

TrGeT: geração de traces para simulação de Redes Veiculares Ad Hoc por meio de Redes de Petri Coloridas

Vasconcelos, L.C.M¹, C. G. Furtado Júnior², Soares, J. M.¹, Barroso, G. C.³

¹Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará (UFC)

Campus do Pici - Bloco 725 - CEP 60455-970 - Fortaleza, CE – Brasil

²Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Av. Parque Central, S/N - Distrito Industrial I - Maracanaú/CE - CEP: 61.939-140

³Departamento de Física - Universidade Federal do Ceará (UFC)
Campus do Pici - Bloco 922 - CEP 60455-554 - Fortaleza, CE – Brasil.

luis.vasconcelos@sti.ufc.br, cjunior@ifce.edu.br, {marques, gcb}@ufc.br

Abstract. *In this work, we introduce the TrGeT tool developed in timed hierarchical coloured Petri nets that generates mobility traces to simulate VANETs in network simulators. The scenarios configuration for generation of traces is carried out in a simplified way and without the need of knowledge of traffic and urban mobility simulators. The performance evaluation of the tool shows the effectiveness of the generated mobility traces.*

Resumo. *Neste trabalho, apresentamos a ferramenta TrGeT desenvolvida em rede de Petri colorida hierárquica temporizada que gera traces de mobilidade para simulação de VANETs em simuladores de rede. A configuração dos cenários para geração dos traces é realizada de forma simplificada e sem necessidade de conhecimento de simuladores de tráfego e mobilidade urbana. A análise de desempenho da ferramenta mostra a eficácia dos traces de mobilidades gerados.*

1. Introdução

As redes veiculares *ad hoc* (VANET - *Veicular ad hoc networks*) são uma sub-classe das redes *ad hoc* móveis (MANET - *Mobile ad hoc network*). Elas representam um sistema de comunicação veicular que tem potencial para melhorar a segurança no trânsito por meio do compartilhamento de informações para a prevenção de acidentes, além de poderem auxiliar também na investigação pós-acidente. A realização de testes em cenários reais é uma tarefa árdua devido a questões como alto custo econômico e constantes mudanças na topologia da rede, fazendo com que a simulação seja considerada a melhor forma de se realizar a análise de desempenho em uma VANET [Al-Sultan et al. 2013].

Para avaliar o desempenho de um protocolo utilizado em uma VANET, é crítica a necessidade de um modelo de mobilidade, cujo principal papel é imitar o comportamento do movimento do móvel. O modelo de mobilidade deve refletir as restrições de movimentação dos veículos (*mobile hosts* - MH), incluindo os dados de modelos de tráfego. De acordo com a taxonomia utilizada em [Harri et al. 2009], os modelos de mobilidade podem ser classificados como: *traces* sintéticos baseados em pesquisa, *traces*

de aplicações reais e *traces* sintéticos gerados por simuladores de tráfego e mobilidade urbana. Como as VANETs ainda não são utilizadas em larga escala, existe dificuldade em se obter *traces* reais. Portanto, os *traces* sintéticos representam grande importância para a simulação das redes veiculares.

A utilização de ferramentas específicas de simulação de tráfego e mobilidade urbana, como o SUMO [Behrisch et al. 2011], demandam estudo de parâmetros do cenário de simulação como cruzamentos, junções, sinais de trânsito, diferentes tipos de vias e fluxos de trânsito. Para mitigar essa complexidade na configuração dos cenários de simulação, foram desenvolvidas ferramentas que permitem a geração de *traces* de forma rápida, como o *VanetMobiSim* e MOVE. Nessas ferramentas a configuração é simplificada por não levar em consideração parâmetros que não tenham relevância para geração de *traces* de mobilidade [Karnadi et al. 2007].

Nessas ferramentas não existe a opção de interagir com a simulação por meio da manipulação das variáveis envolvidas, como velocidade ou localização dos veículos, ou modificar o estado inicialmente configurado, como adicionar novos veículos. Neste trabalho apresentamos a ferramenta *TrGeT: Traces Generator Tool*, desenvolvida em rede de Petri colorida (RPC) hierárquica temporizada por meio da ferramenta *CPN Tools* [cpn]. Assim como as ferramentas *VanetMobiSim* e MOVE, *TrGeT* gera *traces* de mobilidade para simulação de VANETs de forma rápida. Como diferencial, no *TrGeT* é possível alterar o comportamento do modelo ao longo da simulação, permitindo o acompanhamento passo a passo (forma *stepwise*) da geração de *traces* para estudos e análises específicas. O *TrGeT* pode ser estendido e modificado por meio de mudanças no modelo RPC.

Este artigo é organizado da seguinte forma: Na Seção 2 são introduzidas as RPC. Na Seção 3 são apresentados os trabalhos relacionados. Na Seção 4 é tratado o modelo *Intelligent driver model* (IDM). Na Seção 5 a ferramenta *TrGeT* é detalhada. Na Seção 6 é apresentada uma simulação realizada com a ferramenta. Na Seção 7 é discutido o desempenho da simulação realizada. Finalmente, na Seção 8 são apresentados os resultados obtidos e os trabalhos futuros.

2. Redes de Petri coloridas

Redes de Petri (RP) são uma ferramenta de modelagem gráfica e matemática que permite descrever e estudar sistemas de processamento de informações que são caracterizados como sendo concorrentes, assíncronos, distribuídos, paralelos, não determinísticos, e/ou estocásticos. Redes de Petri coloridas (RPC), utilizadas neste trabalho, são uma linguagem gráfica para construção de modelos de sistemas concorrentes e análise de suas propriedades. RPC é uma linguagem de modelagem de eventos discretos que combina o potencial das redes de Petri com o potencial das linguagens de programação de alto nível [Jensen and Kristensen 2009].

