



UM MÉTODO DE FILTRAGEM DE TRAÇOS DE MOBILIDADE BASEADO EM CONTATO PARA SIMULAÇÕES ESCALÁVEIS DE REDES VEICULARES

Eric Pereira Silva de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Miguel Elias Mitre Campista

Rio de Janeiro
Abril de 2019

UM MÉTODO DE FILTRAGEM DE TRAÇOS DE MOBILIDADE BASEADO
EM CONTATO PARA SIMULAÇÕES ESCALÁVEIS DE REDES VEICULARES

Eric Pereira Silva de Oliveira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA
ELÉTRICA.

Examinada por:

Prof. Miguel Elias Mitre Campista, D.Sc.

Prof. Luís Henrique Maciel Kosmowski Costa, Dr.

Prof. Igor Monteiro Moraes, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

ABRIL DE 2019

Oliveira, Eric Pereira Silva de

Um Método de Filtragem de Traços de Mobilidade Baseado em Contato para Simulações Escaláveis de Redes Veiculares/Eric Pereira Silva de Oliveira. – Rio de Janeiro: UFRJ/COPPE, 2019.

XIV, 51 p.: il.; 29, 7cm.

Orientador: Miguel Elias Mitre Campista

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2019.

Referências Bibliográficas: p. 47 – 49.

1. Filtragem. 2. Traços de Mobilidade. 3. VANET. 4. SUMO. 5. Algoritmo Epidêmico. I. Campista, Miguel Elias Mitre. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*“Todo progresso acontece fora da
zona de conforto”
– Michael John Bobak.*

Agradecimentos

Gostaria de agradecer, primeiramente, ao professor Miguel pela paciência e esforço gastos na minha orientação para a concretização dessa tese. Agradeço a todos os membros docentes da UFRJ que contribuíram à expansão dos meus conhecimentos e à minha formação. Aos pesquisadores Fernando da Cunha Luiz e José Eduardo Alves do Centro de Pesquisas em Energia Elétrica (Cepel) por me orientarem em meu estágio e à equipe do Cepel por me introduzirem em um ambiente de trabalho motivante. Agradeço aos meus amigos e colegas pelo suporte e apoio em minha jornada acadêmica. E, por fim, à minha família por me motivar e me tirar da minha zona de conforto, e por me ajudar a encontrar minha vocação.

Agradeço ao 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.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM MÉTODO DE FILTRAGEM DE TRAÇOS DE MOBILIDADE BASEADO EM CONTATO PARA SIMULAÇÕES ESCALÁVEIS DE REDES VEICULARES

Eric Pereira Silva de Oliveira

Abril/2019

Orientador: Miguel Elias Mitre Campista

Programa: Engenharia Elétrica

É comum se deparar com problemas de escalabilidade ao realizar simulações a eventos discretos de larga escala. Esse é um desafio conhecido por todos que estudam redes veiculares, onde traços podem ter até milhares de veículos. Como solução, este trabalho propõe uma técnica de filtragem de traços de mobilidade que leve em conta os contatos entre veículos. Contrária a técnicas de filtragem comuns, a filtragem proposta é feita de forma espaço-temporal com o intuito de manter apenas veículos de interesse para simulação, ou seja, aqueles que participam como origem ou destino de transferência de dados ou aqueles que encaminham pacotes. Um problema identificado na filtragem tradicional que gera a eliminação não intencionada de traços é abordado pelo método de filtragem proposto. Um filtro com base na filtragem proposta é implementado no trabalho. O filtro usa uma estrutura modular, o que facilita seu aperfeiçoamento e manutenção futuros, e garante a sua simplicidade. O código da ferramenta usada é disponibilizado e descrito ao longo do trabalho. Simulações de redes veiculares demonstram que os resultados alcançados com traços filtrados pelo método proposto convergem bem mais rápido e apresentam um consumo de memória reduzido comparado àqueles obtidos sem a proposta. Adicionalmente, mostra-se que é possível entregar um número maior de mensagens quando a técnica proposta é aplicada. Resultado esse que indica que a técnica de filtragem proposta mantém os traços de veículos que participam do roteamento de mensagens de forma melhor quando comparada à técnica tradicional de filtragem.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A CONTACT-BASED MOBILITY TRACE FILTERING METHOD FOR SCALABLE VEHICLE NETWORK SIMULATIONS

Eric Pereira Silva de Oliveira

April/2019

Advisor: Miguel Elias Mitre Campista

Department: Electrical Engineering

It is common to face scalability issues when running large-scale discrete-event simulations. This is a known challenge to everyone studying vehicular networks, where traces can have up to thousands of vehicles. As a solution, this work proposes a filtering technique for mobility traces that takes into account the contacts between vehicles. In opposition to common filtering techniques, the proposed filtering is done in a space-time manner in order to keep only vehicles of interest in the simulation, i.e., those participating either as source or destination of data transfers or those forwarding packets. A problem identified in traditional filtering that unintentionally deletes traces is addressed by the proposed filtering method. A filter based on the proposed filtering is implemented in this work. The filter uses a modular structure in its implementation, which facilitates its future improvement and maintenance, and ensures its simplicity. The code for the tool used is made available and described throughout the text. Vehicular network simulations show that the results achieved with traces filtered using the proposed method converge much faster and exhibit reduced memory consumption compared to those obtained without the proposed filtering. Moreover, we show that it is possible to deliver a larger number of messages when the proposed technique is applied. This result indicates that the proposed filtering technique is better at maintaining the vehicle traces that participate in message routing when compared to the traditional technique.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Lista de Símbolos	xii
Lista de Abreviaturas	xiv
1 Introdução	1
1.1 Filtragem proposta	3
1.2 Organização	5
2 Trabalhos Relacionados	7
2.1 Propostas para aprimoramento de simuladores	7
2.2 Propostas de otimização do uso de traços	9
3 Filtragem Proposta	11
3.1 Notações e definições	12
3.1.1 Membros do traço de mobilidade	13
3.2 Desafios na filtragem e uso de traços em simulações de VANETs	16
3.3 Módulos de filtragem e funcionamento da proposta	19
3.3.1 Algoritmo de filtragem	20
3.3.2 Filtragem por área de interesse	23
3.3.3 Filtragem por veículos de interesse	24
3.3.4 Filtragem por rastreamento de veículos	25
3.3.5 Filtragem por infecção de interesse	26
3.3.6 Filtragem por infecção de interesse retro-temporal	28
3.4 Discussão sobre o uso dos módulos de filtragem	29
3.5 Filtro implementado	31
4 Simulação	34
4.1 Cenários de filtragem avaliados	35
4.2 Análise dos resultados	38
4.2.1 Simulações em grande escala	38
4.2.2 Simulações em pequena escala	40

5	Conclusão	45
5.1	Trabalhos futuros	46
	Referências Bibliográficas	47
A	Uso do Filtro Implementado	50

Lista de Figuras

3.1	Fluxo da simulação de Redes Veiculares usado no trabalho.	11
3.2	(a) O veículo v_r torna-se um veículo rastreado. (b) Após se distanciar de v_i , v_r se reaproxima em momento posterior.	16
3.3	(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	17
3.4	Comparação de trajetórias do traço real (laranja) e filtrado (cinza) quando o problema de saída e retorno se apresenta.	18
3.5	Configuração dos módulos de filtragem.	21
3.6	Representação da infecção por dupla extremidade para amenizar a cobertura de algoritmo epidêmico.	30
4.1	Gráfico do grau dos veículos do cenário de Colônia codificados por cor. <i>Fonte:</i> [26]	35
4.2	Relação entre o tempo simulado em função do tempo real para as simulações de rede DTN em larga escala usando método de filtragem II+RT	39
4.3	Comparação temporal entre simulações resultantes do uso de traços com limites fixos ou flexíveis para a área de interesse.	41
4.4	Comparação temporal entre simulações de traços resultantes de diferentes modos de filtragem em pequena escala.	42

Lista de Tabelas

3.1	Sumário das notações usadas.	13
3.2	Comparação entre os módulos de filtragem de traços.	31
4.1	Comparação entre filtragens para os três casos de estudo de DTN em grande escala.	40
4.2	Comparação entre filtragens para os três casos de estudo de DTN em pequena escala.	41
4.3	Comparação entre filtragens para os três casos de estudo de DTN em pequena escala.	43

Lista de Símbolos

T	Traço ao qual se aplica a filtragem
$T(t)$	Lista de tuplas do traço T referentes ao instante de tempo t
d	Raio de alcance de transmissão considerado para a filtragem
t	Instante de tempo específico da simulação
t_0	Tempo inicial da simulação de rede
t_f	Tempo final da simulação de rede
v	Veículo pertencente ao traço T
v_i	Veículo de Interesse
v_r	Veículo Rastreado
v_s	Veículo Infectado
\mathcal{V}	Conjunto de veículos v pertencentes ao traço T
\mathcal{V}_s	Conjunto de veículos v_s pertencentes ao traço T
\mathcal{V}_i	Conjunto de veículos v_i pertencentes ao traço T
\mathcal{V}_r	Conjunto de veículos v_r pertencentes ao traço T
\mathcal{V}_s^{rt}	Conjunto de veículos v_s pertencentes ao traço T que foram infectados retro-temporalmente
\mathcal{A}	Conjunto de veículos que deixaram a área de interesse e passaram a ser rastreados para caso eles voltem
\mathcal{J}	Conjunto contendo o número de saltos de infecção restantes referentes a cada veículo infectado de forma direta
\mathcal{J}^{rt}	Conjunto contendo o número de saltos de infecção restantes referentes a cada veículo infectado retro-temporalmente
$\mathcal{N}(\mathcal{V})$	Conjunto de veículos pertencentes à união de todas as vizinhanças de raio d dos veículos v pertencentes a \mathcal{V}

$\mathcal{N}(v)$ Conjunto de veículos pertencentes à vizinhança de raio d de um veículo v

Lista de Abreviaturas

COPPE	Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia
DTN	Delay Tolerant Networks
SUMO	Simulation of Urban MObility
VANET	Veicular Ad Hoc Network

Capítulo 1

Introdução

Assiste-se hoje um crescente grau de interconectividade em diversos setores da indústria. No setor automobilístico, por exemplo, pesquisas em redes de carros interconectados e autônomos vêm usufruindo de crescentes investimentos [1]. As previsões apontam que veículos autônomos terão uma crescente demanda no futuro [2]. Como consequência, a área de Sistemas de Transportes Inteligentes (*Intelligent Transportation Systems* - ITS) vem apresentando um rápido passo de desenvolvimento, na qual fabricantes de carros buscam superar a concorrência oferecendo diferencial em suas aplicações. Um desses diferenciais é a aplicação de técnicas de inteligência artificial em aplicações de segurança no trânsito [3].

Em redes veiculares, as comunicações de veículo para veículo (*Vehicle-to-Vehicle* - V2V) e veículo para infraestrutura (*Vehicle-to-Infrastructure* - V2I) cumprem um papel importante na aquisição de inteligência. Os carros inteligentes devem ser capazes de obter informações atualizadas de seu ambiente e de outros veículos em sua vizinhança. O consumo de combustível, tempos de viagens, e riscos de acidentes podem ser todos reduzidos a partir de informações colhidas através de recursos próprios ou da comunicação com outros veículos. Nessa direção, há hoje extensa pesquisa em protocolos de comunicação para redes veiculares. Padrões como o IEEE 802.11p e o IEEE 1609 [4, 5] já vêm sendo consolidados na indústria como protocolos para transferência de dados.

A análise de protocolos de comunicação e suas aplicações em redes veiculares reais de larga escala exige contudo altos investimentos em infraestrutura, o que torna sua experimentação inviável [6, 7]. Portanto, simulações que usam traços sintéticos de mobilidade de veículos, ou ainda experimentos em pequena escala, são as metodologias de análise mais comuns ao se tratar do estudo de VANETs (*Veicular Ad Hoc Networks*) [8]. Enquanto os experimentos em pequena escala sofrem de escopo limitado, as abordagens analíticas se tornam inviáveis devido à sua complexidade. Como consequência, as simulações se destacam como uma metodologia efetiva de análise que viabiliza a generalização das avaliações de protocolos de rede e aplicações inteligentes. As simulações podem reproduzir cenários complexos de fluxos de veículos que seriam de difícil avaliação em plataformas de testes reais.

A análise de desempenho das redes veiculares requer a simulação da movimentação

ção dos veículos. Idealmente, procura-se usar a mobilidade de veículos reais para que a simulação possua o maior grau de fidelidade possível. Porém, devido à dificuldade de se obter traços de mobilidade que abranjam um número considerável de veículos de uma cidade, traços sintéticos são normalmente usados para simulações de larga escala. Esses traços são gerados com base em análises estatísticas do movimento habitual de veículos e devem capturar tanto as dinâmicas macroscópicas quanto microscópicas ligadas à movimentação de veículos em uma cidade [8, 9]. Além disso, os traços sintéticos permitem uma maior liberdade para alteração das suas características, possibilitando a criação de traços grandes o suficiente para capturar o comportamento tanto espacial quanto temporal da análise realizada. Enquanto em termos espaciais, a simulação consegue escalar o número de veículos em uma determinada área de uma cidade; em termos temporais, a simulação torna-se capaz de aumentar o tempo de duração e assim atender os requisitos de realismo da simulação a ser executada. Um exemplo de ferramenta amplamente utilizada para a geração de traços sintéticos é o Simulador de Mobilidade Urbana (*Simulation of Urban MObility* - SUMO [10]), considerado um dos mais completos e confiáveis geradores de mobilidade veicular de código aberto [8]. Os traços gerados pelo SUMO são usados como entrada de simuladores a eventos discretos conhecidos como o ns-3 (*Network Simulator 3* [11]) e o OMNeT++ (*Objective Modular Network Testbed* [12]) na simulação de VANETs.

A manutenção de um alto grau de realismo na simulação de redes veiculares requer, muitas vezes, um número elevado de veículos. Porém, enquanto os traços de movimentação dos veículos 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 escalabilidade. Em simulações de VANETs, é necessário manter em memória o estado de cada veículo e de suas conexões; e para cada transmissão, simuladores a eventos discretos como o ns-3 devem agendar um evento de recepção para cada nó através de um laço que cobre todos os nós da rede [6]. Logo, para N veículos transmissores e receptores de pacotes, é necessário computar $\mathcal{O}(N)$ interações a cada transmissão, já que para cada uma é necessário conhecer quais dos N nós recebem o pacote. O consumo de memória cresce linearmente com cada novo veículo adicionado à simulação. Enquanto isso, o número de ciclos computacionais gastos para a descoberta de receptores, onde todos os nós da rede são transmissores, cresce de forma quadrática [6]. Tais características tornam o consumo de recursos computacionais progressivamente maiores para cada novo veículo adicionado, reduzindo a escalabilidade das simulações.

1.1 Filtragem proposta

A interconectividade dos nós é altamente relacionada com os padrões de mobilidade em redes veiculares. Assim, qualquer método desenvolvido para reduzir ou simplificar a interação entre nós através da filtragem de seus traços de mobilidade corre o risco de levar a resultados errôneos [13]. Contrapondo-se aos métodos de filtragem usuais de traços através de cortes espaciais e temporais ou subamostragem ao longo do tempo, este trabalho propõe um método de filtragem de traços de mobilidade que leva em consideração sua característica espaço-temporal. Ao analisar o contato entre os membros móveis do traço, a filtragem proposta visa limitar o número de nós que devem ser simulados a cada instante da simulação. Essa abordagem é inspirada nos métodos de otimização em tempo de simulação existentes na literatura (discutidos no Capítulo 2), nos quais busca-se limitar o número de veículos que consomem recursos computacionais de acordo com sua capacidade de interação com outros veículos.

O método de filtragem proposto toma como base o contato entre veículos do traço. Assumindo aplicações de rede em que apenas uma fração dos veículos da rede são originadores e receptores de pacotes, os veículos restantes podem ser filtrados de acordo com sua utilidade como roteadores para a simulação da rede. Como apenas veículos que tenham entrado no alcance de veículos originadores ou encaminhadores de pacotes podem realizar roteamento, um filtro pode ser desenvolvido levando em conta essa dinâmica dos traços para fornecer ao simulador um traço apenas com veículos úteis à aplicação sendo simulada. Essa abordagem assemelha-se em comportamento a protocolos de encaminhamento oportunístico de redes tolerantes a atrasos (*Delay Tolerant Networks* - DTN) [14–16], no qual os veículos que podem participar no roteamento da rede são determinados pelo contato momentâneo entre nós que carregam pacotes da aplicação da rede e possíveis nós roteadores. A diferença, porém, é que neste trabalho esta ideia é usada com a finalidade de filtragem. Ao gerar um traço filtrado com um número menor de entradas e número de nós, busca-se mitigar os problemas de escalabilidade decorrentes de traços de grande tamanho. Apesar deste trabalho centrar-se na filtragem de traços de mobilidade de veículos para simulações escaláveis de VANETs, este é genérico o suficiente para ser aplicado a qualquer traço de mobilidade para fins similares e pode ser usado em conjunto com as técnicas de otimização de simulação já existentes. O único pré-requisito para a aplicação dessa proposta é a de que os traços de entrada sejam compostos por tuplas contendo o identificador do veículo e a sua posição, e que estes sejam ordenados segundo os instantes de tempo a que se referem. Tal requisito garante o aspecto de generalidade da proposta.

