funções de filtragem. Dessa forma, as únicas mudanças em relação ao Algoritmo 1 é a inicialização de $t = t_f$ (linha 1 do Algoritmo 1) e o decremento de t que recebe instante anterior ao atual (linha 9 do Algoritmo 1).

B. Discussão

Como já abordado anteriormente, o uso do algoritmo epidêmico corre o risco de cobrir grande parte do traço com apenas alguns saltos a partir do veículo inicial. Quando se tem conhecimento dos nós de origem e destino de transmissões, essa característica pode ser mitigada ao usar a filtragem epidêmica tradicional em conjunto com a retro-temporal. Realizando filtragem epidêmica a partir da fonte e do destino (no sentido inverso) é possível cobrir os caminhos de menor número de saltos de forma que a infecção a partir da origem e do destino possuam cada uma, apenas metade do número de saltos do caminho completo. Dessa forma, a metodologia proposta também permite a combinação de filtrações como o caso da filtragem epidêmica tradicional e retro-temporal.

A Tabela I lista as simulações ou casos de uso vislumbrados para cada método de filtragem proposto. Nesta tabela, **STD** se refere à forma tradicional de filtragem de traços através da seleção de uma área de interesse de forma retangular com corte no tempo que elimina qualquer veículo fora do

Tabela I
COMPARAÇÃO ENTRE AS OPÇÕES DE FILTRAGEM DE TRAÇOS.

Filtragem	Caso de Uso
STD	Simulações restritas com baixo número de veículos e região geográfica limitada.
VI	Simulações de VANETs sem conhecimento da trajetória dos veículos após encontro. Considera-se apenas um único encontro entre veículos do traço com um veículo de interesse (caso o simulador não suporte rotas descontínuas).
RV	Simulações de VANETs ou DTNs onde deseja-se analisar apenas transmissões por um número reduzido de saltos. Preserva a rota dos veículos no traço após encontro e permite múltiplos encontros entre veículos do traço e veículos de interesse. Boa fidelidade para a vizinhança imediata dos veículos de interesse.
II	Simulações de VANETs ou DTNs onde se tem conhecimento da origem de transmissões e deseja-se obter sua disseminação pela rede através de múltiplos saltos.
RT	Simulações de VANETs ou DTNs onde apenas se tem conhecimento do destino das transmissões e deseja-se encontrar suas possíveis origens.
II+RT	Simulações de VANETs ou DTNs onde se conhece os veículos de origem e destino de transmissões. Melhor compromisso entre alta fidelidade do traço e seu tamanho.

corte; **VI** à filtragem por veículos de interesse; **RV** à filtragem por veículos de interesse com rastreamento de veículos; **II** à filtragem com infecção de interesse a partir do veículo de interesse podendo apresentar veículos rastreados; **RT** à filtragem retro-temporal com infecção de interesse e podendo apresentar veículos rastreados; e por fim, **II+RT** refere-se à combinação das filtrações feitas pelos métodos **II** a partir dos veículos de interesse fontes de dados e **RT** a partir dos veículos de interesse consumidores de dados.

V. AVALIAÇÃO

O método de filtragem proposto é avaliado a partir do traço de mobilidade sintético de TAPAS Cologne [TAPAS Cologne, 2018] do SUMO, que representa o

fluxo de veículos na cidade de Colônia. O período total do traço de mobilidade original compreende das 6 às 8h da manhã com granularidade de 1 s entre entradas consecutivas no traço e tamanho de arquivo de 10,1 GB. Após o transiente inicial da simulação, são selecionados veículos em uma mesma região do traço para cumprirem o papel de clientes e servidores.