Constituindo-se de um grafo bipartido com nós representando lugares (elipses) e transições (retângulos), ligados por arcos (fluxos orientados), o estado de um sistema é definido pela quantidade e tipo de fichas que são depositadas nos lugares. Nas RPC, as fichas depositadas em um lugar possuem tipo específico, denominado cor. Para simular o comportamento dinâmico de um sistema, os modelos de RPC alteram os valores e quantidades das fichas nos lugares a partir dos disparos das transições. Uma transição é dita habilitada, e pode ser disparada, caso haja uma quantidade de fichas nos lugares de

entrada que satisfaça as expressões dos seus respectivos arcos de entrada. Após o disparo de uma transição, fichas são removidas dos lugares de entrada e outras inseridas nos lugares de saída de acordo com as expressões dos arcos de entrada e saída, respectivamente, da referida transição. O detalhamento de simulações usando RPC é feito ao longo da apresentação do modelo contido na ferramenta *TrGeT*.

O modelo *TrGeT* foi desenvolvido na ferramenta *CPN Tools*. Para realizar uma simulação, além da parametrização inicial, é possível escolher a forma na qual a simulação será executada, que pode ser de duas formas: (i) *fast forward*, em que, como nas ferramentas convencionais, o usuário inicia a simulação e aguarda o resultado final; (ii) *stepwise*, na qual a simulação evolui passo a passo de acordo com a interação do usuário. Esta última forma permite a manipulação das variáveis envolvidas na simulação por meio da alteração da marcação das fichas nos lugares desejados, possibilitando levar a simulação a situações específicas. É possível, por exemplo, elevar a velocidade de um veículo para além da permitida para verificar o comportamento do modelo ou alterar sua localização. Na forma *stepwise* é também possível verificar determinadas regras de um modelo sem a necessidade de iniciar uma nova simulação e aguardar até que a mesma alcance o estado desejado. Com isso, a modificação e, conseqüentemente, a evolução do modelo pode ser feita de forma mais rápida.

3. Trabalhos relacionados

Os modelos de mobilidade veiculares podem ser classificados como macroscópicos e microscópicos. Os macroscópicos consideram valores como densidade e média das velocidades dos veículos, tempo total de viagem, chegadas e partidas de veículos em larga escala, tratando o tráfego veicular de acordo com a dinâmica dos fluidos. A abordagem microscópica considera cada veículo como uma entidade distinta, modelando o seu comportamento (aceleração, desaceleração, mudança de faixa) de forma mais precisa [Harri et al. 2009]. Os modelos de mobilidade baseados em simuladores de tráfego realizam uma modelagem microscópica. Neles são determinados os movimentos de cada veículo, podendo simular realisticamente infraestrutura de ruas, avenidas e estradas, bem como interações entre veículos.

SUMO (*Simulation of Urban MObility*) [Behrisch et al. 2011] é um simulador de tráfego rodoviário microscópico de código aberto projetado para lidar com grandes redes rodoviárias. Possui funcionalidades como: movimento veicular sem colisões, roteamento individual dos veículos, ruas com várias pistas com mudança de faixa. Porém, como afirmam [Karnadi et al. 2007], o esforço de escrever *scripts* de simulação, bem como aprender sobre os detalhes internos do simulador (parametrização do cenário de simulação com cruzamentos, conexões entre as ruas, sinais de trânsito, diferentes tipos de vias e veículos, fluxos de trânsito) torna demorado o processo de coletar *traces*. A utilização de ferramentas de simulação de tráfego e mobilidade urbana são indicados para *testbeds* de cenários que demandam a utilização de grandes redes rodoviárias e onde se deve considerar os múltiplos fatores envolvidos na mobilidade urbana com o maior nível de realismo possível. Para se obter uma avaliação rápida de protocolos de rede ou de ferramentas que exijam um *time to market* curto, as ferramentas de geração de *traces* de forma rápida são indicadas [D. Cabarkapa 2016].

MOVE (*MObility model generator for VEhicular networks*) [Karnadi et al. 2007]

gera modelos de mobilidade para simulações VANET. A ferramenta é integrada ao SUMO, funcionando o MOVE como *parser* para o SUMO. O MOVE gera *traces* de mobilidade que podem ser usados por ferramentas de simulação de rede, como o NS-2, e possui uma interface gráfica que permite gerar rapidamente cenários de simulação. Entretanto, uma limitação da ferramenta é a exigência de instalação da versão 0.12.3 do SUMO, sendo que a versão corrente (no momento em que esse trabalho está sendo escrito) é a 0.29. Apenas a versão 0.12.3 do SUMO é compatível com o MOVE, comprometendo possíveis atualizações da ferramenta, além de possuir uma instalação não trivial.

VanetMobiSim (*Vehicular Ad Hoc Networks Mobility Simulator*) [Härri et al. 2006] é um simulador de modelos de mobilidade desenvolvido em *Java*. O *VanetMobiSim* gera *traces* de movimento em diferentes formatos, suportando diferentes ferramentas de simulação para redes móveis, incluindo NS-2, GloMoSim e QualNet. *VanetMobiSim* apresenta modelos de movimentos automotivos, tanto em níveis macroscópicos como microscópicos. Permite importar mapas do banco de dados digital TIGER [tig] e adiciona suporte para estradas com várias pistas, fluxos direcionais separados, restrições de velocidade diferenciadas e sinais de trânsito nas interseções. Para que a ferramenta *VanetMobiSim* seja estendida ou modificada é necessário o conhecimento de sua estrutura interna, possuindo a ferramenta mais de 120 classes *Java* sem documentação para desenvolvedores (conta-se apenas com o manual do usuário).

Em [Khan et al. 2014] é apresentado um modelo de redes de Petri coloridas em que nós móveis se deslocam em um terreno bidimensional de forma aleatória, e de acordo com o modelo de mobilidade *Random Walk* (RW) [Camp et al. 2002], sem possuir o efeito de borda e decaimento de velocidade. Os autores afirmam que, até a época em que o trabalho foi desenvolvido, era a primeira implementação do RW usando RPC. Em [Khan et al. 2015] os autores criam em RPC o modelo SHER (*Show Home and Exclusive Regions*), no qual o modelo detalhado em [Khan et al. 2014] é estendido. De acordo com os autores, o modelo elimina as seis questões-chave dos modelos de mobilidade aleatória, isto é, paradas repentinas, movimentos sem memória, efeito de borda, dependência temporal da velocidade, dependência do tempo de pausa e decaimento da velocidade. Ambas as implementações geram modelos de mobilidade para serem utilizados em redes de malhas (*Wireless Mesh Network - WMN*) [Akyildiz et al. 2005], onde não é necessário modelar as restrições de mobilidade como ruas, avenidas e veículos, não sendo adequados para simulações de VANETs.