Pra auxiliar na filtragem espaço-temporal proposta, componentes do traço são

definidos, como área e veículos de interesse, veículos rastreados, e veículos infectados. A área de interesse é a região geográfica relevante à aplicação ou ao protocolo de rede simulado, onde todos os veículos do traço após a filtragem circulam. Veículos de interesse são veículos fonte ou destino de dados cujas interações com suas vizinhanças são de interesse à simulação. Já os veículos infectados, são aqueles que são julgados de possível importância para o encaminhamento de pacotes. Esses também se comportam como veículos de interesse pois deseja-se saber suas possíveis interações com seus veículos vizinhos, mesmo estes não sendo fontes ou destinos de pacotes. Como ambos os veículos de interesse e infectados são essenciais ao funcionamento da aplicação de rede, esses não podem ser filtrados (retirados do traço). Os veículos rastreados são definidos para dar suporte à filtragem e são tratados como veículos que possuem suas entradas no traço final sem exigir que as entradas de seus vizinhos também sejam salvas. A função principal dos veículos rastreados é evitar o problema de descontinuidade de trajetórias, existentes em abordagem mais tradicionais como a filtragem geográfica. Tal problema é definido neste trabalho como problema de “saída e retorno” e será descrito ao longo do trabalho. 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 interfiram direta ou indiretamente na aplicação de rede sendo simulada.

Além disso, o modelo de filtragem introduzido é construído de forma modular onde cada módulo ocupa-se de acompanhar cada um dos diferentes conjuntos de veículos previamente definidos. Uma estrutura modular permite fácil manutenção e expansão do filtro proposto visando perspectivas de desenvolvimento futuro. A separação da proposta em diferentes módulos também permite que a filtragem englobe o maior número possível de casos de estudo diferentes. Focando cada módulo, e o veículo tratado por este, para satisfazer determinada aplicação, é possível atender diversos casos de estudo pela combinação de diferentes módulos.

As filtrações, aqui apresentadas, solucionam problemas de realismo em regiões limite, e, portanto, não sofrem de erros referentes a alteração de trajetórias de veículos como as demais técnicas de filtragem. Além de não precisar se preocupar com problemas de subamostragem e compressão de traços com perdas. Porém, é possível constatar, através de sua descrição de funcionamento, que esta é voltada para um caso de estudo ideal onde exista um conjunto reduzido, em relação ao tamanho da rede, de veículos transmissores ou receptores a partir dos quais a filtragem por contato possa ser realizada. Para redes densas, onde há muito contato entre veículos, ou o número de veículos transmissores e receptores é grande em relação ao tamanho da rede, a filtragem proposta perde escalabilidade. A eficácia da proposta atinge seu valor mínimo quando todos os membros da rede podem ser considerados como transmissores ou receptores de dados. Porém, para esses casos, a proposta ainda

pode apresentar serventia por sua soluç o ao problema de trajet rias descont nuas (problema de “sa da e retorno”).

A an lise da proposta   feita usando traços sint ticos gerados pelo cen rio “TAPAS Cologne” [9] do SUMO que s o filtrados e usados em simulaç es de rede do ns-3. A comparaç o   feita quando o traço   filtrado por diversas combinaç es de m dulos do modelo de filtragem proposto, e pela ferramenta `traceExporter.py` do SUMO que aplica t cnicas convencionais de filtragem geogr fica. Os resultados avaliam a reduç o do tamanho do traço, consumo de mem ria, tempo de simulaç o e o efeito da filtragem na entrega bem sucedida de pacotes durante a simulaç o. Os estudos s o realizados em cen rios em grande e pequena escala. Os resultados em grande escala mostram que   poss vel reduzir significativamente os tempos de simulaç o e tamanhos dos traços, al m de contornar o problema de “sa da e retorno” para simulaç es no ns-3. As simulaç es em pequena escala fornecem resultados mais detalhados quanto ao consumo de mem ria, tempo de simulaç o e capacidade de entrega de dados para diversas configuraç es da filtragem proposta. Seus resultados apresentam tempos de simulaç o menores que a filtragem padr o para cen rios em larga escala, e maiores em cen rios em pequena escala comprovando o ganho reduzido da proposta para traços n o abrangentes. Para esses casos, a aplicaç o do m dulo de filtragem que soluciona o problema de “sa da e retorno” foi a que apresentou resultados mais consistentes, aumentando o n mero de entregas bem sucedidas de pacotes com aumentos relativamente baixos no tempo de processamento e consumo de mem ria. Grande parte das filtraens para cen rios em pequena escala obtiveram uma diminuiç o substancial do consumo de mem ria sem afetar o sucesso na entrega de dados pela rede. Ainda assim, a vari ncia do n mero de entregas de dados bem sucedidas pela rede ao longo das simulaç es incentiva estudos futuros sobre a melhor configuraç o dos m dulos de filtragem para cada caso de estudo. Por fim, a filtragem, aqui proposta, contribui para a reduç o do n mero de n s na rede e, conseq entemente, na reduç o do consumo de mem ria e esforço computacional requisitados por simulaç es. Isso   alcançado de forma original atrav s da filtragem din mica do traço de mobilidade.

1.2 Organizaç o

Nos cap tulos seguintes, o trabalho   organizado da seguinte forma. O Cap tulo 2 apresenta os trabalhos relacionados a esta disserta o, classificando-os em trabalhos para aprimoramento de simuladores, descritos na Seç o 2.1; e trabalhos que otimizam o uso dos traços de mobilidade, descritos na Seç o 2.2. O Cap tulo 3 apresenta a implementa o do modelo de filtragem proposto e detalha o funcionamento dos m dulos desenvolvidos. Para isso, a Seç o 3.1 introduz as notaç es e definiç es

usadas ao longo do trabalho juntamente com os conceitos ligados à sua criação. Já a Seção 3.2 caracteriza os traços resultantes da filtragem padrão de simuladores de mobilidade e os problemas que estes podem ocasionar às simulações. A estrutura do método de filtragem proposto e o funcionamento de seus módulos implementados são apresentados na Seção 3.3, e sua implementação em código é discutida na Seção 3.5. A configuração das simulações usadas para a validação da estratégia de filtragem está no Capítulo 4, na qual a Seção 4.1 caracteriza os cenários das simulações e configurações do filtro usadas na análise. A Seção 4.2 comenta sobre os resultados obtidos. Por fim, o Capítulo 5 conclui este trabalho e apresenta direções futuras na Seção 5.1.

Capítulo 2

Trabalhos Relacionados

Atualmente, métodos que buscam a redução do esforço computacional de simulações assumem duas formas complementares na literatura: a primeira visa aprimorar o funcionamento dos simuladores e a segunda visa otimizar o uso dos traços de mobilidade.

2.1 Propostas para aprimoramento de simuladores

Em relação aos simuladores, a ideia é: ou paralelizar o processamento, ou reduzir o conjunto de veículos que devem agendar eventos de recepção por transmissão de pacote [7], ou reduzir a complexidade de cálculos de interferência entre sinais aplicando algoritmos como o de N-corpos [17].

A paralelização é uma técnica muito usada para agilizar o tempo computacional de processos. Estudos em simulações de rede de grande escala concluíram que é possível paralelizar a simulação de redes tanto no eixo temporal quanto na simulação de eventos discretos [18]. A paralelização no eixo temporal envolve a separação de operações com tempo de processamento fixo em processadores separados. Essa técnica permite a simulação de um tráfego maior de pacotes na rede devido a grande parte da simulação de entrega de pacotes envolverem a mesma sequência de operações. A paralelização de eventos discretos envolve uma abordagem de particionamento espacial de uma grande rede onde os nós de cada partição são simulados em diferentes processadores. Essa segunda forma permite aumentar o tamanho da rede e o tráfego simulado nela. Contudo, esta não soluciona o problema de consumo de memória para redes com número grande de veículos, e requer outras técnicas para superar essa restrição. Já a redução do conjunto de veículos que devem agendar eventos de recepção por transmissão de pacotes é outra técnica usada para o aprimoramento do desempenho de simuladores.

Essa abordagem é atacada por Ricardo Fernandes e Michel Ferreira [6] ao buscarem simulações escaláveis de VANETs no ns-3. Os autores apresentam uma estrutura de dados de indexação espacial para o ns-3 em forma de árvore quaternária. Nela,

a área de simulação representa a raiz da árvore de indexação. Essa é dividida em quatro células de área homogênea que são subdivididas da mesma maneira sucessivamente até que cada subcélula apresente nenhum, ou apenas um nó da rede sendo simulada. Dessa forma, cada nó da árvore corresponde a uma célula de $1/4$ do tamanho do seu nó pai, e as folhas da árvore apresentam nenhum, ou apenas um veículo da VANET. Realiza-se, então, a rotina de recepção de transmissões percorrendo a árvore a partir da folha do veículo transmissor e executando a rotina nas folhas vizinhas do mesmo ramo da árvore. Como a coordenada de cada nó está implícita no índice de sua folha na árvore quaternária, a rotina de recepção de transmissões é executada primeiramente nos veículos mais próximos do veículo transmissor. Apenas as células que apresentam interseção com o raio de alcance de transmissão do veículo transmissor realizam a rotina de recepção; assim que é alcançado um nó interno da árvore, cujo tamanho de sua célula englobe todo o alcance de transmissão do veículo transmissor a exploração da árvore pode ser interrompida. A árvore de indexação deve ser atualizada conforme a dinamicidade da rede para manter a correta execução da rotina de recepção de pacotes. O estudo deste método de otimização demonstrou melhoras na escalabilidade de simulações de redes esparsas. No pior caso, quando cada nó consegue alcançar qualquer nó da rede, o desempenho torna-se comparável com a simulação ns-3 não otimizada com sobrecarga imposta pelo cálculo da árvore de indexação.

Por fim, para reduzir a complexidade do cálculo de interferência entre sinais transmitidos, Felipe Perrone e David Nicol [17] propuseram atacar o problema como se esse fosse um problema de N-corpos. O problema de N-corpos é comumente encontrado na astrofísica e lida com o cálculo da interação gravitacional entre vários pares de corpos distantes. Para solucioná-lo, o trabalho usa o algoritmo Barnes–Hut [19] onde nós são agrupados e tratados como uma única entidade para o cálculo de interferência sofrido por nós distantes. O estudo possibilita a simplificação da complexidade temporal da interação entre N nós transmissores ou receptores de $\mathcal{O}(N^2)$ interações para $\mathcal{O}(N \log N)$, ou até $\mathcal{O}(N)$. O artigo ainda compara o uso do algoritmo Barnes–Hut com a truncagem dos sinais por distância, que equivale à técnica anterior de eliminação de veículos distantes da rotina de recepção de transmissões pelo uso de árvores quaternárias. Seus resultados demonstram que a técnica de truncagem por distância pode atingir um rendimento melhor do custo de processamento em relação ao erro de interferência relativo alcançado para uma certa margem de ângulo de tolerância.

2.2 Propostas de otimização do uso de traços

A outra abordagem para tornar as simulações veiculares mais escaláveis, a otimização no uso de traços, é uma etapa anterior à simulação. A ideia é melhorar o uso da informação fornecida pelos traços para gerar simulações mais escaláveis. Isso pode ser feito através da remoção de veículos ou amostragens que não interfiram nos resultados pretendidos, ou no desenvolvimento de estruturas de dados mais eficientes que reduzam o tempo de acesso à memória. A otimização de traços apresenta a vantagem de que, uma vez feita, o traço pode ser utilizado por várias simulações sem processamentos adicionais. Nessa direção, a redução do tempo de acesso aos traços na memória é ideal para tornar as simulações mais rápidas e escaláveis. Isso pode ser obtido através da compressão dos traços em estruturas de dados mais eficientes, ou da filtragem de dados redundantes ou irrelevantes. A compressão pode ser feita com ou sem perda de informação [20]. Os traços podem ser comprimidos sem perda de informação pelo uso de estrutura de dados binários mais densas e com baixa sobrecarga mas que seriam ilegíveis por humanos. Os traços de mobilidade do ns-3 já apresentam uma estrutura bem compacta por registrar apenas alterações de trajetórias e com sobrecarga mínima para que os traços sejam legíveis. Logo, os ganhos maiores são obtidos na aplicação de técnicas de compressão e filtragem em que se assume a perda de parte das informações.

A técnica de Douglas–Peucker [21], por exemplo, permite a compressão de trajetórias assumindo certa perda de precisão ao reduzir o número de pontos necessários para descrever uma trajetória. Os pontos a serem retirados são determinados como sendo aqueles que resultam no menor erro máximo entre a curva original e a curva simplificada, mantendo o erro de trajetória dentro de margens predefinidas. Como veículos restringem-se aos limites de redes rodoviárias, modelos de dados e particionamento de rede podem ser criados para levar em conta esse fato. Um estudo foi realizado seguindo essa ideia, onde modelos de dados e particionamentos de rede baseados na compressão de Douglas–Peucker foram aplicados para enquadrar as trajetórias de rede dentro de margens de erro predeterminadas [22]. Seus resultados obtiveram taxas de compressão próximas às ideais, referente ao cenário em que cada trajetória é contida por uma única rua. Contudo, para que isso fosse possível, foi necessário assumir certo erro na trajetória e velocidade dos veículos em relação ao cenário original.

Como visto, os estudos anteriores buscam aprimorar o processo de simulação ou otimizar o uso dos traços. Porém, pouca pesquisa existe na redução do tamanho de traços por métodos de filtragem com perda de dados. Estes ficam restritos ao método de Douglas–Peucker e a filtros básicos já fornecidos por exportadores de traços sintéticos como o `traceExporter.py` do SUMO [23]. Os filtros de exportadores

de traços existentes envolvem filtragens estáticas dos dados por subamostragem da posição dos veículos em intervalos fixos de tempo, ou restrição das amostras àquelas referentes a regiões geográficas fixas ou a intervalos de tempo. Mesmo estas sendo usadas para a criação de simulações em escala reduzida, ainda assim sofrem com problemas de realismo devido à descontinuidade da trajetória de veículos nas regiões limites de sua filtragem. Portanto, seus traços gerados não são ideais para simular o comportamento realista de uma parte de uma rede mais extensa.

As limitações nas técnicas de filtragem de traços abrem a oportunidade de desenvolvimento de um método que responda aos problemas constatados e leve a boas reduções no consumo de memória e processamento para a simulação de redes com nós móveis. Para que a filtragem apresente reduções no uso de recursos computacionais sem afetar a qualidade da simulação, é necessário que o método de escolha dos nós que devem ser mantidos no traço seja dinâmico e leve em conta as particularidades da rede. A seção seguinte descreve o método de filtragem de traços proposto neste trabalho. Esse é criado com o objetivo de reduzir o número de nós de um traço de mobilidade, tendo em vista a superação dos métodos já estabelecidos para simulações escaláveis de redes de larga escala.

Capítulo 3

Filtragem Proposta

Na solução proposta, um filtro é adicionado como estágio de processamento intermediário entre a fonte de traços de mobilidade, seja essa um banco de dados ou gerador de traços sintéticos, e o simulador de rede.

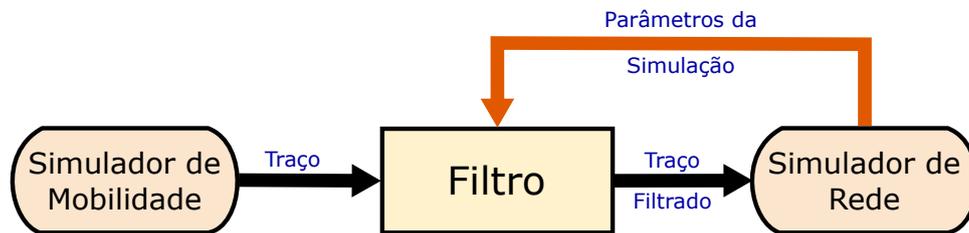


Figura 3.1: Fluxo da simulação de Redes Veiculares usado no trabalho.

Algumas suposições são feitas quanto à natureza da simulação. Para que a filtragem seja realizada, é preciso que se tenha conhecimento dos veículos de interesse à simulação (receptores ou originadores de transmissões) de onde se possa aplicar a filtragem por contato. Caso contrário, o filtro fica restrito a trabalhar com base em filtragens estáticas como por área geográfica ou por cortes de tempo.

A partir dos veículos originadores ou receptores de transmissões, busca-se determinar todos os demais veículos que possam interferir no resultado da aplicação a ser simulada para que estes sejam mantidos no traço final. Portanto, os veículos roteadores de dados são descobertos ao analisar os contatos entre os veículos dos traços de mobilidade. Veículos que tenham entrado em contato com veículos fontes ou destinatários de transmissões são considerados possíveis roteadores e, dependendo da aplicação a ser simulada, devem ser mantidos no traço assim como outros veículos que entrem em contato com estes.

Uma proposta com resultados preliminares desse trabalho foi defendida com a apresentação de um artigo no VIII-SBESC (*VIII Brazilian Symposium on Computing Systems Engineering*) [24].

Para uma melhor compreensão do trabalho, a seção seguinte define algumas notações usadas para descrever os dados presentes nos traços de mobilidade. Também são definidos diversos tipos de veículos do traço que são caracterizados segundo sua função e identificados de acordo com a forma como estes são tratados durante a filtragem.

3.1 Notações e definições

Durante a simulação de uma rede veicular, veículos são tratados como nós móveis que podem trocar dados entre si ao entrarem em contato dentro da região simulada. Denota-se por \mathcal{V} o conjunto de veículos pertencentes ao traço de mobilidade, t_0 e t_f os instantes de início e término definidos para a simulação, e α a região geográfica simulada. Portanto, a partir do traço de mobilidade, pretende-se simular as comunicações de rede referentes aos veículos de \mathcal{V} dentro da região α entre os instantes de tempo t_0 e t_f .