A simulação é realizada no ns-3 usando o arquivo de mobilidade gerado pelo SUMO a partir do traço. Cria-se um cenário determinístico com comunicações sem perdas em um raio ≤ 300 m entre os nós. Testes são realizados usando os protocolos de roteamento *spray and wait* de redes tolerantes a atrasos e desconexões (DTN) implementados por [Lakkakorpi e Ginzboorg, 2013] no ns-3. As redes DTN foram escolhidas em detrimento das redes VANETs conectadas, pois considera-se que os cenários com conexão intermitente em redes veiculares são ainda mais realistas. As transmissões são feitas com UDP, usando pacotes de 1 kB e modelo de propagação de velocidade constante. O nó servidor da simulação tenta realizar uma transmissão a cada um dos nós clientes a cada segundo. Caso um caminho entre o cliente e o servidor for encontrado, o cliente envia a mensagem; caso contrário, ele tenta novamente em um momento posterior.

Os resultados da simulação são comparados entre o uso de um traço filtrado de forma tradicional pelo programa `traceExporter.py` fornecido pelo SUMO e traços filtrados com o método proposto. A área de interesse considerada para a filtragem tradicional é aquela que contém todos os instantes de tempo em que ambos os veículos do par cliente-servidor estão presentes e que é limitada geograficamente por um retângulo que contenha toda a trajetória de ambos os nós. Nos resultados, a filtragem tradicional é referida por “**STD**”. Já a ferramenta proposta é avaliada a partir da opção de filtragem “**II+RT**” com apenas um salto de infecção e rastreamento de veículos pelos veículos de interesse e infectados a partir do servidor. Note que a filtragem considera veículos de interesse como sendo ou clientes, ou servidores, ou ambos. A vizinhança dos veículos de interesse para filtragem é definida como todos os veículos em um raio de ≤ 305 m ao redor dos veículos de interesse, cobrindo assim todas as suas possíveis transmissões (raio ≤ 300 m). Dessa forma, obtém-se um bom nível de filtragem que conseguirá abranger todos os caminhos de até 4 saltos entre o servidor e o cliente (servidor, veículo infectado pelo servidor, veículo rastreado a partir do veículo infectado pelo servidor, veículo infectado pelo cliente, cliente). As outras opções serão avaliadas em trabalhos futuros.

A. Resultados

As simulações exploraram 3 casos de estudos diferentes: uma simulação com um par cliente-servidor em uma área de interesse de 40km^2 , uma simulação de um servidor para dois clientes na mesma área, e uma simulação de um servidor para cinco clientes em uma área de 164km^2 . A filtragem tradicional demora 4 e 8 minutos respectivamente para ser realizada nas áreas de interesse de 40km^2 e 164km^2 enquanto o filtro proposto demora 2 e 15 horas para realizar sua filtragem. Apesar da filtragem proposta ser mais custosa, a vantagem

fica evidente na simulação, principalmente se o mesmo traço filtrado for usado mais de uma vez. Além disso, foi usado um script `awk` (acessível em www.gta.ufrj.br/FCDfilter), para realizar a filtragem. Espera-se que o tempo de filtragem possa ser reduzido com melhoras na eficiência do código ou usando outra linguagem de programação. A Figura 3 ilustra a relação entre o tempo da simulação e o tempo real gasto para a simular a rede veicular, excluindo o tempo de geração de traço filtrado.

A Figura 3 demonstra nos três casos de estudos avaliados que as simulações com o método de filtragem proposto convergem em tempo hábil. Todas as simulações que utilizaram a filtragem padrão foram interrompidas antes de terminarem por estarem ou usando uma quantidade muito grande de memória ou por não se ter expectativas de convergência em intervalos de semanas. Comparando as Figuras 3(a), 3(b) e 3(c) nota-se que, com o aumento do número de servidores, a situação fica mais crítica já que mais entradas devem ser mantidas no traço.