A ferramenta *TrGeT* possibilita a geração de *traces* para simulações de VANETs de forma rápida e com configuração do cenário da simulação simplificada, sem que seja necessário que o usuário tenha um conhecimento prévio de simulações de tráfego. Como pré-requisitos para a execução da simulação são necessários apenas os programas *CPN Tools* e *Java*, disponibilizados gratuitamente e com versões atualizadas. A ferramenta apresenta um grade potencial de ser estendida ou modificada já que seu núcleo foi desenvolvido de forma modular em RPC.

4. *Intelligent driver model* (IDM)

O modelo de mobilidade *Intelligent driver model* (IDM) [Kesting et al. 2010] pertence à classe de modelos microscópicos chamada de *car following models* (CFM). Nestes modelos a mobilidade de um veículo é adaptada de acordo com um conjunto de regras para

evitar contato com o veículo que está à frente. O modelo IDM mostra uma dinâmica coletiva livre de colisões, exibe propriedades de estabilidade controláveis e implementa uma estratégia de frenagem inteligente com transições suaves entre comportamento de aceleração e desaceleração. No algoritmo IDM, se não existir outro veículo à frente, um veículo segue livre até que alcance a velocidade máxima desejada, caso contrário, regras precisam ser obedecidas para evitar a colisão. O modelo foi desenvolvido para delinear o comportamento da condução humana. Portanto, possui os seguintes parâmetros comportamentais: velocidade desejada, aceleração, desaceleração confortável e tempo de segurança para avanço [Kesting et al. 2010].

A aceleração no modelo IDM é uma função contínua que incorpora diferentes modos de condução para todas as velocidades no tráfego de rodovia, assim como no tráfego urbano. Além da distância s para o veículo da frente (principal) e a velocidade atual v , o modelo IDM também leva em conta a diferença de velocidade do veículo principal (taxa de aproximação) $\Delta v = v - v_1$. A aceleração do modelo IDM é calculada de acordo com a equação 1.

$${}^{\alpha}IDM(s, v, \Delta v) = \frac{dv}{dt} = \alpha \left[1 - \left(\frac{v}{v_0} \right)^{\delta} \left(\frac{s^*(v, \Delta v)}{s} \right)^2 \right] \quad (1)$$

em que

$$s^*(v, \Delta v) = S_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}. \quad (2)$$

Na equação 1, v_0 representa a velocidade desejada, δ o expoente de aceleração livre (representa como a aceleração diminui com a velocidade), T o intervalo de tempo desejado entre os veículos, S_0 a distância de congestionamento, a a aceleração máxima, b a desaceleração confortável (ou desejada). Os parâmetros IDM v_0 , T , S_0 , a e b são empiricamente mensuráveis e têm valores realistas [Treiber et al. 2000].

5. *TrGeT: Traces Generator Tool*

TrGeT é uma ferramenta de geração rápida de *traces* para simulação de VANETs. O usuário pode visualizar o comportamento do modelo pela interação com a ferramenta e pode também alterar parâmetros do sistema durante a simulação. O modelo de mobilidade adotado foi o IDM, dado que o mesmo é um dos modelos mais utilizados em simulações de tráfego [Harri et al. 2009], mas é possível realizar mudanças no modelo de mobilidade utilizado e estender a ferramenta com a adição de novos modelos.

A arquitetura da ferramenta é apresentada na Figura 1. A ferramenta é composta por dois componentes principais: (i) o modelo RPC onde é implementada a lógica da simulação; (ii) o *parser* que transforma a ficha gerada pelo *CPN Tools* em *traces* que podem ser lidos por simuladores de rede.

5.1. Modelo RPC

O modelo *TrGeT* foi desenvolvido em RPC hierárquicas temporizadas por meio da ferramenta *CPN Tools* [cpn]. Redes hierárquicas permitem a construção de modelos com

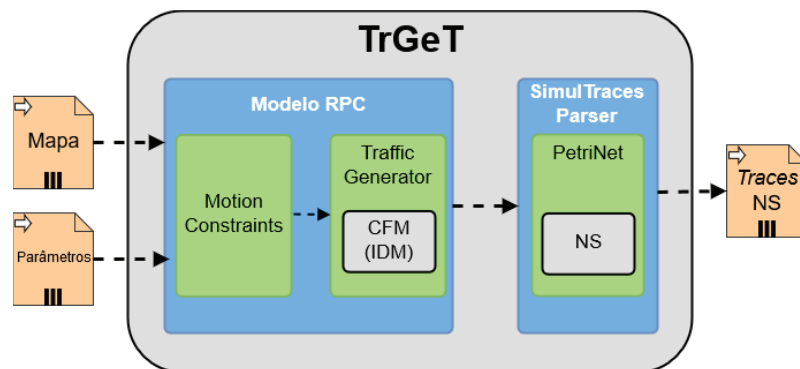


Figura 1. Arquitetura

múltiplos módulos organizados em diferentes níveis de abstração, como em linguagens de programação. A hierarquização é realizada em RPC, substituindo-se uma transição de substituição por uma sub-rede. Uma RPC temporizada possui, além das cores das fichas, um valor chamado de "selo de tempo" e um "relógio global". Isso permite que sistemas possam ser modelados e avaliados levando em consideração restrições de tempo. A definição formal das RPC hierárquicas temporizadas pode ser encontrada em [Jensen and Kristensen 2009].