No traço de mobilidade 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 veículo, e (X^v, Y^v) , sua posição. Dessa forma, o traço pode ser visto como uma lista ordenada de tuplas em função do instante de tempo t . Para fins de análise, define-se $T(t) \subseteq T$ como a lista de tuplas referentes a um determinado instante de tempo t .

Cada veículo $v \in \mathcal{V}$ entra e sai da região simulada em certos momentos e pontos da simulação. Sendo assim, para cada veículo da simulação, define-se seu instante de entrada t_e^v na posição (X_e^v, Y_e^v) e seu instante de saída 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 $(X_e^v, Y_e^v), (X_s^v, Y_s^v) \in \alpha$.

Para fins de filtragem, a vizinhança de cada veículo v , $\mathcal{N}(v) \subseteq \mathcal{V}$, é definida pelos veículos dentro de um raio de alcance d a partir deste. Onde d representa o raio máximo de contato onde se espera que, durante a simulação, uma transmissão de dados a partir de v possa ser realizada com sucesso. O grupo de veículos pertencentes a $\mathcal{N}(v)$ varia de acordo com o instante de tempo t .

A seguir são definidos alguns termos para melhor classificar os membros dos traços durante a filtragem. Os veículos definidos nesta seção são aplicados nos diferentes módulos de filtragem descritos na Seção 3.3. As notações propostas estão sumarizadas na Tabela 3.1.

Tabela 3.1: Sumário das notações usadas.

Notação	Significado
T	Traço ao qual se aplica a filtragem
$T(t)$	Lista de tuplas do traço T referentes ao instante de tempo t
t	Instante de tempo específico da simulação
v	Veículo pertencente ao traço T
v_i	Veículo de Interesse
v_s	Veículo Infectado
v_r	Veículo Rastreado
t_0	Tempo inicial da simulação de rede
t_f	Tempo final da simulação de rede
\mathcal{V}	Conjunto de veículos v pertencentes ao traço T
\mathcal{V}_s	Conjunto de veículos v_s pertencentes ao traço T
\mathcal{V}_i	Conjunto de veículos v_i pertencentes ao traço T
\mathcal{V}_r	Conjunto de veículos v_r pertencentes ao traço T
\mathcal{V}_s^{rt}	Conjunto de veículos v_s pertencentes ao traço T que foram infectados retro-temporalmente
\mathcal{A}	Conjunto de veículos que deixaram a área de interesse e passaram a ser rastreados para caso eles voltem
\mathcal{J}	Conjunto contendo o número de saltos de infecção restantes referentes a cada veículo infectado de forma direta
\mathcal{J}^{rt}	Conjunto contendo o número de saltos de infecção restantes referentes a cada veículo infectado retro-temporalmente
d	Raio de contato considerado para a filtragem
$\mathcal{N}(v)$	Conjunto de veículos pertencentes à vizinhança de raio d de um veículo v
$\mathcal{N}(\mathcal{V})$	Conjunto de veículos pertencentes à união de todas as vizinhanças de raio d dos veículos v pertencentes a \mathcal{V}
ρ	Número de saltos de infecção restantes referentes a um veículo infectado v_s
ρ_{max}	Número máximo de saltos de infecção a partir de um veículo de interesse v_i definido pelo usuário

3.1.1 Membros do traço de mobilidade

Abaixo definem-se alguns membros presentes em traços de mobilidade segundo suas funções na simulação de uma rede veicular.

Definição 1 (*Veículo de interesse*). São veículos fontes ou destinos de pacotes transmitidos na rede ou qualquer veículo cujos contatos com a vizinhança sejam de interesse para a simulação. Após a filtragem, as entradas referentes a estes veículos e seus vizinhos são mantidas no traço.

Tomando $\mathcal{V}_i \subset \mathcal{V}$ como o conjunto de todos os veículos de interesse, e $\mathcal{N}(\mathcal{V}_i)$ o conjunto de veículos vizinhos a pelo menos um veículo v_i pertencente ao conjunto

de veículos de interesse \mathcal{V}_i . Após a filtragem, deseja-se manter no traço as tuplas referentes aos veículos de interesse $v_i \in \mathcal{V}_i$ e aos veículos em suas vizinhanças ($v \in \mathcal{N}(\mathcal{V}_i)$) para cada instante de tempo t avaliado.

Definição 2 (*Veículo infectado*). São todos os veículos que, após se tornarem vizinhos de um veículo de interesse v_i , passam a ser tratados como veículos de interesse pelo processo de filtragem (como no caso dos veículos realizando encaminhamento de dados). Assim como os veículos de interesse, os veículos infectados possuem tanto suas posições quanto as posições de seus vizinhos mantidas no traço após a filtragem.

Esses são chamados de veículos infectados devido à aplicação de algoritmo epidêmico e para que seja possível diferenciá-los dos veículos de interesse. Considera-se que um veículo v , após se tornar vizinho de um veículo de interesse v_i , torna-se um veículo infectado v_s . A infecção de interesse pode ocorrer de forma recursiva com veículos infectados originando novos veículos infectados. Tal infecção epidêmica corre o risco de inundar o conjunto de nós de mobilidade do traço em apenas alguns saltos. Portanto, é determinado um limite superior ρ_{max} para o número de saltos de infecção a partir de um veículo de interesse. Para monitorar o número de saltos de infecções restantes para cada veículo infectado, é definido um conjunto \mathcal{J} contendo o número de saltos de infecção restantes referentes a cada veículo infectado.

O conjunto de todos os veículos infectados \mathcal{V}_s pode ser obtido a partir do conjunto de veículos de interesse \mathcal{V}_i ao analisar todos os veículos que em algum instante de tempo pertenceram à união de suas vizinhanças $\mathcal{N}(\mathcal{V}_i)$ podendo ser expandido em seguida a qualquer veículo que tenha pertencido a $\mathcal{N}(\mathcal{V}_i) \cup \mathcal{N}(\mathcal{V}_s)$ e assim sucessivamente contanto que não se atinja o limite de número de infecções.

Definição 3 (*Veículo rastreado*). São veículos cujas trajetórias devem ser mantidas no traço para o objetivo da simulação, mas não necessariamente realizam transmissões de dados. A vizinhança destes veículos não é considerada para fins de filtragem e apenas suas entradas são mantidas no traço.

Esses veículos são normalmente determinados através de contato com veículos de interesse ou infectados, ou predefinidos na configuração do filtro. Eles podem ser predefinidos antes da filtragem como destinos de pacotes ou mesmo como veículos que interajam com veículos de interesse ou infectados, mas normalmente não efetuam o papel de transmissores de pacotes por apenas poderem receber ou transmitir pacotes a veículos que já pertençam ao traço. Isso é consequência da vizinhança dos veículos rastreados não ser necessariamente mantida no traço final.

Um veículo v , ao entrar no conjunto de vizinhos de um veículo de interesse v_i ($\mathcal{N}(v_i)$) ou infectado v_s ($\mathcal{N}(v_s)$), pode tornar-se um veículo rastreado v_r e ter todas as

suas entradas mantidas no traço. As entradas no traço referentes ao veículo rastreado podem ser mantidas para todos os instantes de tempo anteriores ou posteriores ao seu contato com v_i ou v_s . Essa possibilidade é de interesse à filtragem pois permite que veículos que interajam com veículos de destino em um ponto futuro também tenham as suas posições anteriores rastreadas. Logo, de forma generalizada, o conjunto de veículos rastreados \mathcal{V}_r é formado por todos os veículos que, em algum momento do traço, pertenceram a $\mathcal{N}(v_i) \cup \mathcal{N}(v_s)$.

Para fins de simulação, adicionalmente, a definição do veículo rastreado é importante para contornar o problema de “saída e retorno” de nós do traço, como será explicado posteriormente, que afeta alguns simuladores de rede a eventos discretos.

Definição 4 (*Área de interesse*). É a área geográfica α selecionada do traço de mobilidade relevante aos fins da simulação. A área de interesse deve incluir a trajetória que todos os veículos de interesse v_i ($i \in \mathbb{N}$) percorrem.

A área de interesse é a região por onde os veículos de interesse v_i ($i \in \mathbb{N}$) se deslocam. Sendo assim, para um conjunto de veículos de interesse v_i e seus pontos de entrada $(X_e^{v_i}, Y_e^{v_i})$ e saída $(X_s^{v_i}, Y_s^{v_i})$ do traço, a área de interesse é uma figura geométrica que abrange todos os pontos da trajetória de cada v_i entre ambos os pontos. Independente da figura geométrica, a área de interesse α delimita os nós de interesse da rede que devem ser mantidos no traço. Tal região é usada pela filtragem clássica de traços na realização de cortes de áreas do traço para se obter simulações em menor escala. Apesar do método proposto se basear no contato entre nós de mobilidade para a realização da filtragem, uma filtragem inicial que restrinja os traços a uma região que contenha os nós de interesse acelera a filtragem por contato quando os arquivos de traços são grandes e abrangentes.

As Figuras 3.2 e 3.3 ilustram as definições desta seção. A área de interesse α é definida a partir das posições de entrada $(X_e^{v_i}, Y_e^{v_i})$ e de saída $(X_s^{v_i}, Y_s^{v_i})$ do veículo de interesse v_i . O veículo v_r da Figura 3.2, após entrar na vizinhança de v_i , torna-se um veículo rastreado e, logo, passa a ter todas as demais posições de sua trajetória armazenadas no traço, mesmo que este saia da área de interesse α .

Já a Figura 3.3 apresenta o uso dos veículos infectados na mesma área de interesse. O veículo de interesse v_i gera um veículo infectado v_s que, por sua vez, infecta posteriormente v'_s . As vizinhanças dos nós estão representadas nas figuras por círculos concêntricos aos nós. A infecção ocorre quando um nó do traço se torna vizinho de outro já infectado. Diferentemente dos veículos rastreados, os veículos infectados e de interesse também mantêm, no traço, as entradas referentes aos veículos em suas vizinhanças.

3.2 Desafios na filtragem e uso de traços em simulações de VANETs

Antes de descrever o procedimento de como o traço será filtrado, vale ressaltar alguns problemas enfrentados pelo uso de traços em simulações de VANETs que devem ser abordados por este trabalho. Como introduzido anteriormente, o uso de traços em escala real de veículos circulando em uma cidade para simulações de rede a eventos discretos enfrenta dois grandes desafios: o consumo de memória e o tempo de processamento.

Devido à forma como simuladores de rede como o OMNeT++ e o ns-3 tratam os nós da rede, todos os nós a serem simulados devem ser carregados e mantidos em memória durante a simulação. Nestes simuladores, cada nó é considerado como uma máquina e não pode ser removida da simulação, apenas desativada. Assim, a partir do momento que o arquivo de mobilidade da simulação é lido, todos os nós

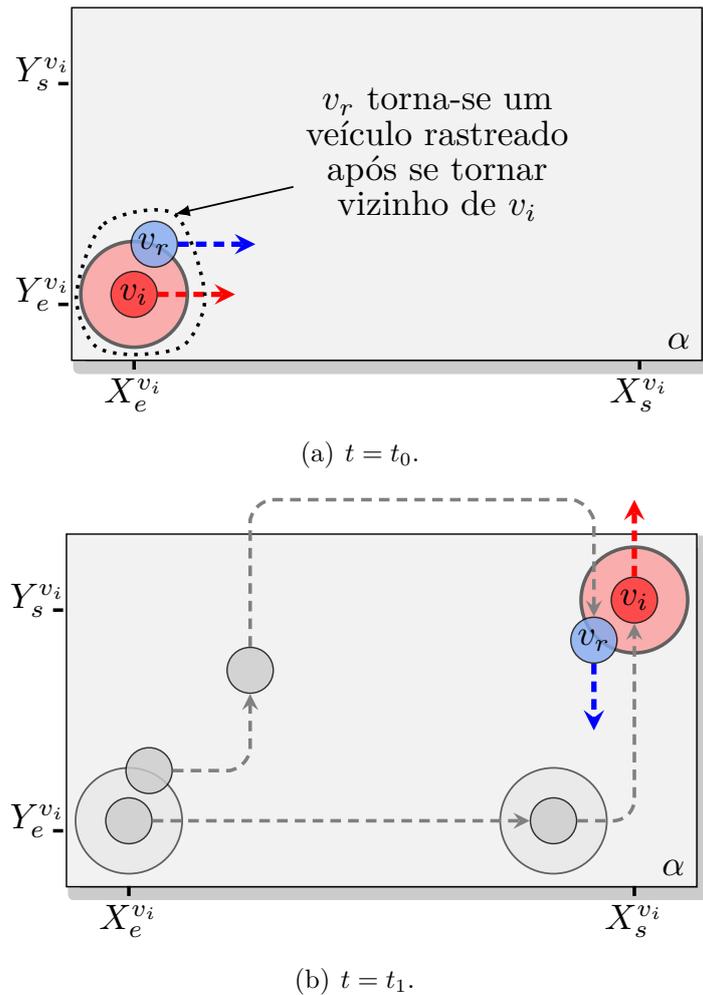


Figura 3.2: (a) O veículo v_r torna-se um veículo rastreado. (b) Após se distanciar de v_i , v_r se reaproxima em momento posterior.

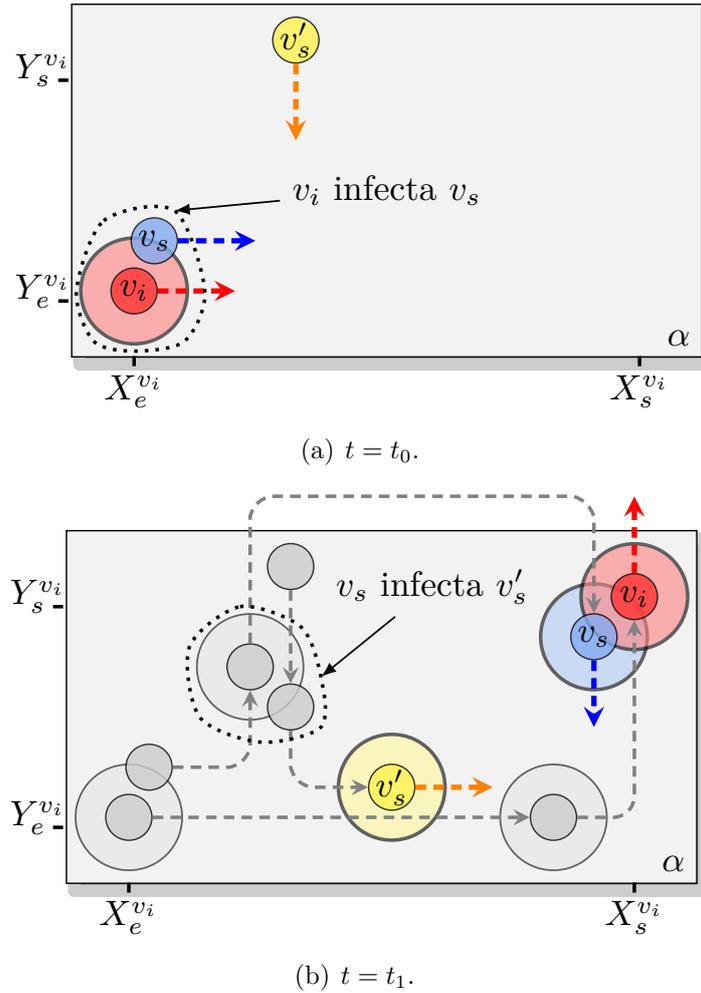


Figura 3.3: (a) O veículo v_s é infectado por v_i . (b) v_s e v'_s também se comportam como v_i , sendo v'_s infectado por v_s ao se encontrarem entre t_0 e t_1 .

são carregados em memória e fornecidos identificadores únicos. Essa inicialização desconsidera os instantes de entrada e saída dos nós do traço. Por exemplo, um nó v definido em um instante de tempo $t_e^v \gg t_0$ e que posteriormente deixe o traço de mobilidade em um instante de tempo $t_s^v \ll t_f$ é alocado em memória desde o início da simulação e ocupa um espaço em memória até que a simulação termine.

O segundo problema envolve o fato de que cada evento de transmissão de pacote leva a um laço de repetição envolvendo todos os nós da simulação, mesmo aqueles que não podem receber a transmissão. Como o número de nós em traços de simulações de cidades é grande e nós não são retirados da memória ao completarem seu percurso, tanto a memória quanto o tempo de processamento crescem rapidamente, tornando a escala de simulações um problema para simuladores como OMNeT++ e ns-3.

Para solucionar esses problemas sem alterar o funcionamento dos simuladores de rede, deve-se reduzir o número de nós no traço de mobilidade. Processos de filtragem podem ser usados para esse fim, aumentando a escalabilidade das simulações. Atualmente, as únicas formas de filtragem de traços suportadas pelo SUMO são através

da ferramenta `traceExporter.py` [23]. Essa ferramenta reduz o traço de forma espacial, ao recortar a área de simulação em um retângulo; e de forma temporal, de acordo com um período de tempo delimitado por um instante inicial e final. Apesar de melhorar a escalabilidade da simulação, essa filtragem ainda é muito simples, especialmente quando os veículos de interesse possuem trajetórias longas e esparsas entre si. Quando estes apresentam esse tipo de trajetória, a área de interesse se torna grande e leva à inclusão de vários veículos que não participam da aplicação. As Figuras 3.2(a) e 3.3(a) podem ser usadas para ilustrar o problema. Em ambas as figuras, os pontos (X_s^{vi}, Y_s^{vi}) não possuem nenhuma interação envolvendo veículos de interesse ou infectados para $t = t_0$. Portanto, qualquer veículo naquela região da área de interesse deve ser excluído do traço.