A Tabela II relaciona os dados de cada um dos três casos de estudo de acordo com o método de filtragem usado (**STD** ou **II+RT**). A tabela mostra o número de nós e o tamanho do arquivo de mobilidade gerado pelas duas filtrações para cada um dos casos simulados. Imediatamente é possível perceber a diferença em termos de número de veículos removidos e redução dos traços. Essas diferenças demonstram que boa parte do traço de mobilidade, dependendo do caso de uso, é gasta com entradas referentes a veículos que não participam ativamente da simulação. Ainda, devido às simulações pelo método de filtragem tradicional não convergirem mesmo após longos períodos de tempo, a comparação entre número de transmissões e entregas dentre os métodos de filtragem é feita com base em valores parciais das simulações. Comparando os resultados parciais entre o filtro **STD** e o proposto aos 331s de simulação do caso “1-to-1” (“1-to-1 STD” e “1-to-1 II+RT”), ambos possuem o mesmo número de transmissões. Se a mesma comparação for feita para o caso “1-to-2” aos 211s, novamente o mesmo número parcial de transmissões é realizado. Idem para o caso “1-to-5” no instante de simulação 4s. Essa equivalência nos resultados é importante pois demonstra que a filtragem não afeta a simulação.

Dado que as simulações a partir dos traços filtrados pelo método tradicional não convergiram, apenas foi possível realizar a comparação entre os resultados parciais do número de entregas do caso “1-to-1”. Note que o número de entregas é diferente para a filtragem **STD** e a filtragem **II+RT**. Esse resultado é efeito da correção proposta ao problema de “saída e retorno” do traço de mobilidade. A filtragem **STD** remove 437 veículos do traço de mobilidade ao saírem da área de interesse, algo que não acontece com o traço filtrado pelo método **II+RT**. Dentre as 35 entregas realizadas na simulação “1-to-1 STD” é possível identificar as 11 entregas feitas pela simulação “1-to-1 II+RT”. Entretanto, as entregas da simulação “1-to-1 II+RT” apresentam em média um atraso de entrega menor (7,23 segundos em comparação a 12,6 segundos) e um número de saltos até a entrega maior indicando que o protocolo *spray and wait* encontrou um caminho de menor atraso através de um dos veículos que foram retirados do modelo de mobilidade da

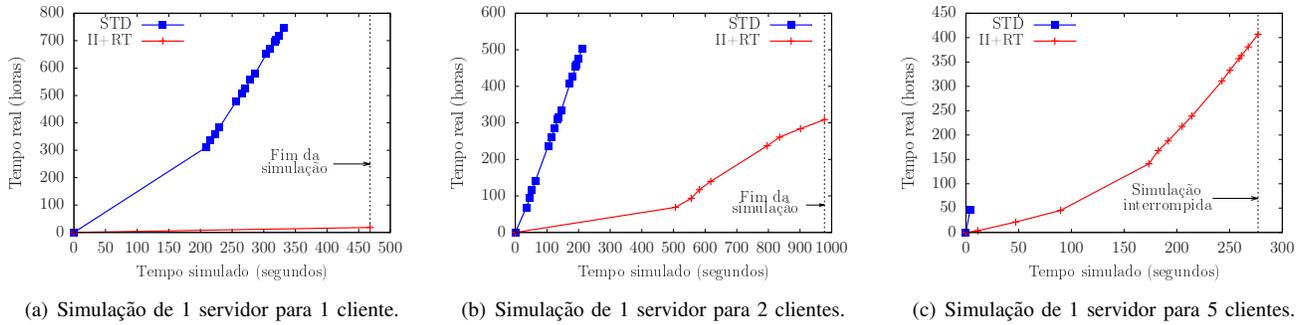


Figura 3. Relação entre o tempo simulado em função do tempo real para as simulações de rede DTN usando método de filtragem **II+RT**.

Tabela II
COMPARAÇÃO ENTRE FILTRAGENS PARA OS TRÊS CASOS DE ESTUDO DE DTN.