O modelo RPC possui dois módulos assim denominados: *Motion Constraints* e *Traffic Generator*; *IDM* é um submódulo de *Traffic Generator* e representa a parte do modelo que pode ser estendida por meio da adição (ou substituição) de novos CFMs no mesmo nível hierárquico do submódulo *IDM*. A hierarquia é apresentada na Figura 2.

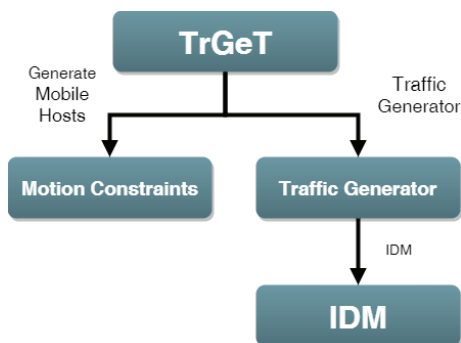


Figura 2. Hierarquia do modelo

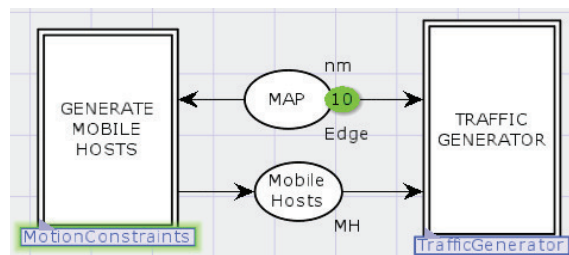


Figura 3. Página principal

Em RPC, uma sub-rede também é denominada página, acomodando-se, portanto, cada módulo do simulador em uma página. A página *TrGeT* apresentada na Figura 3 é a de maior nível hierárquico, correspondendo à página principal do modelo. Nela pode-se observar duas transições de substituição (TS) representadas por retângulos de borda dupla. A TS GENERATE MOBILE HOSTS recebe, por meio do arco de entrada, o mapa da área que será utilizada na simulação, e retorna, por meio do arco de saída, após executada a TS, um conjunto de *Mobile Hosts* (MHs), cada um com sua posição inicial, velocidade inicial zero e velocidade desejada escolhida aleatoriamente dentro das faixas definidas por meio de parâmetros de entrada. A TS TRAFFIC GENERATOR recebe pelos arcos de entrada o mesmo mapa e os MHs gerados anteriormente.

A estrutura e o comportamento das duas transições de substituição da página principal são descritas nas sub-redes *Motion Constraints* e *Traffic Generator*. A página *Motion Constraints*, é responsável pela geração dos MHs e por direcioná-los ao gerador de tráfego. A página *Traffic Generator* recebe os MHs e verifica se os mesmos estão próximos o suficiente de um cruzamento para determinar se mudam de via ou se continuam a se locomover na mesma via, obedecendo a um algoritmo CFM. No modelo *TrGeT*, o algoritmo escolhido foi o IDM, modelado na página IDM.

O mapa do cenário modelado é representado por meio do lugar *MAP* de cor *Edge* na página principal. A cor *Edge* é a representação de uma rua. Para modelar o mapa utilizamos dois conjuntos de cores: *Vertex* e *Edge*. O *Vertex*, representa uma coordenada (x, y) com um determinado *ID*. Um *Edge* é o equivalente a uma reta no plano cartesiano, ligada por dois pontos, no caso dois *Vertex*. O lugar *Mobile Hosts* de cor *MH* da página principal representa os veículos (*mobile hosts* - MH) presentes na simulação. Durante a simulação, um MH está em uma determinada rua e faixa com uma dada velocidade. O lugar MHs presente no submódulo *IDM* recebe fichas da cor *Platoon*, que é modelada como uma lista de MHs. O lugar é utilizado para tratar a posição em que um MH está em relação aos outros MHs na mesma rua e faixa que ele se encontra, de acordo com o algoritmo IDM [Treiber et al. 2000].

As funções utilizadas no modelo estão disponíveis no arquivo *functions.sml*, na pasta *declarations* do repositório GitHub [git]. Um exemplo é a função *addMovement*. Ela recebe um ponto $P(px, py)$ que representa a localização do MH, a rua, representada por meio da reta que passa pelos pontos $v1(x1, y1)$ e $v2(x2, y2)$, o deslocamento s que será realizado, a distância euclidiana n entre os pontos $v1$ e $v2$ (tamanho da rua) e a faixa em que o MH se encontra. A função adiciona a distância n ao ponto P em que o MH se encontra, e de acordo com a faixa da rua em que se encontra. Foi utilizado o sentido de circulação pela direita (mão francesa) em que os veículos trafegam pela faixa da direita, que no caso do modelo *TrGeT* é a *Lane 1*. Os MH que trafegam pela *Lane 1* se movem no sentido crescente do eixo x (abscissa) e y (ordenada), enquanto os que trafegam na *Lane 2* se movem no sentido decrescente dos eixos x e y . As coordenadas dos pontos são limitadas ao 1º quadrante do plano cartesiano.

5.2. SimulTraces Parcer

O módulo *SimulTraces Parcer* foi implementado em Java e transforma a saída da simulação RPC (marcação final gerada no *CPN Tools*) em *traces* de mobilidade que podem ser lidos por simuladores de rede. O módulo está preparado para adição de formato de novos simuladores, de acordo com o que foi apresentado na Figura 1. Atualmente apenas o simulador NS (NS-2 e NS-3) foi implementado.

6. Simulação

Para realizar uma simulação na ferramenta *TrGeT* é necessário configurar *a priori* a lista de parâmetros indicados na Tabela 1. Os valores configurados para estes parâmetros em nossa simulação estão listados na terceira coluna desta tabela.

Os parâmetros *numberMH* e t são inerentes à simulação. Os demais parâmetros são inerentes ao modelo IDM. Mais detalhes sobre a escolha dos valores para os parâmetros IDM e como estes interagem entre si podem ser encontrados em [Kesting et al. 2010].