A filtragem por área apresenta, contudo, um novo problema à simulação de VANETs. Como dito anteriormente, simuladores de rede apresentam estruturas eficientes para o tratamento de dados de mobilidade. Para reduzir processamento, alguns simuladores de rede, como o ns-3 e o OMNeT++, assumem que cada nó móvel do traço de mobilidade se desloca de sua atual posição à sua próxima posição amostrada em uma trajetória retilínea e contínua. Devido a esses simuladores de rede não permitirem o teleporte de seus nós dentro do traço de mobilidade, um nó que apresente parte de sua trajetória fora da área de interesse é simulado como se este tivesse se deslocado de seu ponto de saída de α diretamente ao seu ponto de retorno como ilustrado na Figura 3.4. Essa mudança na trajetória do veículo pode gerar interações do veículo com outros nós do traço que não seriam possíveis em sua trajetória original.

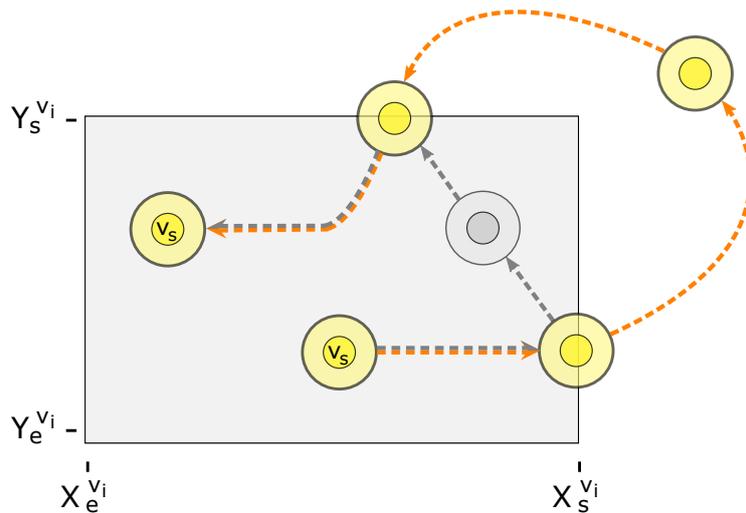


Figura 3.4: Comparação de trajetórias do traço real (laranja) e filtrado (cinza) quando o problema de saída e retorno se apresenta.

A ferramenta `traceExporter.py` soluciona o problema para traços de mobilidade do ns-2 ao desconsiderar veículos que tenham saído da área de interesse caso

estes retornem a ela. Já para traços de mobilidade OMNeT, essa considera veículos que retornem à área de interesse como novos veículos e os fornece um novo identificador único. Essas soluções, entretanto, geram inconsistências como pode ser exemplificado pela Figura 3.2(b) e 3.3(b), onde os veículos v_r e v_s saem e retornam à área de interesse, e passam a interagir com o veículo de interesse. No caso de uma filtragem realizada pela ferramenta `traceExporter.py`, essa interação não seria considerada, ou seria necessário a transferência de estado do veículo que saiu da área de interesse e foi desativado para o novo veículo introduzido à simulação.

Uma outra solução seria desativar o módulo de rede do veículo enquanto este estiver fora da área de interesse para evitar interações do veículo enquanto este percorre a trajetória entre seu ponto de saída e reentrada. Contudo, essa solução, assim como a transferência de estado a um novo veículo criado, requer uma coordenação entre a geração do traço de mobilidade filtrado e a simulação de rede gerando novas complicações. Dado que o traço de mobilidade filtrado não armazena em si a informação sobre qual veículo deixa e retorna à área de interesse, e não há como diferenciar, um veículo que tenha saído da área de interesse de um veículo que tenha simplesmente realizado uma trajetória retilínea de seu ponto de saída a seu ponto de retorno. Assim, o traço de mobilidade filtrado não pode ser usado em simuladores de rede sem um arquivo gerado contendo os instantes que cada veículo saiu e retornou ao traço. Esse problema, no qual um veículo deixa o traço de mobilidade e retorna em um momento posterior, é referenciado neste trabalho como o problema de “saída e retorno” e deverá ser abordado pelo método de filtragem proposto.

3.3 Módulos de filtragem e funcionamento da proposta

Com base nas definições da Seção 3.1, a filtragem dinâmica por contato foi separada em diversos módulos para simplificar a implementação do filtro e suportar diferentes casos de uso. Esses módulos são separados em:

- filtragem por Área de Interesse,
- filtragem por Veículos de Interesse,
- filtragem por Rastreamento de Veículos,
- filtragem por Infecção de Interesse e
- filtragem por Infecção de Interesse Retro-temporal.

Dependendo da aplicação, apenas alguns módulos podem ser suficientes para se obter um traço com um número reduzido de nós de mobilidade e, por conseguinte, diminuir o esforço computacional da simulação de rede. Estes também podem ser usados em conjunto para alcançar um traço mais realista em troca de incluir um número maior de nós.

A subseção abaixo descreve como estes módulos são agrupados para formar o filtro proposto e, em seguida, descreve-se o funcionamento de cada módulo.

3.3.1 Algoritmo de filtragem

O traço de mobilidade T é lido como uma lista de tuplas ordenadas no tempo t . Para que uma entrada do traço no instante t seja mantida após a filtragem, é preciso que seu veículo correspondente v pertença a $\mathcal{V}_i \cup \mathcal{V}_s \cup \mathcal{V}_r \cup \mathcal{N}(\mathcal{V}_i) \cup \mathcal{N}(\mathcal{V}_s)$ no instante t analisado entre t_0 e t_f . A subdivisão da filtragem em módulos permite determinar a qual grupo cada veículo pertence. Funções específicas, responsáveis por atualizar cada conjunto de veículos e julgar se determinado veículo do traço pertence ao respectivo grupo ou a sua vizinhança, são criadas para cada módulo. Assim, um fluxograma geral do processo de filtragem pode ser descrito na figura 3.5.

O primeiro estágio do algoritmo de filtragem realiza uma filtragem com base na área de interesse através da função `IS_IN_AREA`. O objetivo desta filtragem inicial é reduzir o número de entradas do traço que seguem para os próximos estágios de filtragem com o menor esforço computacional possível. As demais funções de filtragem usam algoritmos de filtragem dinâmica que exigem um processamento maior. Logo, uma filtragem inicial por área torna o processamento mais eficiente quando os veículos de interesse da simulação se limitam em uma região comparativamente pequena do traço. Após a filtragem inicial por área, a entrada do traço é fornecida aos demais módulos, para cada qual uma função determina se a entrada deve ser mantida no traço. Se ao menos uma dessas funções retorne verdadeira, a entrada é mantida pois considera-se que o veículo pertença a algum dos conjuntos de veículos relevantes para a simulação.

Nem sempre todos os módulos são necessários para se realizar a filtragem. Dependendo do caso de uso da simulação, talvez não se precise realizar infecção de interesse; ou todos os veículos que realizarão o roteamento são predefinidos e basta que estes sejam rastreados; ou a área de interesse engloba grande parte do traço tornando desnecessário o uso da filtragem por área. Sendo assim, cada módulo pode ser desativado dependendo da filtragem que se deseja realizar para poupar tempo de processamento e incluir no traço final apenas os veículos de interesse à simulação. Em código, isso pode ser feito de forma simples através de uma *flag* que desative a execução do módulo (de modo que este retorne *false* imediatamente após

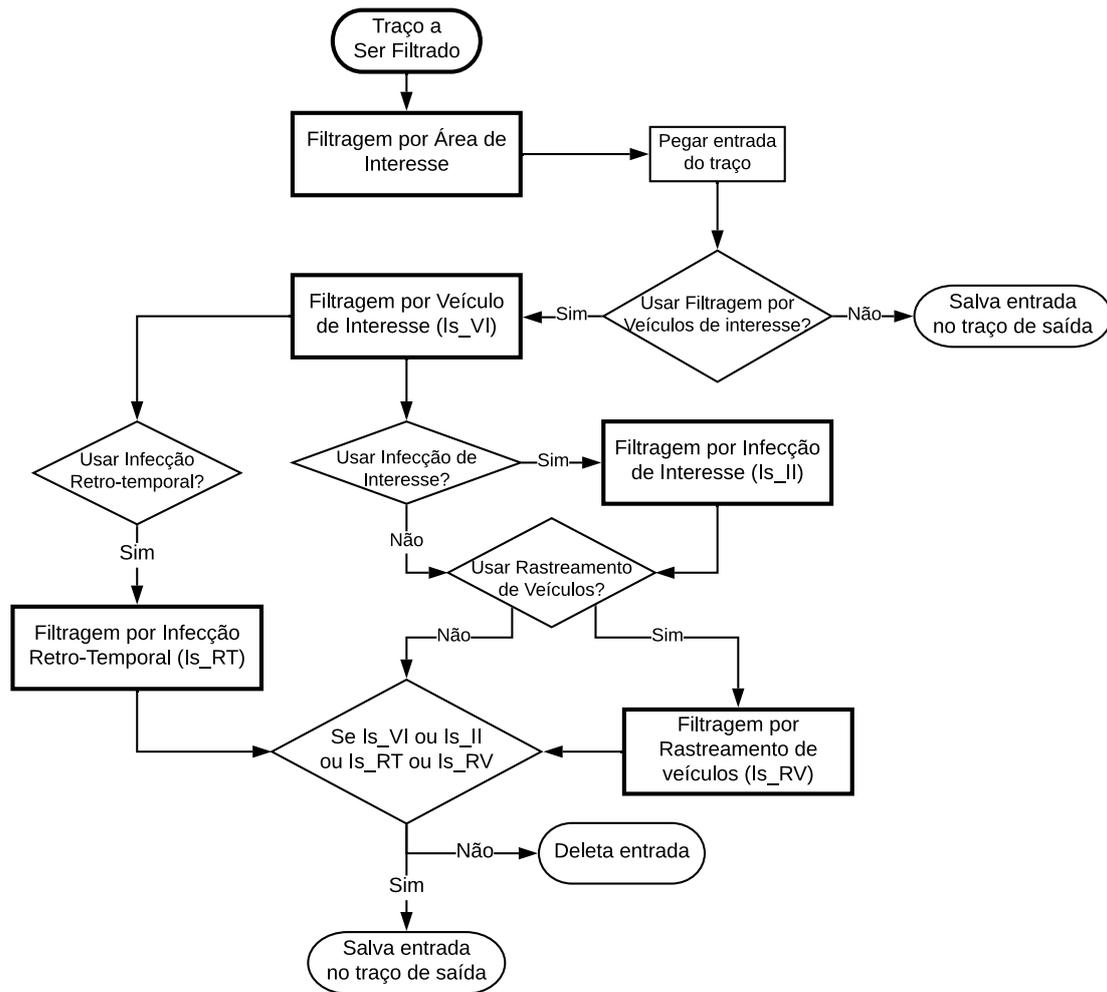


Figura 3.5: Configuração dos módulos de filtragem.

ser chamado).

Como visto na Subseção 3.1.1, os veículos de interesse são os membros centrais dos módulos de filtragem. Por serem origem ou destinos dos pacotes de dados da rede, eles determinam, através de seus contatos com outros veículos, os veículos rastreados e infectados a serem adicionados. Cada módulo é responsável por atualizar o conjunto de veículos do traço de determinado tipo. Portanto, apesar de cada um dos quatro módulos poder determinar se a entrada deve ser mantida no traço, eles devem ser executados em uma ordem determinada. Como o conjunto de veículos de interesse e infectados influenciam na atualização do conjunto de veículos rastreados, seus módulos devem ser executados primeiro.

O modo em como os módulos são unificados em uma única solução de filtragem é detalhado no no Algoritmo 1. Nela, a linha 1 inicializa as variáveis a serem usadas. A filtragem com base na área de interesse é realizada na linha 2, limitando o traço à área e intervalo de tempo desejados. As linhas 3 a 11 aplicam a filtragem por

infecção retro-temporal definida na Subsecção 3.3.6, e salvam um traço temporário T_{rt} contendo as entradas dos veículos que foram infectados retro-temporalmente e seus vizinhos. A linha 14 colhe as entradas do traço referentes ao instante de tempo t ($T(t)$) para análise. Isso é feito de forma ordenada graças ao laço de repetição da linha 13 com o acréscimo temporal da linha 22. As linhas 15 a 19 abordam cada veículo v do subconjunto $T(t)$ e julgam se sua entrada para o instante de tempo sendo analisado deve ser mantida no traço. O julgamento é feito pelas funções IS_VI, IS_RV, e IS_II dos módulos de filtragem (descritos nas subsecções 3.3.3, 3.3.4, e 3.3.5 respectivamente). Se a função de pelo menos um dos módulos ativos aceita a entrada do veículo v , essa é mantida no traço de saída. Por fim, a linha 21 unifica os traços filtrados diretamente e retro-temporalmente e atualiza o traço de saída.

Algoritmo 1 Algoritmo da estrutura geral da filtragem.

Entrada: $\mathcal{V}_i, T_{rt}, \rho_{max}, T, \alpha, t_0, t_f, d$

Resultado: T filtrado

```

1:  $t = t_f; T(t), \mathcal{V}_s, \mathcal{V}_s^{rt}, \mathcal{V}_r, \mathcal{J}, \mathcal{J}^{rt} = \{\}$ 
2: IS_IN_AREA( $T, \alpha, t_0, t_f$ )           ▷ Filtra o traço  $T$  segundo a função IS_IN_AREA
3: enquanto  $t \neq t_0$  faça           ▷ Realiza a filtragem retro-temporal
4:    $T_{rt}(t) \leftarrow$  PEGA_TRAÇO( $T, t$ )   ▷ Filtra as entradas do traço  $T$  referentes ao instante  $t$ 
5:   para todo  $v$  em  $T(t)$ 
6:     se IS_RT( $v, \mathcal{V}_i, \&\mathcal{V}_s^{rt}, T_{rt}(t), \&\mathcal{J}^{rt}, \rho_{max}$ ) então
7:       Continue
8:     senão
9:        $T_{rt}(t) \leftarrow T_{rt}(t) -$  entrada de  $v$ 
10:     $T_{rt} \leftarrow$  ATUALIZA_TRAÇO( $T_{rt}, T_{rt}(t), t$ )
11:     $t \leftarrow$  instante de tempo anterior
12:   $t = t_0$            ▷ Inicializa o instante de tempo inicial
13: enquanto  $t \neq t_f$  faça           ▷ Enquanto o traço não chega ao fim em  $t_f$ 
14:   $T(t) \leftarrow$  PEGA_TRAÇO( $T, t$ )       ▷ Filtra as entradas do traço  $T$  referentes ao instante  $t$ 
15:  para todo  $v$  em  $T(t)$ 
16:    se IS_VI( $v, \mathcal{V}_i, T(t)$ ) ou IS_RV( $v, \mathcal{V}_i, \&\mathcal{V}_r, \&\mathcal{V}_s, T(t)$ )
17:      ou IS_II( $v, \mathcal{V}_i, \&\mathcal{V}_s, T(t), \&\mathcal{J}, \rho_{max}$ ) então
18:        Continue;           ▷  $T(t)$  permanece inalterado
19:      senão
20:         $T(t) \leftarrow T(t) -$  entrada de  $v$            ▷ Retira-se a entrada de  $v$  do traço  $T(t)$ 
21:      para todo  $T_{rt}(t)$  em  $T_{rt}$ 
22:         $T \leftarrow$  ATUALIZA_TRAÇO( $T, (T(t) \cup T_{rt}(t)), t$ )   ▷ Unifica a filtragem direta com a
23:      retro-temporal e atualiza  $T$  em  $t$ 
24:   $t \leftarrow$  próximo instante de tempo
25: retorne  $T$            ▷ Retorna  $T$  filtrado

```

A seguir, é descrito o funcionamento de cada um dos diferentes módulos. O algoritmo da função usada por cada módulo para efetuar sua filtragem é incluído em sua respectiva subsecção.

3.3.2 Filtragem por área de interesse

O módulo de filtragem por área de interesse é o primeiro a ser aplicado ao traço. Este tem por objetivo eliminar o maior número possível de nós usando uma técnica de baixo custo computacional antes de submeter o traço aos algoritmos de filtragem mais complexos dos outros módulos. A técnica usada é uma filtragem por área geográfica. Assumindo traços de mobilidade grandes e abrangentes e uma região de interesse para a simulação de rede relativamente pequena, essa técnica elimina boa parte dos nós de mobilidade com baixo esforço computacional.

No entanto, como visto anteriormente, a filtragem de traços de mobilidade por área de interesse pode ocasionar o problema de “saída e retorno” se não for corretamente executado. Levando em conta as limitações dos simuladores de rede, a filtragem deste módulo faz uso de uma área de filtragem com limites flexíveis para a área de interesse. Em primeiro lugar, define-se uma área retangular de interesse na qual os veículos de interesse à simulação circulam. Em seguida, mantém-se no traço as entradas de qualquer veículo que saia da área de interesse, de forma análoga a um veículo rastreado. Dessa forma, veículos fora da área de interesse que não tenham saído dela são eliminados do traço. Sendo assim, a área de interesse pode ser considerada como uma região geográfica sem delimitações para exclusão de traços para veículos que já circulem dentro desta, e com limites fixos de exclusão de traços para veículos que circulem de fora para dentro desta. Desta forma, esse estilo de filtragem suporta o uso de veículos rastreados em futuras filtrações ao não gerar trajetórias descontínuas no traço.