Filtragem	# de nós do arquivo	Tamanho	Tempo de simulação	Tempo total simulado	# parcial de transmissões	# parcial de entregas	# final de transmissões	# final de entregas
1-to-1 STD	2657	25,9MB	746h	331 s	332	11	332	11
1-to-1 II+RT	531	11,7MB	18h	468 s	332	35	468	35
1-to-2 STD	9.258	169,3MB	503h	211 s	422	0	422	0
1-to-2 II+RT	1.381	49,2MB	309h	977 s	422	0	1134	191
1-to-5 STD	34.358	1,1GB	47h	4 s	20	0	20	0
1-to-5 II+RT	3.952	218,6MB	407h	277 s	20	0	1014	14

simulação “1-to-1 STD”.

As últimas duas colunas da Tabela II apresentam o número final de transmissões e entregas dos casos estudados. Observe que o número é maior que os parciais apresentados apenas para os casos **II+RT**. Os casos com filtragem padrão não convergiram e, portanto, apresentam os mesmos resultados parciais. A diferença entre os resultados parciais e finais demonstraram a necessidade de simulações de longa duração para estudos em DTN, visto que o método de filtragem tradicional não conseguiu simular entregas que foram detectadas mais a frente na simulação pela filtragem **II+RT**. Os resultados deste artigo demonstram que o método de filtragem proposto reduz o tempo de simulação, sem perda de realismo, como desejado.

VI. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo propôs um método de filtragem de traços para redes móveis com o intuito de aumentar a escalabilidade das simulações de redes veiculares. Os resultados demonstraram que a ferramenta reduz o número de nós móveis no traço final, sem prejudicar os resultados da simulação. Além disso, a filtragem proposta supera problemas de filtragem por técnicas tradicionais conhecidas para modelos de mobilidade em ns-3. Futuramente, planeja-se tanto adaptar o código a uma linguagem além do `awk` e otimizá-lo por paralelização quanto aumentar o número de opções de filtragem.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES), Código de Financiamento 001; do CNPq; da FAPERJ; e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processos nº 15/24494-8 e 15/24490-2.

REFERÊNCIAS

- [Fall, 2003] Fall, K. (2003). A delay-tolerant network architecture for challenged internets. Em *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2003)*, p. 27–34.
- [Fernandes e Ferreira, 2012] Fernandes, R. e Ferreira, M. (2012). Scalable vanet simulations with ns-3. Em *IEEE VTC Spring*, p. 1–5.
- [Fernandes et al., 2012] Fernandes, R., Vieira, F. e Ferreira, M. (2012). VNS: an integrated framework for vehicular networks simulation. Em *IEEE VNC*, p. 195–202.
- [Krajzewicz et al., 2012] Krajzewicz, D., Erdmann, J., Behrisch, M. e 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.
- [Lakkakorpi e Ginzboorg, 2013] Lakkakorpi, J. e Ginzboorg, P. (2013). ns-3 module for routing and congestion control studies in mobile opportunistic DTNs. Em *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2013 International Symposium on*, p. 46–50. IEEE.
- [ns-3, 2018] ns-3 (2018). Network Simulator 3. [Online; acessado em 17 de julho 2018].
- [OMNeT++, 2018] OMNeT++ (2018). OMNeT++: Discrete event simulator. [Online; acessado em 17 de julho 2018].
- [Spyropoulos et al., 2005] Spyropoulos, T., Psounis, K. e Raghavendra, C. S. (2005). Spray and Wait: An efficient routing scheme for intermittently connected mobile networks. Em *ACM SIGCOMM Workshop on Delay-tolerant Networking (WDTN 2005)*, p. 252–259.
- [SUMO Tools, 2018] SUMO Tools (2018). TraceExporter. Acessado em <http://sumo.dlr.de/userdoc/Tools/TraceExporter.html>.
- [TAPAS Cologne, 2018] TAPAS Cologne (2018). Data/scenarios/tapascologne. Acessado em <http://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne>.
- [Uppoor et al., 2014] Uppoor, S., Trullols-Cruces, O., Fiore, M. e Barcelo-Ordinas, J. M. (2014). Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075.
- [Vahdat et al., 2000] Vahdat, A., Becker, D. et al. (2000). Epidemic routing for partially connected ad hoc networks.