Além dos parâmetros é necessário o mapa do local que será simulado. As coordenadas das ruas e dos cruzamentos dos mapas podem ser obtidos por meio de serviços de mapa como o **OpenStreetMap** [ope]. Porém, é necessário um tratamento posterior do arquivo gerado nesses serviços para transformá-lo em marcações de fichas que podem ser lidas pelo *CPN Tools*. As cores das fichas que representam um mapa são detalhadas na Seção 7. Foi utilizado um trecho da cidade de Málaga, Espanha, representado na Figura 4. As ruas e cruzamentos delimitados pela área mais clara foram utilizados na simulação. A sequência da simulação é apresentada nas próximas subseções.

Tabela 1. Parâmetros da simulação

| Parâmetro | Descrição | Valor |
|------------|---------------------------------|----------------------|
| $numberMH$ | Número de MH | 10 |
| t | Duração da simulação | 180 s |
| l | Tamanho do veículo | 5,0 m |
| a | Aceleração | 0,6 m/s ² |
| b | Desaceleração confortável | 0,9 m/s ² |
| T | Tempo de segurança entre os MHs | 0,5 s |
| S_0 | Distância mínima entre os MHs | 2,0 m |
| v_0 | Velocidade desejada | 13,88 m/s |
| v_{min} | Velocidade mínima | 2,78 m/s |
| $delta$ | Expoente de aceleração δ | 4,0 m/s ² |



Figura 4. Mapa

6.1. Página principal: coordenação da simulação no *TrGeT*

A coordenação de uma simulação é realizada pela execução sequencial de dois blocos. O primeiro é representado pela TS GENERATE MOBILE HOSTS. Na Figura 3 pode-se ver a página principal no estado inicial da RPC, na qual apenas a TS GENERATE MOBILE HOSTS está habilitada, o que é garantido pela disponibilidade de 10 fichas do tipo Edge no lugar *MAP*.

Após a execução da TS GENERATE MOBILE HOSTS, o lugar *Mobile Hosts* será alimentado por fichas do tipo MH, habilitando a TS TRAFFIC GENERATOR. Portanto, este segundo bloco recebe como valores de entrada tanto o Mapa (fichas do lugar *MAP*) como os MHs gerados no primeiro bloco (fichas do lugar *Mobile Hosts*). Os dois lugares da página principal do *TrGeT* correspondem a *sockets de RPC* que podem ser acessados nas sub-páginas pelas suas respectivas *ports*.

6.2. Página *Motion Constraints*: geração e posicionamento inicial de MHs no mapa

A página *Motion Constraints* apresentada na Figura 5 recebe, pela porta de entrada (*port in*) no lugar *Edges*, o mapa da simulação. Apenas a transição *Generate Map* estará habilitada na marcação inicial. Ela irá disparar até que todas as distâncias euclidianas n das fichas contidas no lugar *Edges* sejam calculadas e atualizadas.

O lugar *Available Spots* possui uma ficha da cor *Spots*, que representa uma lista de *Spot*. O conjunto de cor *Spot* representa a localização aleatória dentro do mapa em que um MH pode estar no início da simulação. A cada disparo da transição *Generate Map* é retirada uma ficha do lugar *Edges* e executada a função *makeSpots* que irá alterar a marcação do lugar *Available Spots*. Ao final desse processo, a marcação do lugar terá uma lista de todos os lugares disponíveis (*Spot*) para a alocação dos MHs.

Após a atualização de todas as fichas *Edge* e da lista de *Spot*, a transição *Generate New Hosts* é habilitada. Ela é responsável por selecionar aleatoriamente um *Spot* e

um veículo. A transição permanece habilitada até que todos os veículos sejam posicionados. A última transição da página *Motion Constraints* é a *Generate MHs*. O disparo da transição gera um veículo (MH) no lugar *Mobile Hosts* que funciona como uma porta de saída (*out port*).

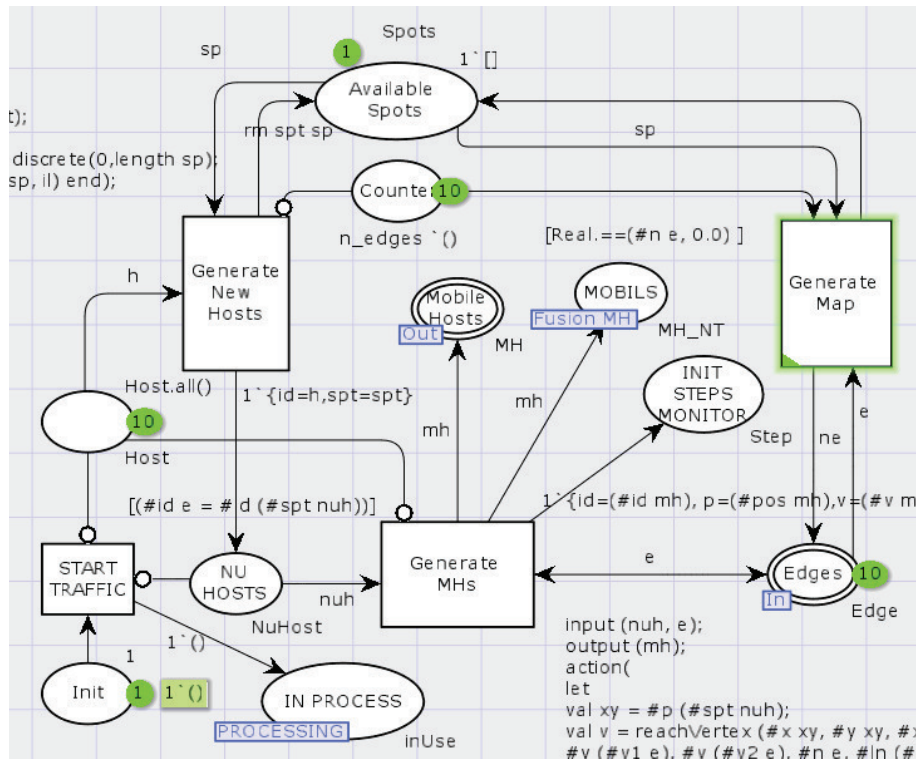


Figura 5. Página *Motion Constraints*.