A função que este módulo usa para definir os veículos que devem continuar no processo de filtragem é descrita pelo Algoritmo 2. Nele, um conjunto de veículos \mathcal{A} é definido para manter registro dos veículos que saíram e que podem, possivelmente, retornar à área de interesse. Os veículos pertencentes a \mathcal{A} são rastreados para formar uma área de interesse com limites flexíveis, e suas entradas são passadas para os demais estágios de filtragem. Para determinar se um veículo saiu da área de interesse, o algoritmo mantém uma versão temporária do $T(t)$ referente ao instante de tempo anterior ($T(t_{t-1})$). Nas linhas 6 a 14, verifica-se, para todo veículo v no intervalo de traço presente $T(t)$, se esse pertence ao conjunto \mathcal{A} ou à área de interesse. Se o veículo pertencer a qualquer um dos dois, ele é mantido no traço e o algoritmo continua. Caso contrário, verifica-se se este pertenceu à área de interesse no instante anterior. Se positivo, o veículo é adicionado ao conjunto \mathcal{A} na linha 11; se negativo, a entrada do veículo é retirada do traço na linha 14.

Algoritmo 2 Função de delimitação da Área de Interesse.

```
1: função IS_IN_AREA( $T, \alpha, t_0, t_f$ )
2:    $t = t_0; T(t), \mathcal{A} = \{\}$ 
3:   enquanto  $t \neq t_f$  faça                                ▷ Enquanto o traço não chega ao fim em  $t_f$ 
4:      $T(t_{t-1}) \leftarrow T(t)$                                 ▷ Mantém uma versão temporária do  $T(t)$  anterior
5:      $T(t) \leftarrow$ PEGA_TRAÇO( $T, t$ )                        ▷ Filtra as entradas do traço  $T$  referentes ao instante  $t$ 
6:     para todo  $v$  em  $T(t)$ 
7:       se  $v \in \mathcal{A}$  ou  $v \in \alpha$  então
8:         Continue
9:       senão
10:        se  $v \in T(t_{t-1})$  então
11:           $\mathcal{A} \leftarrow \mathcal{A} + v$                             ▷  $v$  faz parte do conjunto de veículos que saem de  $\alpha$ 
12:          Continue
13:        senão
14:           $T(t) \leftarrow T(t) -$  entrada de  $v$                 ▷ Retira-se a entrada de  $v$  do traço  $T(t)$ 
15:         $T \leftarrow$ ATUALIZA_TRAÇO( $T, T(t), t$ )                ▷ Atualiza  $T(t)$  em  $T$ 
16:         $t \leftarrow$  próximo instante de tempo
```

3.3.3 Filtragem por veículos de interesse

Tendo em vista a simulação de redes sem roteamento de pacotes, onde as comunicações são realizadas em um único salto, basta apenas considerar todos os veículos transmissores e suas vizinhanças para analisar o comportamento da rede. Sendo assim, uma filtragem espaço-temporal, onde se considera os transmissores de mensagens como veículos de interesse, é suficiente para se obter um traço realista e de tamanho consideravelmente menor sem a necessidade de se executar qualquer outra operação de filtragem.

Esse módulo funciona através da recepção de uma lista de veículos de interesse do usuário do filtro. A partir desses veículos, desconsidera-se qualquer entrada ao longo do traço que não os pertença ou às suas vizinhanças. A operação feita é análoga a uma filtragem espacial para cada instante de tempo do traço de mobilidade, onde a área de interesse é equivalente à união do raio de cobertura de cada veículo de interesse. A filtragem temporal ótima 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 de interesse a entrar na simulação, e o instante de saída $t_s^{v^i}$ do último veículo de interesse a sair da simulação respectivamente.

A filtragem desse módulo se diferencia em comparação à espaço-temporal ao evitar o desperdício de recursos com nós que não participam das interações com veículos de interesse. Porém, esse modo de filtragem torna o problema de “saída e retorno” ainda mais acentuado. Isso se deve pela área de filtragem ser uma região reduzida e descontínua que varia ao longo do tempo. Diferentemente da área geográfica, que é contínua e estática. Logo, um número ainda maior de veículos passa a estar constantemente entrando e saindo dela. Portanto, o uso exclusivo desse módulo é ideal apenas em casos onde os veículos não apresentem futuros contatos após estes

saírem da vizinhança dos veículos de interesse. Caso contrário, geraria-se trajetórias descontínuas, o que resultaria nos problemas de “saída e retorno” discutidos.

A função `Is_VI` realiza a filtragem dos veículos de interesse e suas respectivas vizinhanças para este módulo e está descrita no Algoritmo 3. O único par de comparações exigidas para determinar se a entrada de um veículo v deve ser mantida no traço está na linha 2 do algoritmo. Nele compara-se se o veículo v pertence ao conjunto de veículos de interesse \mathcal{V}_i ou à vizinhança de um veículo de interesse $\mathcal{N}(\mathcal{V}_i)$.

Algoritmo 3 Função para detecção de veículos de interesse e suas respectivas vizinhanças.

```

1: função 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

```

3.3.4 Filtragem por rastreamento de veículos

Na ausência de suporte a teleporte de nós da rede, como é o caso de grande parte dos simuladores de rede como o ns-3 e o OMNET++, um nó que saia da área de interesse de uma filtragem tradicional não possui suas entradas mantidas no traço. Dessa forma, este nó deve ser desativado para evitar interações que não ocorreriam no caso real. Caso o nó retorne, de forma mais simples, este deve ser configurado como um novo nó. Caso, porém, deseje-se que haja uma continuidade com o seu comportamento antes do nó sair da área de interesse, é preciso identificar o nó desativado referente a este e transferir seu estado para o novo nó criado.

Esse módulo contorna o problema de “saída e retorno” de veículos da área de interesse descrito na Seção 3.2. Isso é alcançado usando os veículos rastreados definidos na Seção 3.1.1. Após tornar-se vizinho de um veículo de interesse v_i ou infectado v_s , o veículo v_r torna-se rastreado e passa a ter todas as suas entradas mantidas no traço, mesmo que este saia da vizinhança de um veículo de interesse ou da área de interesse. Dessa forma, sua trajetória não se torna descontínua e ele não é ignorado ou configurado como um novo nó pelo exportador de traços `traceExporter.py`, por exemplo, caso retorne à área de interesse.

A solução aplicada por este módulo é simples e não requer a integração entre a ferramenta de filtragem e o programa de simulação de rede. Tal integração seria necessária no caso de soluções que envolvessem o desativamento temporário dos nós de mobilidade quando esses estivessem fora da área de interesse, informação essa não contida no traço de mobilidade. Sendo assim, novos nós não são gerados após a reentrada do veículo na área de interesse e mantém-se um número reduzido de nós

na pilha de nós simulados da rede.

A Figura 3.2 ilustra o uso deste módulo de filtragem. Note que apenas as posições ocupadas pelo nó rastreado v_r são mantidas no traço. Em contrapartida, os nós de interesse v_i e infectados v_s têm, além de suas posições, as suas vizinhanças mantidas no traço.

A função `IS_RV` que identifica e atualiza os veículos rastreados é descrita no Algoritmo 4. Os veículos rastreados vão sendo adicionados ao conjunto \mathcal{V}_r , já que todos devem ser mantidos no traço final filtrado. Por isso, 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 ao fim de seu percurso e, portanto, pode ser retirado de \mathcal{V}_r para liberar memória (essa operação é realizada nas linhas 7 a 9 do Algoritmo 4). 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 5).

Algoritmo 4 Função de identificação e atualização dos veículos rastreados.

```

1: função 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:      $\mathcal{V}_r \leftarrow \mathcal{V}_r + \{v\}$  ▷ Atualiza-se  $\mathcal{V}_r$ 
6:     retorne verdade
7:    $\forall v \in \mathcal{V}_r$  faça
8:     se  $v \notin T(t)$  então
9:        $\mathcal{V}_r \leftarrow \mathcal{V}_r - \{v\}$  ▷ Libera memória
10:  retorne falso

```

3.3.5 Filtragem por infecção de interesse

Aplicando a filtragem por veículos de interesse e rastreados a partir dos veículos identificados como origem de transmissões permite incluir todas as transmissões de um salto a partir desses que possam ocorrer durante a simulação. Conhecendo os veículos destinatários de mensagens da rede e configurando estes como veículos rastreados, é possível levar em conta todas as transmissões de até dois saltos (do veículo remetente para seu vizinho, e deste para o veículo destinatário). Entretanto, roteamento de dados envolvendo um número maior de saltos, ou simulações envolvendo veículos de destino desconhecidos que não entram na vizinhança dos veículos transmissores requerem um novo método de filtragem além dos introduzido nos módulos anteriores. Isso ocorre pois os nós que realizam encaminhamento nem sempre se tornam vizinhos dos nós de origem ou destino.

Para incluir, no traço final, todos os possíveis destinatários e nós envolvidos no roteamento de mensagens da rede, um algoritmo epidêmico baseado em contato é considerado. Para isso, determina-se que veículos que entrem na vizinhança de um veículo de interesse passem a se comportar como veículos de interesse, podendo infectar outros veículos que entrem em contato e assim por diante. A infecção é limitada por um número máximo de saltos (ρ_{max}) a partir do veículo de interesse original pois, caso contrário, a filtragem incluiria muitos nós, reduzindo sua eficácia. A ideia desta filtragem é garantir que quaisquer veículos roteadores e receptores de mensagens dentro do alcance de ρ_{max} saltos a partir dos veículos de interesse iniciais estejam presentes no traço final.

A Figura 3.3 ilustra a filtragem deste módulo. Na Figura 3.3(a), o veículo v_s é infectado e passa a se comportar como um veículo de interesse, após entrar na vizinhança de v_i ($v_s \in \mathcal{N}(v_i)$). Já na Figura 3.3(b), mostra-se que o veículo v'_s também deve se comportar como um veículo de interesse após entrar em contato com v_s (caso ρ_{max} permita). Conseqüentemente, 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 5. O número de saltos de infecção restantes a partir de cada veículo infectado v_s é monitorado por uma variável ρ_{v_s} correspondente que é armazenada no conjunto \mathcal{J} . Caso um veículo entre na vizinhança de um veículo de interesse v_i , ele torna-se um veículo infectado v_s e seu número de saltos de infecção restantes ρ atualizado para o valor de ρ_{max} e adicionado ao conjunto \mathcal{J} (operação feita pelas linhas 2 a 5). Se um veículo infectado não se encontra no instante de tempo sendo analisado $T(t)$, conclui-se que este chegou ao fim de seu percurso e, portanto, pode ser retirado de \mathcal{V}_s para liberar memória (essa operação é realizada nas linhas 7 a 9). Se um veículo qualquer se encontra na vizinhança de um veículo infectado, determina-se o maior valor de ρ possível que este pode obter e este se torna um veículo infectado com o respectivo valor de ρ se $\rho \neq 0$ (essa operação é realizada nas linhas 10 a 20). As linhas 11 a 14 determinam o valor de ρ que o veículo já possui, e caso esse não seja um veículo infectado, seu ρ é igual a 0. As linhas 15 a 17 determinam, dentre os veículos infectados que podem infectar o veículo v , aquele que possui o maior valor de ρ para comparar com o valor de ρ de v . Se o valor de ρ encontrado para v for maior do que o valor que este possui, este é atualizado; e caso o valor de ρ encontrado for 0 e v não for um veículo infectado, a função retorna “verdadeiro” (sua entrada é mantida no traço) e este não se torna um veículo rastreado. As linhas 18 a 20 são responsáveis por atualizar os conjuntos \mathcal{J} e \mathcal{V}_s . Como os veículos infectados devem ser todos mantidos no traço final, esses vão sendo adicionados ao conjunto \mathcal{V}_s que deve ser passado por referência para a função `Is_II`.

Algoritmo 5 Função de identificação e atualização dos veículos infectados.

```
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:      $\mathcal{J} \leftarrow \mathcal{J} + \{\rho_v\}$            ▷ Atualiza-se o número de saltos restantes referentes a  $v$ 
5:      $\mathcal{V}_s \leftarrow \mathcal{V}_s + \{v\}$        ▷ Adiciona-se  $v$  ao conjunto de veículos infectados
6:     retorne verdade
7:    $\forall v \in \mathcal{V}_s$  faça
8:     se  $v \notin T(t)$  então
9:        $\mathcal{V}_s \leftarrow \mathcal{V}_s - \{v\}$            ▷ Libera memória
10:  se  $v \in \mathcal{N}(\mathcal{V}_s)$  então
11:    se  $v \in \mathcal{V}_s$  então
12:       $\rho_v = \text{GET\_SALTOS}(v, \mathcal{J})$ 
13:    senão
14:       $\rho_v = 0$ 
15:    para todo  $v_s \in \mathcal{V}_s$  tal que  $v \in \mathcal{N}(\mathcal{V}_s)$  faça
16:      se  $\text{GET\_SALTOS}(v_s, \mathcal{J}) > \rho_v$  então           ▷ Recebe o número de saltos de infecção
                                                                restantes referentes a  $v_s$ 
17:         $\rho_v \leftarrow \text{GET\_SALTOS}(v_s, \mathcal{J}) - 1$        ▷ Atualiza-se o maior valor já obtido
18:      se  $\rho_v \neq 0$  então
19:         $\mathcal{J} \leftarrow \mathcal{J} + \{\rho_v\}$            ▷ Atualiza-se o número de saltos restantes referentes a  $v$ 
20:         $\mathcal{V}_s \leftarrow \mathcal{V}_s + \{v\}$        ▷ Atualiza-se o conjunto de veículos infectados
21:      retorne verdade
22:    retorne falso
```

3.3.6 Filtragem por infecção de interesse retro-temporal

A filtragem por contato visa considerar todos os nós do traço que podem receber pacotes ou participar no seu roteamento por múltiplos saltos, podendo até mesmo incluir os seus destinos, mesmo quando estes não são identificados a priori. De forma análoga, quando os nós destinatários de pacotes são conhecidos, mas os de origem não são, torna-se necessário fazer uma filtragem onde a infecção de interesse por contato ocorra para trás no tempo (retro-temporalmente). Essa filtragem pode ser realizada lendo o traço de amostragem temporal no sentido inverso e lidando com a infecção de interesse, a partir dos veículos de destino, de forma igual a como essa ocorreria no módulo de infecção de interesse. 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.

Por ser completamente independente dos demais módulos, o módulo de infecção de interesse retro-temporal pode ser executado paralelamente aos demais módulos de filtragem propostos. O algoritmo deste módulo analisa o arquivo de traço no sentido inverso do tempo de forma análoga ao Algoritmo 5, aplicando as mesmas funções de filtragem. Dessa forma, as únicas mudanças em relação ao tratamento do Algoritmo 5 seriam a inicialização de $t = t_f$ (linha 1 do Algoritmo 1) e o decremento de t que deve receber instante anterior ao atual (linha 11 do Algoritmo 1).

No Algoritmo 1 do filtro proposto, a função `IS_RT` do módulo é aplicada de forma sequencial aos demais módulos a pesar de poder ser executada paralelamente. Neste caso, as entradas dos veículos que foram infectados retro-temporalmente e suas vizinhanças são adicionadas a um traço temporário T_{rt} que deve ser unificado com o traço filtrado de forma direta antes de retornar o traço de saída do filtro.

O Algoritmo 6 usa o mesmo pseudocódigo do Algoritmo 5, mas deve usar conjuntos de saltos restantes \mathcal{J}^{rt} e conjunto de veículos infectados \mathcal{V}_s^{rt} distintos da filtragem feita de forma direta. Como esses grupos são comuns ao Algoritmo 1 estes devem ser distintos para impedir a transição de veículos infectados diretamente para veículos infectados retro-temporalmente.

Algoritmo 6 Função de identificação e atualização dos veículos infectados retro-temporalmente.

```

1: função IS_RT( $v, \mathcal{V}_i, \&\mathcal{V}_s^{rt}, T_{rt}(t), \&\mathcal{J}^{rt}, \rho_{max}$ )
2:   se  $v \in \mathcal{N}(\mathcal{V}_i)$  então
3:      $\rho_v \leftarrow \rho_{max} - 1$ 
4:      $\mathcal{J}^{rt} \leftarrow \mathcal{J}^{rt} + \{\rho_v\}$            ▷ Atualiza-se o número de saltos restantes referentes a  $v$ 
5:      $\mathcal{V}_s^{rt} \leftarrow \mathcal{V}_s^{rt} + \{v\}$            ▷ Adiciona-se  $v$  ao conjunto de veículos infectados
6:     retorne verdade
7:    $\forall v \in \mathcal{V}_s^{rt}$  faça
8:     se  $v \notin T_{rt}(t)$  então
9:        $\mathcal{V}_s^{rt} \leftarrow \mathcal{V}_s^{rt} - \{v\}$            ▷ Libera memória
10:  se  $v \in \mathcal{N}(\mathcal{V}_s^{rt})$  então
11:    se  $v \in \mathcal{V}_s^{rt}$  então
12:       $\rho_v = \text{GET\_SALTOS}(v, \mathcal{J}^{rt})$ 
13:    senão
14:       $\rho_v = 0$ 
15:    para todo  $v_s \in \mathcal{V}_s^{rt}$  tal que  $v \in \mathcal{N}(\mathcal{V}_s^{rt})$  faça
16:      se  $\text{GET\_SALTOS}(v_s, \mathcal{J}^{rt}) > \rho_v$  então           ▷ Recebe o número de saltos de infecção
                                                                restantes referentes a  $v_s$ 
17:         $\rho_v \leftarrow \text{GET\_SALTOS}(v_s, \mathcal{J}^{rt}) - 1$            ▷ Atualiza-se o maior valor já obtido
18:      se  $\rho_v \neq 0$  então
19:         $\mathcal{J}^{rt} \leftarrow \mathcal{J}^{rt} + \{\rho_v\}$            ▷ Atualiza-se o número de saltos restantes referentes a  $v$ 
20:         $\mathcal{V}_s^{rt} \leftarrow \mathcal{V}_s^{rt} + \{v\}$            ▷ Atualiza-se o conjunto de veículos infectados
21:      retorne verdade
22:    senão
23:      retorne falso

```

3.4 Discussão sobre o uso dos módulos de filtragem

Os módulos de filtragem acima definidos foram desenvolvidos para operarem em conjunto com a possibilidade de se desativar alguns módulos de acordo com o nível de filtragem que se deseja obter. Nesta seção se discute a melhor forma como os módulos de filtragem devem ser configurados pelo usuário para que a filtragem atenda os requisitos de traços de seu caso de uso.

Como já abordado anteriormente, uma das preocupações do uso de algoritmos epidêmicos em filtragem de traços é que esses tendem a cobrir grande parte dos traços em poucos saltos de infecção. Essa característica pode ser mitigada com o uso conjunto dos módulos de filtragem por infecção tradicional e retro-temporal. Realizando a filtragem epidêmica a partir da fonte, e do destino no sentido inverso, é possível cobrir os caminhos entre o veículo remetente e destinatário de pacotes de forma que a infecção a partir da origem e do destino possuam cada uma, apenas metade do número de saltos do caminho completo. O conceito pode ser observado na Figura 3.6. Configurando o valor máximo de saltos de infecção ρ_{max} como 1 salto de infecção, e usando os módulos de rastreamento de veículos e infecção retro-temporal, o filtro proposto consegue garantir a cobertura de qualquer caminho entre o veículo fonte e destino de até cinco saltos. Para que a mesma garantia fosse obtida pela infecção de interesse a partir de uma única extremidade (ou pela fonte ou pelo destino), seria necessário que o valor máximo de saltos de infecção ρ_{max} fosse no mínimo três com a outra extremidade sendo um veículo rastreado. Como o número de veículos potencialmente cobertos por um algoritmo epidêmico cresce quadraticamente em relação ao número de saltos, duas infecções com um número menor de saltos de infecção tende a cobrir um número menor de veículos. O conceito pode ser visualizado como um raio de influência centrado no veículo de interesse cuja área cresce de forma quadrática em relação ao seu raio (número de saltos). Essa configuração serve de exemplo de como diferentes métodos de filtragem podem ser combinados através de seus módulos dentro da estrutura de filtragem proposta para a obtenção de filtros mais eficazes.

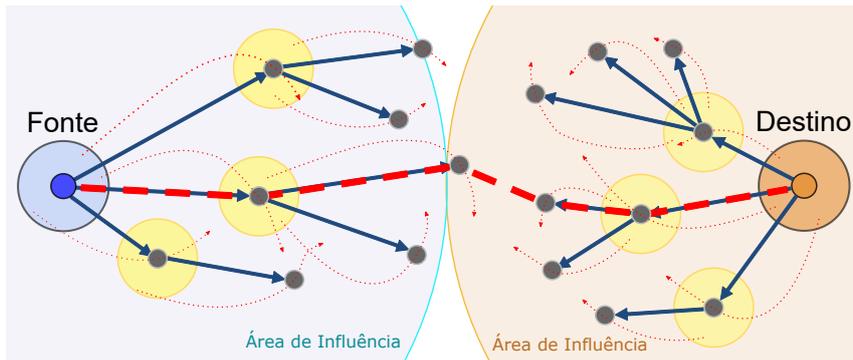


Figura 3.6: Representação da infecção por dupla extremidade para amenizar a cobertura de algoritmo epidêmico.

A Tabela 3.2 lista as simulações ou casos de uso vislumbrados para cada método de filtragem proposto. Nessa tabela, **AI** se refere à filtragem pelo módulo de área de interesse do filtro proposto; **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 da área ou período de tempo selecionado;

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 retrotemporal com infecção de interesse e podendo apresentar veículos rastreados; e por fim, **II+RT** refere-se à filtragem por dupla extremidade onde se combina a filtragem feita pelo método **II** a partir dos veículos de interesse fontes de dados com a filtragem feita por **RT** a partir dos veículos de interesse consumidores de dados.

Tabela 3.2: Comparação entre os módulos de filtragem de traços.

Filtragem	Caso de Uso
AI / 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.

3.5 Filtro implementado

Para realizar a filtragem usada na validação do método de filtragem proposto, um filtro foi programado seguindo a estrutura modular delineada na Figura 3.5 e adaptando o algoritmo de cada módulo descrito na Seção 3.3 em linguagem `awk`. O código de filtragem `awk` é contido em um *script bash* que utiliza as funções `gawk`, `cut`, `sed`, `cat`, e `tac` como dependências. Além disso, o filtro usa a ferramenta embutida da linha de comando `getopts` para obter opções de filtragem fornecidas pelo usuário e instruir o código `awk`.

As opções da ferramenta de filtragem criada fornecem, ao usuário, controle sobre seus módulos de filtragem. Essas permitem realizar a filtragem proposta neste tra-

balho, assim como as filtrações tradicionais já encontradas. Abaixo é descrito cada opção do filtro implementado, seu uso e sua utilidade.

-v [ARQUIVO/TEXTO]: permite que se dê os identificadores dos veículos de interesse a serem usados nos módulos de filtração por infecção de interesse, rastreamento de veículos e infecção retro temporal. Os veículos fornecidos são usados por todos os módulos de filtração, portanto é necessário ter cuidado na escolha dos veículos. Os identificadores podem ser fornecidos em um arquivo `.csv`, ou de forma direta entre aspas com os identificadores de veículos separados por espaço.

-k [ARQUIVO/TEXTO]: permite ao usuário fornecer identificadores de veículos a serem rastreados ao longo do traço. Os veículos fornecidos são tratados como veículos rastreados e não afetam a adição de novos veículos rastreados pelos outros módulos de filtração. Pode ser usado para fornecer ao filtro nós da rede que são de importância e que não devem ser retirados do traço. Como, por exemplo, nós de infra-estrutura, roteamento, ou nós de destinatários de transmissões. De forma idêntica à opção **-v**, os identificadores podem ser fornecidos em um arquivo `.csv`, ou de forma direta entre aspas com os identificadores de veículos separados por espaço.

-d [distance]: permite que o usuário defina o raio usado para a definição da vizinhança dos veículos de interesse e infectados. É aconselhável que seja usado um raio de vizinhança que englobe o alcance máximo de transmissão esperado para os nós da rede a ser simulada.

-c: é uma *flag* que, quando ativa, substitui a área considerada vizinha de um nó de um círculo para um quadrado de largura duas vezes maior que o valor de `distance` centrado nos veículos de interesse e infectados. Isso é usado para acelerar o processo de filtração pelo filtro. Como o esforço computacional para determinar se um ponto pertence a uma área retangular é menor do que o esforço para se determinar se este pertence a um círculo, a filtração é realizada de forma mais rápida.

-b "x1 y1 x2 y2": é uma opção para que o usuário configure a área de interesse desejada. A área de interesse é definida por um retângulo e a entrada desta opção deve ser dada no formato "x1 y1 x2 y2", onde (x1,y1) define o ponto inferior esquerdo, e (x2,y2) o superior direito do retângulo da área de interesse. Caso não seja dado nenhum argumento a esta opção, a área de interesse é determinada de forma automática sendo definida como a menor área retangular que englobe os trajetos dos veículos de interesse juntamente com suas respectivas vizinhanças (definidas pelo valor de `distance`). Essa opção permite não apenas a determinação da área de interesse, mas também permite delimitar a região que englobe todos os veículos de interesse apenas através de seus identificadores.

-t "BEGIN END": permite ao usuário a determinação dos instantes de tempo mínimo BEGIN e máximos END a serem considerados para fins de filtração. Qualquer entrada

do traço a ser filtrado referente a um instante de tempo $t < \text{BEGIN}$ ou $t > \text{END}$ é desconsiderada do traço. Se não forem definidos valores para **BEGIN** e **END**, apenas os instantes de tempo do traço em que veículos de interesse estejam presentes serão considerados.

-i: opção que habilita o módulo de rastreamento de veículos. Após essa opção ser habilitada, qualquer veículo que entre na vizinhança de um veículo de interesse ou infectado passa a ser rastreado até que este complete a sua trajetória.

Caso “-1” seja dado como argumento a essa opção, veículos fora do retângulo delimitador da área de interesse do traço são ignorados a não ser que estes sejam um dos veículos rastreados fornecidos à ferramenta pela opção **-k** ou um veículo de interesse. Sendo assim, a filtragem do traço por área de interesse passa a ser feita usando uma delimitação rígida semelhante à filtragem tradicional que não rastreia veículos que saiam da área de interesse.

-s: é uma opção que habilita o módulo de filtragem por infecção de interesse. Faz com que veículos que entrem na vizinhança dos veículos de interesse tornem-se veículos infectados, e que estes infectem outros veículos seguindo o valor de ρ_{max} fornecido pela opção **-j**.

-j [# max_jumps]: define o número máximo de saltos que uma infecção de interesse pode realizar definida como ρ_{max} nesse trabalho. Seu valor padrão é 1.

-a: é a opção que habilita a filtragem por dupla extremidade. Na implementação deste filtro, o traço a ser filtrado é lido de trás para frente e veículos que entrem em contato com veículos de interesse tornam-se veículos infectados retro-temporalmente. Após chegar à primeira linha do traço, os veículos infectados retro-temporalmente são configurados como veículos rastreados e o traço é filtrado no sentido normal do tempo segundo as opções configuradas. Isso é feito para unificar os traços de filtragem direta e retro-temporal.

-z: é uma opção usada para padronizar os instantes de tempo da simulação de redes do traço. Esta desloca todas as marcações dos instantes de tempo do traço de saída pelo mesmo valor de forma que o instante de tempo inicial do traço ocorra no instante de 0 segundos da simulação.

-o [filename]: Renomeia o arquivo de saída do filtro implementado.

Todas as opções são fornecidas à ferramenta através da linha de comando. A ferramenta pode ser encontrada em github.com/ericpso/FCD_Filter e sua documentação e vídeo demonstrando o seu uso estão disponíveis em www.gta.ufrj.br/FCDfilter. O manual de uso da ferramenta está incluído no Apêndice A.

Capítulo 4

Simulação

O traço de mobilidade a ser usado na análise do filtro proposto foi gerado de forma sintética usando o cenário TAPAS Cologne [9] do SUMO. A geração do modelo de mobilidade do cenário usado é feita com base em estudos do fluxo habitual de veículos pela cidade de Colônia na Alemanha. Este compreende o período de 6 às 8h da manhã com granularidade de 1 segundo entre entradas consecutivas do mesmo veículo, e possui tamanho de 10,1 GB.

A ferramenta de filtragem descrita na Seção 3.5, a qual segue o modelo de filtragem proposto, e o filtro `traceExporter.py` do SUMO foram usados para gerarem os traços de entrada do simulador de rede. O ns-3 foi escolhido como simulador de rede por se tratar de um simulador a eventos discretos amplamente usado. Como desejasse trabalhar com simulações de longo tempo de processamento e grande consumo de memória, este também foi escolhido por aparentemente possuir um consumo de memória e tempo de processamento menor dentre os simuladores de rede normalmente utilizados: OMNet++, ns-2, TOSSIM, e J-sim [25]. Escolheu-se uma rede tolerante a atrasos (DTN) como modelo de rede para qual o método de filtragem é avaliado. Pelo método de filtragem ser baseado em contato com rastreamento da trajetória de veículos, esse é ideal para redes DTN, que seguem o mesmo princípio de contato oportunístico. Além disso, a validade do método de filtragem para redes DTN deve garantir sua validade para outros métodos de roteamento de dados em redes que não suportam contatos momentâneos. Um modelo de atraso de propagação de velocidade constante e um modelo determinístico de perda, onde a transmissão é garantida de ser bem sucedida dentro de um raio a partir do transmissor, são usados para os dados transmitidos. O comportamento determinístico da perda de dados garante a reprodutibilidade dos cenários simulados, e permite a comparação de roteamento de dados bem sucedidos em simulações de pequena escala, quando há poucas transmissões.

Após a simulação, os resultados são comparados segundo o consumo de memória, tamanho do traço de mobilidade, número de nós da rede simulada, tempo total de simulação da rede, e número bem sucedido de entrega de pacotes pela rede. Gráficos do tempo simulado da rede em função do tempo real de computação gasto são criados para fins de comparação.

Todas as simulações de redes veiculares foram executadas em um único processo ocupando constantemente dentre 99% a 100% de uma CPU física. Foram feitas simulações em pequena e larga escala. Para as simulações em pequena escala, usou-se uma máquina de CPU Intel Xeon E5-2650 operando a 2.00 GHz. Já para as simulações em larga escala, usou-se uma máquina de CPU Intel Core i7-4930K operando a 3.40 GHz. Ambas possuíam memória suficiente para a realização de suas operações para evitar a ocorrência de *swap*, o que poderia afetar o tempo de simulação.

4.1 Cenários de filtragem avaliados

A região urbana referente ao centro da cidade de Colônia é escolhida como área de foco do traço para as simulações de rede. Essa região apresenta uma densidade maior de veículos cujas velocidades de circulação são menores em relação ao resto do traço.

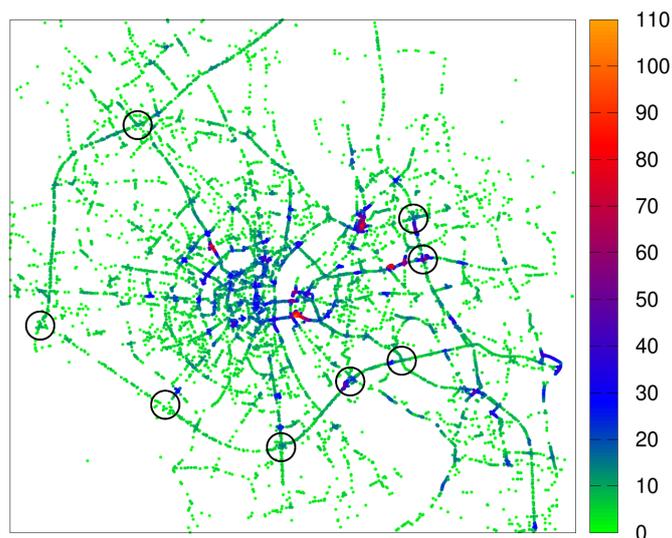


Figura 4.1: Gráfico do grau dos veículos do cenário de Colônia codificados por cor.
Fonte: [26]

Veículos clientes e servidores são selecionados dentro da área utilizada como cenário de simulação. Após um transiente inicial da simulação, no qual veículos ainda estão iniciando suas trajetórias, seleciona-se, aleatoriamente, veículos segundo dois critérios. Os veículos selecionados devem circular durante todo o período da simulação e não podem deixar a região urbana do traço. A área de interesse usada na filtragem é definida pela menor área retangular que engloba todas as trajetórias e vizinhanças dos veículos servidores e clientes. Para a simulação de rede, o alcance máximo usado no modelo de transmissão determinística de sinais foi de 300 metros, sendo esse um alcance típico para transmissões desobstruídas em aplicações V2V

(veículo a veículo) [27]. A distância da filtragem por contato, “ d ”, escolhida foi de 305 metros. Seu valor foi escolhido com base no alcance máximo configurado da transmissão de pacotes da rede, adicionando-se uma margem de erro para levar em conta a flutuação da precisão devido à amostragem a cada 1 segundo e tempo de propagação do sinal.

Inicialmente, realizou-se simulações em larga escala, filtrando-se o intervalo de tempo de todo o trajeto dos veículos clientes e servidores. Em seguida, simulações em pequena escala foram feitas, onde se limitou o intervalo de simulação a 2 minutos. Devido ao tempo de simulação maior para as simulações em larga escala, os veículos de interesse cobrem uma área maior. Isso força a área de interesse a ser mais extensa para cobrir todos os seus trajetos. Consequentemente, as áreas de interesse dos cenários em pequena escala cobrem, em média, apenas 1,8 km², enquanto os cenários em grande escala cobrem, em média, 81 km². Durante a simulação de rede, cada um dos veículos servidores tenta transmitir um pacote de 1kB para cada um dos veículos clientes usando roteamento epidêmico de redes DTN. As tentativas de transmissão são feitas apenas uma vez (sem tentativas de retransmissão) a cada 5 segundos nos cenários em pequena escala, e a cada segundo nos cenários em grande escala.

Para evitar a inundação de mensagens nos cenários em larga escala, foi usado o protocolo “Spray and Wait” [15] com limite de 16 pacotes por mensagem. Este protocolo consiste no compartilhamento do serviço de disseminação de mensagens com os nós de roteamento. Isto é, metade do limite de mensagens é disseminado para o próximo nó de roteamento para que esse auxilie no processo de disseminação. Como exemplo, no caso estudado, ao final desse primeiro contato cada veículo ficaria com 8 pacotes para disseminar. Esse processo se repete recursivamente para cada nó de roteamento até cada nó possuir apenas uma mensagem a ser entregue ao destinatário.