6.3. Traffic Generator: gerador de movimentação dos veículos

As fichas chegam na página *Traffic Generator* por meio das portas de entrada (*in ports*) dos lugares *Map* e *Mobile Hosts* de acordo com a Figura 6. Esses lugares são interfaces com a página principal. A página *Traffic Generator* é responsável pela movimentação dos veículos pelas ruas, identificando se o mesmo deve permanecer na mesma ou chegou a um cruzamento.

A página possui três transições: *TO INTERSECTION*, *CHANGE STREET* e *IDM*. De acordo com a velocidade que o MH se move é calculada a distância percorrida em 1 s, tempo que dura um passo de simulação (*step*). Se a distância a ser percorrida no passo for maior do que a distância para alcançar o fim da rua o veículo deve se preparar para chegar a um cruzamento, habilitando a transição *TO INTERSECTION*. Caso contrário, o veículo permanecerá na rua e habilitará a transição *IDM*, responsável pelo *Car Following Model IDM*.

A transição *TO INTERSECTION* seleciona um MH apto a mudar de rua. A transição *CHANGE STREET* recebe o MH e verifica a rua (*Edge*) e faixa (*Lane*) que o mesmo se encontra. Na função *changeStreet* é verificado quais ruas se ligam ao cruzamento que o MH se encontra e aleatoriamente é escolhida uma delas. Então, a marcação do MH é atualizada com sua nova localização, velocidade e tempo de simulação. Para

modelar o comportamento real do condutor ao se deparar com um cruzamento, foi adicionada uma probabilidade de 50% para que o veículo pare (velocidade zero) ou continue sua viagem com a mesma velocidade.

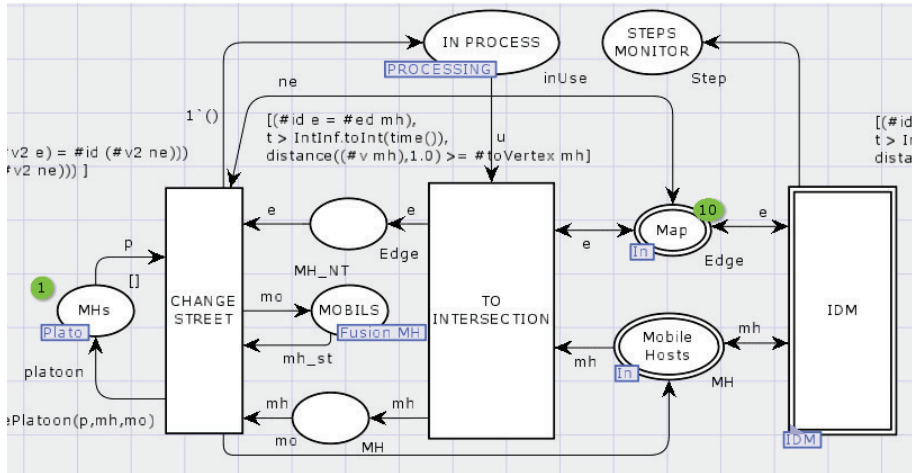


Figura 6. Página *Traffic Generator*

6.4. IDM: car following model

A página IDM controla a movimentação dos veículos que permanecem na mesma faixa e rua. É importante ressaltar que o módulo IDM pode ser substituído por outro módulo que implemente um algoritmo CFM, devido à forma hierárquica que a ferramenta *TrGeT* foi concebida. Para tal, basta que a nova página tenha as mesmas *ports* de entrada e saída que a página IDM tem.

Na sub-página IDM temos as transições *SELECT ONE MH*, *MAKE PLATOON*, *FREE TO GO* e *STOP AND GO* de acordo com a Figura 7. A transição *SELECT ONE MH* seleciona aleatoriamente um dos MH que estão aptos a se movimentar, de acordo com as restrições temporais inerentes às RPC e aos critérios já citados anteriormente que levaram o MH a necessidade de se mover na mesma rua e faixa. A transição *MAKE PLATOON* monta um pelotão (*PLATOON*) com os MHs que estão na mesma rua e faixa do MH selecionado. O pelotão só é montado completamente uma única vez durante a simulação. Após sua montagem inicial, a cada passo dado por um MH, a posição do mesmo é atualizada dentro do pelotão.

Se o MH estiver à frente do pelotão, ele estará livre para se mover e alcançar a velocidade desejada $v_{desired}$. Isso ocorre no disparo da transição *FREE TO GO* em que, de acordo com a velocidade atual do MH e a aceleração a (configurada como constante no modelo), será calculada a distância percorrida, a nova posição e velocidade do MH.

Caso do MH possua um MH a sua frente, sua nova velocidade deve ser calculada de acordo com o algoritmo IDM. Isso ocorre no disparo da transição *STOP AND GO*. Antes do cálculo da nova velocidade é calculada a distância percorrida com a velocidade atual e sua nova localização. Após isso, será calculada a nova velocidade e atualizada a marcação do MH junto com a nova posição.

Após o disparo da transição *STOP AND GO* ou *FREE TO GO*, o lugar *Mobile Hosts* pertencente às sub-páginas *Traffic Generator* e *TrGeT* será atualizado e a simulação

seguirá até que todos os MHs se movam e alcancem o tempo de parada configurado por meio da constante t , que representa o tempo de simulação desejado em segundos.