As simulações em larga escala foram feitas para determinar o ganho de escala obtido pela aplicação da filtragem, e em pequena escala, para avaliar o funcionamento do método de filtragem proposto. As simulações em larga escala envolvem a simulação de 1 servidor para 1, 2, e 5 clientes. Enquanto isso, as simulações em pequena escala usam pares de 1 servidor para 3 clientes, 3 servidores para 3 clientes, e 3 servidores para 6 clientes.

Na seção de resultados, a filtragem realizada pela ferramenta `traceExporter.py` é referida por **STD**. Esta emprega a filtragem tradicional por corte geográfico retangular com limites rígidos para sua área de interesse. Para a filtragem **STD**, qualquer entrada de um veículo que saia da área de interesse é desconsiderada, mesmo que este retorne a ela. A filtragem **STD** é usada como referência para os resultados obtidos pelas simulações que usam traços filtrados pelo modelo proposto. A simula-

ção feita a partir do traço filtrado pelo método proposto é classificada segundo sua configuração e módulos ativos. Essas configurações são:

- **RV** para filtragem rastreamento de veículos; filtragem na qual os veículos servidores atuam como veículos de interesse e os veículos que entram em suas vizinhanças são rastreados;
- **II** para a filtragem por infecção de interesse de um salto incluindo rastreamento de veículos. Essa usa os veículos servidores configurados como veículos de interesse a partir dos quais realiza-se a infecção de interesse em apenas um salto ($\rho_{max} = 1$). O módulo de rastreamento de veículos é usado para evitar trajetórias descontínuas;
- **RT** para filtragem com infecção de interesse retro-temporal. Essa é realizada a partir dos veículos clientes, que são definidos como veículos de interesse dos quais realiza-se a infecção de interesse de apenas um salto; e por fim,
- **II+RT** é a configuração de filtragem por dupla extremidade através do rastreamento de veículos. Nessa, ambos os veículos clientes e servidores são configurados como veículos de interesse e a infecção ocorre em apenas 1 salto de infecção a partir destes de forma direta para os servidores e retro-temporalmente para os clientes.

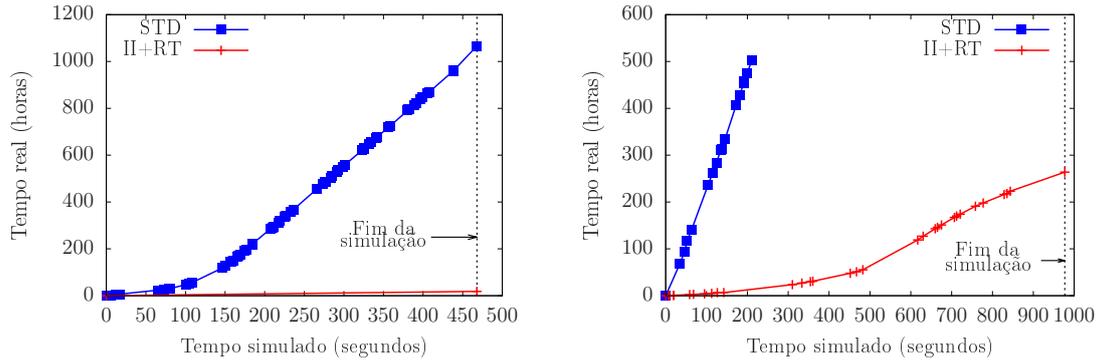
O efeito do uso de limites flexíveis na determinação da área de interesse é analisado para os cenários em pequena escala. Para isso, os cenários simulados a partir dos traços resultantes da filtragem **STD** são designados como “Bound Box”. Estes são comparados contra os cenários simulados a partir dos traços filtrados pelo módulo de área de interesse, designados por “Fuzzy Box”, e que apresentam limites de área de interesse flexíveis. Espera-se que as simulações resultantes do uso de áreas com limites flexíveis obtenham melhores resultados quando comparados às simulações de áreas de limites rígidos. Devido ao módulo de filtragem por área de interesse solucionar o problema de “saída e retorno”, essas devem apresentar uma maior conectividade. Enquanto isso, as simulações com limites de área de interesse rígidos desconsideram veículos que retornem à área de interesse. Logo, esses devem apresentar uma conectividade menor devido à perda de nós de roteamento que retornam à área de interesse.

4.2 Análise dos resultados

4.2.1 Simulações em grande escala

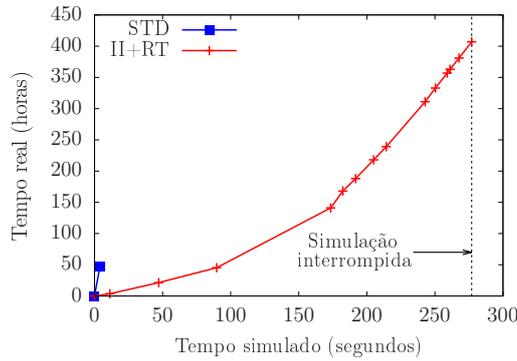
Inicialmente, são simulados os cenários em larga escala. Devido ao grande tempo de simulação desses cenários, decidiu-se comparar apenas as filtrações **STD** e **II+RT**. A filtração **II+RT** foi escolhida devido a esperar-se que esta apresente o melhor rendimento de qualidade de conectividade da rede em relação ao custo computacional.

A Figura 4.2 ilustra a relação entre o tempo da simulação e o tempo real gasto para simular a rede veicular em larga escala. Pode-se também observar que, nos três casos de estudos avaliados, as simulações feitas usando traços filtrados pelo método proposto convergem em tempo consideravelmente menor do que as filtradas pelo método tradicional. Todas as simulações que utilizaram a filtração padrão foram interrompidas antes de terminarem por não terem expectativas de convergência em intervalos de semanas e por estarem usando uma quantidade muito grande de memória demonstrando sua baixa escalabilidade. Comparando as Figuras 4.2(a), 4.2(b) e 4.2(c) nota-se que, com o aumento do número de clientes, a situação fica mais crítica já que mais entradas devem ser mantidas no traço.



(a) Simulação de 1 servidor para 1 cliente.

(b) Simulação de 1 servidor para 2 clientes.



(c) Simulação de 1 servidor para 5 clientes.

Figura 4.2: Relação entre o tempo simulado em função do tempo real para as simulações de rede DTN em larga escala usando método de filtragem **II+RT**.

A Tabela 4.1 relaciona os dados de cada um dos três casos de estudo em larga escala de acordo com o método de filtragem usado (**STD** ou **II+RT**). A tabela mostra o número de nós, o tempo simulado, e o tamanho do arquivo de mobilidade gerado pelas duas filtrações para cada um dos casos de simulação. Imediatamente, é possível perceber uma grande diferença em termos do número de veículos removidos, redução do tamanho e tempo de simulação dos traços. Os ganhos obtidos são justificados, principalmente, pela retirada de nós do traço da rede que foram julgados não participantes da simulação segundo o filtro proposto. Já as simulações realizadas a partir de traços filtrados pelo método tradicional não convergiram mesmo após longos períodos de tempo. Portanto, a comparação entre o número de transmissões e entregas de pacotes para os métodos de filtragem foram feitas com base em resultados parciais das simulações.

Os resultados parciais da Tabela 4.1 demonstraram que o número de entregas foi diferente para a filtragem **STD** e a filtragem **II+RT** para o cenário de apenas um par cliente-servidor. O número de entregas bem sucedidas obtido pela simulação que usou a filtragem proposta foi maior do que o obtido pela simulação que usou a filtragem tradicional. Deduz-se portanto, que a técnica de filtragem proposta em

Tabela 4.1: Comparação entre filtragens para os três casos de estudo de DTN em grande escala.

Filtragem	Nº de nós	Tamanho do traço	Tempo de simulação	Tempo simulado	Nº parcial de transm.	Nº parcial de entregas
1-to-1 STD	2657	25,9MB	1066h	468s	468	11
1-to-1 II+RT	531	11,7MB	18h	468s	468	35
1-to-2 STD	9.258	169,3MB	503h	211s	422	0
1-to-2 II+RT	1.381	49,2MB	264h	977s	422	0
1-to-5 STD	34.358	1,1GB	47h	4s	20	0
1-to-5 II+RT	3.952	218,6MB	407h	277s	20	0

II+RT conseguiu melhorar a conectividade da rede. Além disso essa soluçõa o problema de “saída e retorno” da rede o que pode justificar o número maior de entregas com sucesso. Os resultados também demonstraram a inviabilidade de se simular redes veiculares desta escala sem o uso de técnicas que reduzam seu esforço de simulação. As simulações do filtro proposto conseguiram alcançar tempos de simulação maiores com todas obtendo um número de entregas bem sucedidas (191 entregas para 1-to-2 II+RT, e 14 entregas para 1-to-5 II+RT). Resultados estes que não seriam possíveis obter através de simulações de traços filtrados de forma convencional. Além disso, essas simulações demonstram o sucesso do filtro proposto em tornar escalável a simulação de redes em grande escala.

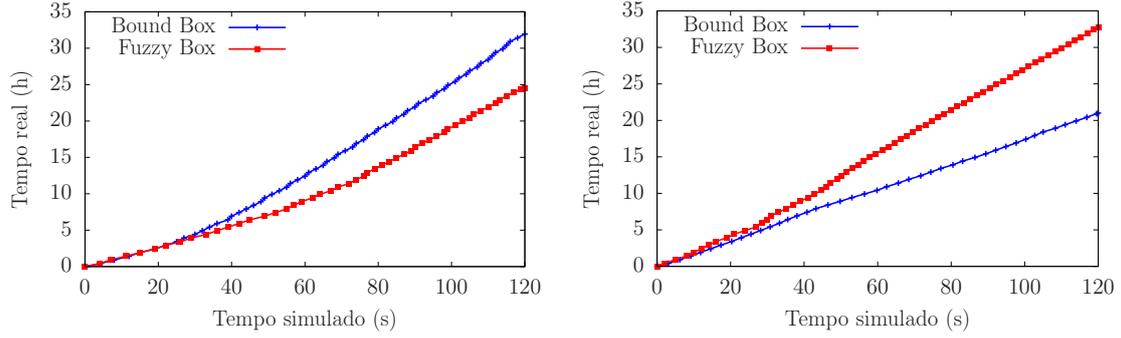
4.2.2 Simulações em pequena escala

Uma vez vista a validade do filtro em cenários em grande escala, são feitas simulações em pequena escala para avaliar detalhes de seu comportamento e seu ganho em outros cenários.

Inicia-se a análise em pequena escala pela comparação entre as duas formas de filtragem mais gerais. Os resultados da filtragem geográfica com limites fixo (*Bound Box*) e com limites flexíveis (*Fuzzy Box*) para os três cenários em pequena escala são apresentados na Figura 4.3.

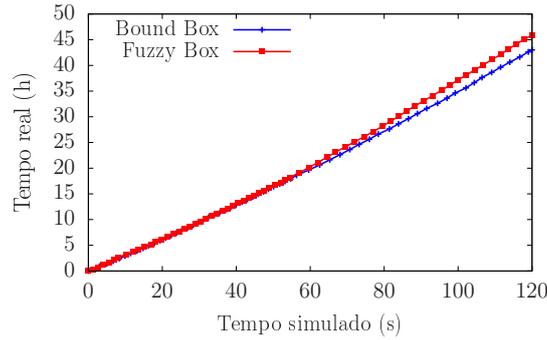
Na Figura 4.3, observa-se que em duas das três simulações a filtragem pelo filtro proposto resultou em um aumento do tempo de simulação. Esse resultado era esperado, uma vez que o simulador deve simular o deslocamento de veículos que saiam da área de interesse devido a estes serem rastreados. Esse deve também simular suas interações quando estes retornam à área de interesse e passam a interagir com outros veículos.

A Tabela 4.2 compara os resultados obtidos pela filtragem tradicional e pelo módulo de filtragem por área de interesse proposto para simulações de rede DTN em pequena escala. É possível observar que o número de nós da rede é idêntico para os traço filtrados de ambas maneiras. Esse resultado é esperado devido a cada



(a) Cenário com 1 servidor e 3 clientes.

(b) Cenário com 3 servidores e 3 clientes.



(c) Cenário com 3 servidores e 6 clientes.

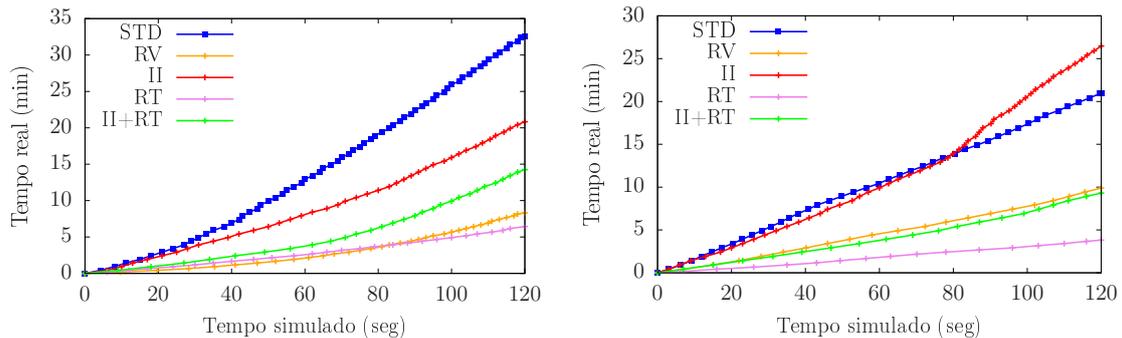
Figura 4.3: Comparação temporal entre simulações resultantes do uso de traços com limites fixos ou flexíveis para a área de interesse.

veículo que entra na área de interesse ser adicionado por ambos os traços. Entretanto, as simulações realizadas a partir da filtragem proposta obtiveram um traço de tamanho maior, usaram mais memória e tiveram tempos de simulação majoritariamente maiores. Esses resultados devem-se ao uso de rastreamento de veículos pela filtragem por área de interesse, feita para solucionar o problema de descontinuidade

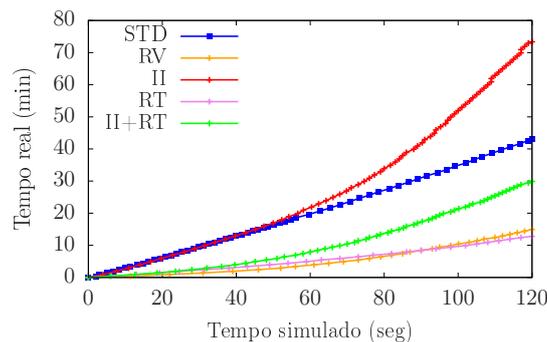
Tabela 4.2: Comparação entre filtragens para os três casos de estudo de DTN em pequena escala.

Filtragem	Nº de nós	Tamanho do traço	Memória máx. usada	Tempo de simulação	Nº de transm.	Nº de entregas
1-to-3 Fuzzy Box	830	4,4MB	3,58GB	24h 33min	72	11
1-to-3 Bound Box	830	3,6MB	3,32GB	31h 57min	72	3
3-to-3 Fuzzy Box	835	4,6MB	3,28GB	32h 44min	215	54
3-to-3 Bound Box	835	3,9MB	3,08GB	21h	215	14
3-to-6 Fuzzy Box	972	5,2MB	3,93GB	45h 55min	432	83
3-to-6 Bound Box	972	4,3MB	3,73GB	43h	432	70

de trajetórias (problema de “saída e retorno”). Como veículos continuam sendo simulados mesmo ao saírem da área de interesse, esses levam a consumos maiores de memória, processamento, e tamanho de arquivo de traço. Os custos adicionais de processamento e memória podem, entretanto, serem justificados pela obtenção de uma rede veicular mais realista devido à sua solução do problema de “saída e retorno” de veículos pelo filtro proposto como pode ser observado no maior número de entregas de pacotes bem sucedidas.



(a) Comparação para 1 servidor e 3 Clientes. (b) Comparação para 3 servidores e 3 clientes.



(c) Comparação para 3 servidores e 6 clientes.

Figura 4.4: Comparação temporal entre simulações de traços resultantes de diferentes modos de filtragem em pequena escala.

Por fim, os resultados dos tempos de simulação das filtrações por rastreamento de veículo **RV**; Infecção de interesse de um único salto com rastreamento de veículo **II**; infecção de interesse retro-temporal de um único salto **RT**; e infecção de interesse de um salto por dupla extremidade **II+RT**, são mostrados na Figura 4.4.

Dentre os módulos de filtragem propostos, é possível observar uma tendência. A filtragem por infecção de interesse com rastreamento de veículos **II** tende a requisitar um maior tempo de simulação dentre as filtrações propostas, superando a simulação resultante da filtragem tradicional em alguns casos após um período de tempo. A filtragem **II** demora um tempo para superar a filtragem tradicional. Isso ocorre pois, diferentemente da filtragem por área de interesse, a filtragem de infecção por contato entre veículos demora até cobrir grande parte dos veículos da área de interesse.

Ponto no qual os veículos que saem da área de interesse começam a fazer diferença no processamento da simulação em relação à filtragem tradicional.

Tabela 4.3: Comparação entre filtragens para os três casos de estudo de DTN em pequena escala.