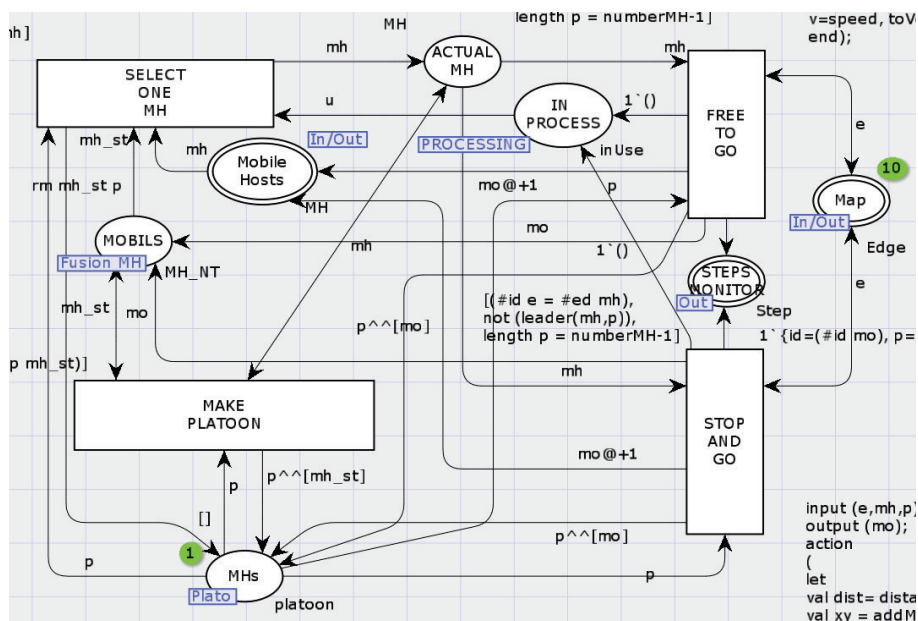


Figura 7. Página com o módulo IDM

Ao final da simulação, toda a movimentação dos MHs se encontra nas fichas do lugar *Steps Monitor* da sub-página *TrafficGenerator*. O lugar recebe fichas da cor *Step* e vinculado a ele existe um monitor *CPN Tools* do tipo *write in file* chamado *Steps*. O monitor gera um arquivo de formato texto na pasta *output* dentro do diretório corrente com a marcação do lugar. Um exemplo da marcação gerada é:

$1\{id = 0, p = \{x = 260.602831871, y = 242.481309487\}, v = 15.8698059436\}@6.$

A marcação traz a velocidade ($v = 15.8698059436$) e a localização ($x = 260.602831871, y = 242.481309487$) de um determinado veículo ($id = 0$) em um dado tempo ($@6$).

7. Validação

A validação da ferramenta *TrGeT* foi feita em duas etapas: i) comparação dos *traces* gerados pelas ferramentas *TrGeT* e *VanetMobiSim*; ii) simulação e comparação da VANET no simulador de rede NS-2 utilizando os *traces* de mobilidade gerados pelas ferramentas *TrGeT* e *VanetMobiSim*. Os arquivos utilitários criados para a análise de desempenho se encontram no repositório GitHub [git].

O cenário de simulação no *VanetMobiSim* é configurado por meio de um arquivo *xml*. O arquivo *scenario_malaga.xml* foi configurado com as mesmas propriedades utilizadas pelo modelo *TrGeT*: tempo de simulação, número de veículos, mapa (Málaga, Espanha) e modelo de mobilidade IDM. No final da execução é gerado o arquivo de *traces* *dumpTraces.tcl*.

Para a simulação do cenário da VANET no NS-2, foi definido um fluxo de dados específico representando as diferentes possibilidades de comunicação existentes entre os

veículos, tomando como base os experimentos realizados em [Toutouh et al. 2012]. O modelo de fluxo de dados executa 10 sessões de um gerador de dados com taxa de bits constante (*constant bitrate* - CBR). Este fluxo opera sobre agentes UDP (*user datagram protocol*) definidos nos nós (veículos). Assim, os veículos interconectados trocam os dados gerados pelos agentes CBR. O tamanho do pacote de dados CBR é 512 *bytes* a taxa de 4 pacotes/s. O restante dos parâmetros de simulação estão resumidos na Tabela 2.

Tabela 2. Parâmetros da simulação NS-2

| Parâmetro | Valor | Parâmetro | Valor |
|-------------------------|-----------------|-------------------------|----------|
| Modelo de propagação | <i>Nakagami</i> | Frequência da portadora | 5.89 GHz |
| Largura de banda | 6 Mbps | Camada PHY/MAC | 802.11p |
| Camada de roteamento | <i>OLSR</i> | Camada de transporte | UDP |
| Taxa de transmissão CBR | 100 kbps | Intervalo CBR | 60 s |

Para avaliar a qualidade do modelo *TrGeT*, utilizamos duas métricas de *QoS*, como proposto em [Toutouh et al. 2012]: (i) taxa de entrega de pacotes (*packet delivery ratio* - PDR), correspondente à fração de pacotes de dados originado por um aplicativo completamente e corretamente entregue; (ii) o atraso de fim a fim (*end-to-end delay* - E2ED), que é a diferença entre o tempo em que um pacote de dados é originado por um aplicativo e o tempo que o mesmo é recebido em seu destino.

7.1. Resultados

Na Figura 8 é mostrado o gráfico da variação das médias das velocidades por quantidade de veículos no cenário de simulação. Foram realizadas 10 rodadas de simulação para cada uma das seguintes quantidades de veículos: 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100. O gráfico mostra como a média das velocidades se comporta à medida que a quantidade de veículos aumenta, simulando o comportamento de um engarrafamento.

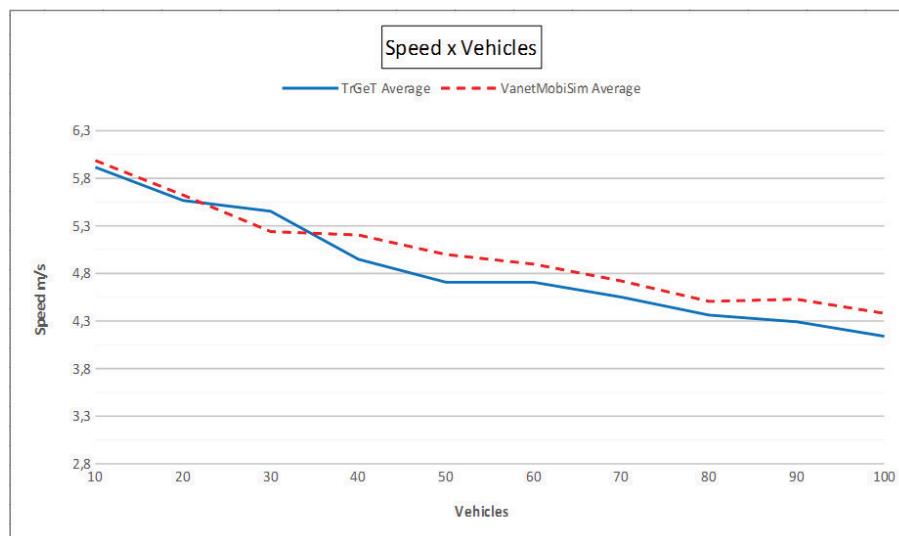


Figura 8. Média das velocidades por quantidade de MHs

Pode-se observar a tendência de diminuição na média das velocidades a medida que a quantidade de veículos é incrementada em ambas as ferramentas. Pelo fato da

aleatoriedade permear toda a simulação (escolha da localização inicial de cada veículo, velocidades desejadas e locais de origem e destino de cada jornada) é esperado que as linhas do gráfico apresentem pontos diferentes para cada ferramenta. Porém, a tendência da diminuição das médias das velocidades também é esperada e é confirmada por meio do gráfico, mostrando o mesmo padrão de comportamento das velocidades nos *traces* gerados por ambas as ferramentas.

Na simulação da VANET no NS-2 os seguintes resultados foram obtidos: a simulação realizada por meio dos *traces* de mobilidade gerados pela ferramenta *TrGeT* apresentou a mediana da PDR de 99,52% e a mediana do E2ED de 3,76 *ms*; a simulação realizada com os *traces* de mobilidade gerados por meio da ferramenta *VanetMobiSim* apresentou a mediana da PDR de 99,99% e a mediana do E2ED de 1,97 *ms*. Os valores das métricas apresentam variações mínimas e de acordo com os encontrados em [Toutouh et al. 2012], validando a ferramenta *TrGeT*. A variação das medianas de PDR e E2ED entre as ferramentas é de 0,47% e 1,79 *ms*, respectivamente.

8. Conclusão

Neste trabalho, foi apresentada a ferramenta *TrGeT*, que tem por objetivo gerar, de forma rápida e com configuração de cenário simplificada, os *traces* de mobilidade usados para simulação de VANETs. Os *traces* gerados, por sua vez, servem como dados de entrada para simuladores de rede.

Constituindo-se em um simulador modelado em RPC, o que sugere grande potencial de expansão e adaptação quando comparado a implementações feitas exclusivamente em linguagem de programação, a ferramenta *TrGeT*, durante a análise de desempenho, mostrou resultados comparáveis ao *VanetMobiSim*. Isto foi verificado por meio do gráfico da variação das médias das velocidades dos veículos por quantidade de veículos, bem como pelo uso do modelo de mobilidade IDM gerado como entrada no simulador de rede NS-2. Como diferencial relativamente a outras ferramentas, pelo fato de ser modelado em RPC, o *TrGeT* permite a depuração da geração de *traces* e a interferência no estado do sistema ao longo da simulação, o que pode ser útil para estudos e análises específicas do modelo de mobilidade utilizado.

Como trabalho futuro pretendemos realizar um estudo minucioso sobre os algoritmos de mobilidade microscópicos CFM para posterior adição à ferramenta *TrGeT*.

References

- [cpn] CPN Tools. <http://cpntools.org/>. [Online; acessado em 22/02/2017].
- [git] GitHub Repository - TrGeT. <https://github.com/lcvasconcelos/trget>. [Online; acessado em 22/02/2017].
- [ope] OpenStreetMap. <https://www.openstreetmap.org>. [Online; acessado em 22/02/2017].
- [tig] TIGER - Topologically Integrated Geographic Encoding and Referencing. https://tigerweb.geo.census.gov/tigerwebmain/TIGERweb_main.html. [Online; acessado em 22/02/2017].
- [Akyildiz et al. 2005] Akyildiz, I. F., Wang, X., and Wang, W. (2005). Wireless mesh networks: a survey. *Computer networks*, 47(4):445–487.

- [Al-Sultan et al. 2013] Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., and Zedan, H. (2013). A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, 37:380–392.
- [Behrisch et al. 2011] Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo—simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.
- [Camp et al. 2002] Camp, T., Boleng, J., and Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502.
- [D. Cabarkapa 2016] D. Cabarkapa, M. M. (2016). Importance of realistic mobility software models for vanets simulations. In *The 7th International Conference - Information Technologies and Education Development - ITRO 2016*.
- [Harri et al. 2009] Harri, J., Filali, F., and Bonnet, C. (2009). Mobility models for vehicular ad hoc networks: a survey and taxonomy. *IEEE Communications Surveys & Tutorials*, 11(4).
- [Härri et al. 2006] Härri, J., Filali, F., Bonnet, C., and Fiore, M. (2006). Vanetmobisim: generating realistic mobility patterns for vanets. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 96–97. ACM.
- [Jensen and Kristensen 2009] Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri nets: modelling and validation of concurrent systems*. Springer Science & Business Media.
- [Karnadi et al. 2007] Karnadi, F. K., Mo, Z. H., and Lan, K.-c. (2007). Rapid generation of realistic mobility models for vanet. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 2506–2511. IEEE.
- [Kesting et al. 2010] Kesting, A., Treiber, M., and Helbing, D. (2010). Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605.
- [Khan et al. 2014] Khan, N., Ahmad, F., and Baig, W. (2014). Formal model of random walk mobility model for wmn using coloured petri nets. In *Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on*, pages 436–444. IEEE.
- [Khan et al. 2015] Khan, N. A., Ahmad, F., and Khan, S. A. (2015). Sher: a colored petri net based random mobility model for wireless communications. *PloS one*, 10(8):e0133634.
- [Toutouh et al. 2012] Toutouh, J., García-Nieto, J., and Alba, E. (2012). Intelligent olsr routing protocol optimization for vanets. *IEEE transactions on vehicular technology*, 61(4):1884–1894.
- [Treiber et al. 2000] Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805.