Filtragem	Nº de nós	Tamanho do traço	Memória máx. usada	Tempo de simulação	Nº de transm.	Nº de entregas
1-to-3 STD	830	3,6MB	3,32GB	31h 57min	72	3
1-to-3 RV	285	1,4MB	896MB	8h 19min	72	3
1-to-3 II	828	4,4MB	3,6GB	20h 50min	72	4
1-to-3 RT	412	1MB	1,1GB	6h 27min	72	5
1-to-3 II+RT	487	2,2MB	1,74GB	14h 15min	72	3
3-to-3 STD	835	3,9MB	3,08GB	21h	215	14
3-to-3 RV	405	2MB	1,37GB	9h 54min	215	32
3-to-3 II	834	4,5MB	3,3GB	26h 48min	215	30
3-to-3 RT	305	756,1kB	764MB	3h 49min	215	50
3-to-3 II+RT	443	2,1MB	1,51GB	9h 19min	215	39
3-to-6 STD	972	4,3MB	3,73GB	43h	432	70
3-to-6 RV	327	1,5MB	990MB	14h 56min	432	31
3-to-6 II	971	5,1MB	3,95GB	73h 22min	432	33
3-to-6 RT	493	1,1MB	1,22GB	12h 48min	432	22
3-to-6 II+RT	526	2MB	1,64GB	29h 55min	432	29

A Tabela 4.3 é referente à simulação da Figura 4.4 dos diversos modos de filtragem propostos e da filtragem tradicional em cenário em pequena escala. Analisando os resultados da tabela, algumas observações interessantes podem ser feitas. As simulações dos filtros propostos obtiveram ganhos no número de entregas de mensagens bem sucedidas comparáveis ou melhores à filtragem tradicional para os casos de 1 servidor para 3 clientes e 3 servidores para 3 clientes. Em todas as simulações foi possível obter reduções no consumo de memória, número de nós da rede, e tempo de simulação, com a exceção de **II**. Por **II** se tratar de uma filtragem epidêmica sem o uso da técnica de cobertura por dupla extremidade como a filtragem **II+RT**, essa tende a cobrir todos os nós da rede de forma rápida. Esse comportamento pode ser inferido pelo número final de nós da rede que a filtragem conseguiu incluir. A amenização dessa característica da filtragem de infecção de interesse **II** pode ser observada nos resultados da tabela para a filtragem por dupla extremidade **II+RT**, a qual foi configurada para garantir cobrir todos os caminhos de mesmo número de saltos (através de 1 salto de rastreamento a partir de cada extremidade ao invés de um salto de infecção e um de rastreamento). Essa obteve, consistentemente, um consumo de memória menor, um menor tamanho de traço e menor tempo de simulação.

A análise dos caminhos das mensagens bem sucedidas para o cenário de 3 servidores para 6 clientes revelou que grande parte dos caminhos de roteamento encontrados que não foram descobertos pela filtragem proposta envolveram caminhos

com número grande de saltos (por volta de 20). Estes caminhos foram, portanto, inviáveis para o filtro proposto descobrir, resultando em um número menor de entregas de mensagens bem sucedidas. Esses caminhos poderiam ser cobertos com a determinação de um número máximo de saltos de infecção de interesse (ρ) maior. Entretanto, mesmo em baixa escala, a determinação do número máximo de saltos de infecção elevado geraria uma sobrecarga de processamento extra para as operações de infecção tornando a filtragem proposta por área de interesse a mais viável.

Outra observação com base na Tabela 4.3 foi a de que a filtragem usando o módulo de infecção retro-temporal para o cenário de 3 pares de clientes-servidores simulou um número maior de entregas de pacotes com sucesso do que a filtragem por infecção de interesse mesmo que a segunda filtragem garantisse a cobertura de um número maior de saltos. Isso é possível caso haja uma baixa simetria da rede que faça com que os nós em contato com os nós clientes participem majoritariamente do roteamento de mensagens. Esse resultado, em conjunto com os caminhos longos de roteamento do cenário de 3 servidores para 6 clientes, revelam que a escolha da melhor combinação de configurações dos módulos de filtragem depende dinamicamente da rede. Entretanto, como a predeterminação do tamanho dos caminhos de roteamento a serem tomados e a previsão da dinâmica da rede são difíceis de serem predeterminados, sua escolha pode ser muito difícil.

Sendo assim, após considerar os resultados obtidos na comparação dos filtros propostos e na comparação das filtrações por áreas de interesse de limites rígidos (filtragem estática) e flexíveis (com rastreamento), concluiu-se que: para cenários em pequena escala vale a pena o uso do módulo proposto de filtragem por área de interesse.

Capítulo 5

Conclusão

Esse trabalho surgiu da busca pela obtenção de dados realistas referentes a redes veiculares de grande escala e de dinâmicas complexas para o estudo de VANETs e protocolos de comunicação veiculares. Obtenção de dados a partir de testes reais se mostraram impraticáveis. Porém, mostrou-se que resultados bastante confiáveis poderiam ser obtidos a partir de simulações de rede veiculares. Uma primeira análise de simulações de redes veiculares mostrou que essas não escalam para o número grande de nós desejável na simulação de veículos por uma cidade. Além disso, estudos em escala menor se mostraram com perda de acurácia devido à deficiência do ferramental de filtragem em considerar as particularidades dos simuladores de rede a eventos discretos. O trabalho, aqui apresentado, veio de encontro às necessidades de simulação acima descritas. Com o intuito de tornar as simulações de redes veiculares mais escaláveis, foi apresentado um modelo modular de filtragem de traços de mobilidade baseado em contato. O modelo apresentado soluciona o problema de descontinuidade de trajetórias de mobilidade, aqui caracterizado como problema de “saída e retorno”, sofrido por outras ferramentas de filtragem estática.

O uso de veículos rastreados na filtragem por área de interesse solucionou o problema de “saída e retorno”. Esse também apresentou um número maior de entregas bem sucedidas de pacotes em sua simulação com pouca variação relativa no tempo de simulação da rede e consumo de memória. Seus resultados superiores foram atribuídos à solução do problema de trajetórias descontínuas por este considerar os veículos que retornaram à área de interesse e, portanto, aumentar a concentração de veículos próxima à sua região limite de filtragem.

Também foi possível observar como a técnica de infecção por dupla extremidade foi capaz de reduzir o número de nós, consumo de memória, e tamanho dos arquivos de traço quando comparada com a infecção de interesse por uma única extremidade. Ambas obtiveram resultados comparáveis para o número de entregas bem sucedidas de pacotes em cenários em pequena escala quando configuradas para cobrir todos os caminhos de até 3 saltos.

Os resultados também mostraram como a qualidade da simulação varia de acordo com a filtragem e casos a serem estudados. O modelo de filtragem do trabalho, para as configurações estudadas, apresentou resultados melhores em alguns casos de es-

tudo e piores em outros para as simulações em pequena escala. Portanto, concluiu-se que, para simulações em pequena escala, é preferencial o uso exclusivo do módulo de filtragem por área de interesse, pois consegue-se resolver o problema de “saída e retorno” da área de interesse com baixos custos computacionais adicionais. Esses resultados também incentivam estudos futuros para a determinação da melhor configuração a ser usada para cada traço de mobilidade. Novos testes com diferentes configurações podem demonstrar como a dinamicidade da rede pode afetar na efetividade dos módulos de filtragem.

Os resultados para traços maiores consolidam a necessidade do uso de técnicas de filtragem de traços e otimização das simulações de rede para que as simulações sejam escaláveis. Quando apenas técnicas tradicionais de filtragem são aplicadas, a simulação não converge em tempo hábil. O problema de “saída e retorno” ainda se mostrou presente nos resultados de sua simulação, como pode ser observado na entrega bem sucedida de mensagens na simulação que fez uso do filtro dinâmico por contato. Mesmo com uma área de interesse maior, a falta de veículos retornando à área de interesse impediu a formação de rotas de mensagens entre os veículos servidores e clientes.

Finalmente, conclui-se que a filtragem dinâmica por contato proposta teve sucesso em tornar as simulações mais escaláveis. Os resultados da filtragem proposta apresentaram taxas de sucesso de entrega de pacotes maiores para consumos de memória e tempos de simulação comparativamente menores. Ainda mais, a filtragem proposta abre novas vias de estudo para a geração de opções de filtragem dinâmica de traços mais complexas.

5.1 Trabalhos futuros

Pelo trabalho aqui apresentado se tratar de um modelo de filtragem modular, este incita a adição de novos módulos de filtragem que incluam novos casos de uso, como a adição de filtragem de pedestres ou módulos que filtrem uma classificação nova de veículo. Além disso, os resultados obtidos neste trabalho incentivam que simulações adicionais sejam feitas para se determinar a melhor configuração dos módulos de filtragem a ser usada para cada cenário a ser simulado a partir do traço de mobilidade.

A ferramenta de filtragem desenvolvida neste trabalho foi feita em linguagem `awk`, que limita futuros desenvolvimentos de estruturas de filtragem mais complexas. Por essa razão, é pretendido a conversão do código do filtro para uma linguagem de programação mais flexível. Também busca-se tornar o código de filtragem mais eficiente aplicando paralelismo com base na divisão da filtragem do traço no eixo temporal.

Referências Bibliográficas

- [1] CAMERON F. KERRY, J. K. “Gauging investment in self-driving cars”. The Brookings Institution. 16 de outubro de 2017. Disponível em: <<https://www.brookings.edu/research/gauging-investment-in-self-driving-cars/>>. [Online; acessado em 3 de abril 2019].
- [2] LITMAN, T. *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, Canada, 2017.
- [3] HALIM, Z., KALSOOM, R., BASHIR, S., et al. “Artificial intelligence techniques for driving safety and vehicle crash prediction”, *Artificial Intelligence Review*, v. 46, n. 3, pp. 351–387, 2016.
- [4] JIANG, D., DELGROSSI, L. “IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments”. Em: *VTC Spring 2008-IEEE Vehicular Technology Conference*, pp. 2036–2040. IEEE, 2008.
- [5] UZCÁTEGUI, R. A., DE SUCRE, A. J., ACOSTA-MARUM, G. “Wave: A tutorial”, *IEEE Communications magazine*, v. 47, n. 5, pp. 126–133, 2009.
- [6] FERNANDES, R., FERREIRA, M. “Scalable VANET Simulations with NS-3”. Em: *IEEE VTC Spring*, pp. 1–5, maio 2012.
- [7] FERNANDES, R., VIEIRA, F., FERREIRA, M. “VNS: An integrated framework for vehicular networks simulation”. Em: *IEEE VNC*, pp. 195–202, nov. 2012.
- [8] UPPOOR, S., TRULLOLS-CRUCES, O., FIORE, M., et al. “Generation and Analysis of a Large-Scale Urban Vehicular Mobility Dataset”, *IEEE Transactions on Mobile Computing*, v. 13, n. 5, pp. 1061–1075, maio 2014.
- [9] TAPAS COLOGNE. “Data/Scenarios/TAPASCologne”. Acessado em <http://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne>, mar. 2018.
- [10] KRAJZEWICZ, D., ERDMANN, J., BEHRISCH, M., et al. “Recent Development and Applications of SUMO - Simulation of Urban MOBility”, *International Journal On Advances in Systems and Measurements*, v. 5, n. 3&4, pp. 128–138, dez. 2012.

- [11] NS-3. “Network Simulator 3”. 2018. Disponível em: <<https://www.nsnam.org/>>. [Online; acessado em 17 de julho 2018].
- [12] OMNET++. “OMNeT++: Discrete Event Simulator”. 2018. Disponível em: <<https://www.omnetpp.org/>>. [Online; acessado em 17 de julho 2018].
- [13] MUSSA, S. A. B., MANAF, M., GHAFOOR, K. Z., et al. “Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives”. Em: *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–8, out. 2015.
- [14] VAHDAT, A., BECKER, D., OTHERS. *Epidemic routing for partially connected ad hoc networks*. Technical Report CS-200006, Duke University, 2000.
- [15] SPYROPOULOS, T., PSOUNIS, K., RAGHAVENDRA, C. S. “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks”. Em: *ACM SIGCOMM Workshop on Delay-tolerant Networking (WDTN 2005)*, pp. 252–259, 2005.
- [16] FALL, K. “A Delay-tolerant Network Architecture for Challenged Internets”. Em: *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2003)*, pp. 27–34, 2003.
- [17] PERRONE, L. F., NICOL, D. M. “Using n-body algorithms for interference computation in wireless cellular simulations”. Em: *Proceedings 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (Cat. No. PR00728)*, pp. 49–56. IEEE, 2000.
- [18] FUJIMOTO, R. M., PERUMALLA, K., PARK, A., et al. “Large-scale network simulation: how big? how fast?” Em: *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003.*, pp. 116–123. IEEE, 2003.
- [19] BARNES, J., HUT, P. “A hierarchical O (N log N) force-calculation algorithm”, *nature*, v. 324, n. 6096, pp. 446, 1986.
- [20] SALOMON, D. *Data compression*. Springer, 2004.
- [21] DOUGLAS, D. H., PEUCKER, T. K. “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature”, *Cartographica: the international journal for geographic information and geovisualization*, v. 10, n. 2, pp. 112–122, 1973.

- [22] POPA, I. S., ZEITOUNI, K., ORIA, V., et al. “Spatio-temporal compression of trajectories in road networks”, *GeoInformatica*, v. 19, n. 1, pp. 117–145, 2015.
- [23] SUMO TOOLS. “TraceExporter”. Acessado em <http://sumo.dlr.de/userdoc/Tools/TraceExporter.html>, mar. 2018.
- [24] ERIC P. S. O., M. E. M. C. “Um Método de Filtragem de Traços para Simulações Escaláveis de Redes Veiculares”. Em: *VIII Brazilian Symposium on Computing Systems Engineering*, pp. 1–8, Salvador, Brasil, nov. 2018.
- [25] KHAN, M. A., HASBULLAH, H., NAZIR, B. “Recent open source wireless sensor network supporting simulators: A performance comparison”. Em: *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 324–328. IEEE, 2014.
- [26] LOULLOUDES, N., PALLIS, G., DIKAIAKOS, M. D. “The dynamics of vehicular networks in large-scale urban environments”. Em: *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*, pp. 192–199. IEEE, 2015.
- [27] WANG, S., HUANG, A., ZHANG, T. “Performance Evaluation of IEEE 802.15.4 for V2V Communication in VANET”. Em: *2013 International Conference on Computational and Information Sciences*, pp. 1603–1606. IEEE, 2013.

Apêndice A

Uso do Filtro Implementado

A configuração padrão da ferramenta usa distância radial de 500 metros ao redor dos veículos de interesse e não usa nem rastreamento, nem infecção de interesse. Quando nenhum veículo de interesse é fornecido, apenas a filtragem em área e por intervalo de tempo são executadas, caso os argumentos apropriados forem fornecidos.

O uso da ferramenta na linha de comando segue o formato:

```
./filterFCD.sh [OPÇÕES...] [ARQUIVO]
```

O resultado da filtragem de traço é salva em um arquivo separado em `./filtered/` com o prefixo `filtered_` anexado. As opções da ferramenta são fornecidas abaixo.

- h:** exibe uma página de ajuda com as opções da ferramenta.
- v [ARQUIVO/TEXTO]:** fornece os identificadores dos veículos de interesse através de um arquivo `.csv`, ou através de um texto entre aspas com os nomes dos veículos de interesse separados por espaço.
- k [ARQUIVO/TEXTO]:** fornece os identificadores dos veículos a serem rastreados ao longo do traço através de um arquivo `.csv`, ou através de um texto entre aspas com os nomes dos veículos de interesse separados por espaço.
- d [distance]:** define o valor da distancia (`distance`) a ser usada para delimitar a vizinhança ao redor dos veículos de interesse.
- c:** substitui o uso de distância radial ao redor do veículo de interesse para definir sua vizinhança por um quadrado de largura duas vezes maior que o valor de `distance` centrado nos veículos de interesse. Isso é usado para acelerar o processo de filtragem.
- b "x1 y1 x2 y2":** delimita uma região geográfica de interesse retangular no traço com base nas coordenadas do ponto inferior esquerdo (`x1,y1`) e superior direito (`x2,y2`) do retângulo. Apenas veículos que passam por essa região são considerados para a saída da filtragem. Caso o argumento "`x1 y1 x2 y2`" não seja fornecido, a área ideal de filtragem é definida como sendo a menor área retangular que englobe os trajetos dos veículos de interesse juntamente com suas respectivas vizinhanças (definidas pelo valor de `distance`).

- t "BEGIN END":** filtra apenas instantes de tempo entre BEGIN e END. Se não forem definidos valores para BEGIN e END, serão considerados apenas os instantes de tempo em que veículos de interesse estejam circulando no traço.
- i:** habilita a filtragem com rastreamento de veículos. Caso "-1" seja dado como argumento, veículos fora do retângulo delimitador do traço são ignorados a não ser que este seja um dos veículos rastreados fornecidos à ferramenta ou um veículo de interesse. Sendo assim, a filtragem do traço por área de interesse passa a ser feita usando uma delimitação rígida que não rastreia veículos que saiam desta.
- s:** habilita a filtragem por infecção de interesse.
- j [# max_jumps]:** define o número máximo de saltos que uma infecção de interesse pode realizar. O valor padrão é 1.
- a:** habilita a filtragem com infecção de interesse por dupla extremidade.
- z:** desloca toda as marcações dos instantes de tempo do traço de saída de forma que o instante de tempo inicial do traço ocorra no instante de 0 segundos da simulação.
- o [filename]:** Renomeia o arquivo de saída do filtro que é enviado para ./filtered/ como "filename".