Avaliação de Simuladores para Redes Veiculares Ad Hoc para Implementação de Aplicações P2P de Gerenciamento

Lucas De Marchi Dreger[1], **Jéferson Campos Nobre**[1]

[1]Instituto de Informática – Universidade do Vale do Rio dos Sinos – UNISINOS
São Leopoldo – RS – Brazil

lucasdreger@gmail.com, jcnobre@unisinos.br

Abstract. *Vehicular ad hoc networks (VANETs) were designed to bring benefits to the driver and passengers of a vehicle. These networks are characterized by the high degree of dynamism and constant mobility of its nodes. These characteristics imply, especially in its management, and may hinden the application of traditional management solutions. Management systems based on P2P technology (P2P-Based Network Management-P2PBNM) can be an alternative effective management. Uses tests can be performed through network simulators. However, the simulators need to be evaluated. Therefore, features and functionalities of the simulators were pointed. At the end, the results obtained from the simulation within a P2P management application were satisfied.*

Resumo. *Redes veiculares ad hoc (Vehicular Ad Hoc Networks - VANET) foram projetadas para trazer benefícios ao motorista e aos passageiros de um veículo. Estas redes são caracterizadas pelo alto grau de dinamismo e constante mobilidade de seus nós. Tais características implicam, principalmente, no seu gerenciamento e podem dificultar a aplicação de soluções tradicionais. Sistemas de gerenciamento baseados em tecnologia P2P (P2P-Based Network Management - P2PBNM) podem ser uma alternativa eficaz. Testes de aplicações podem ser realizados através de simuladores de rede. No entanto, convém que os simuladores sejam avaliados. Sendo assim, foram elencadas características e funcionalidades dos simuladores trabalhados. Os resultados obtidos das simulações com o uso de uma aplicação P2P de gerenciamento foram satisfatórios.*

1. Introdução

Atualmente, a maior parte das pesquisas sobre Redes Veiculares *Ad Hoc* (*Vehicular Ad Hoc Networks* - VANET) é realizada com o auxílio de simuladores para este fim. Isto ocorre, entre outros motivos, porque o custo de implementação de VNs reais é considerado elevado [Alves et al. 2009]. A realização de simulações também traz vantagens, se comparada à implementação real. Além da redução do custo, podem ser citadas a comodidade na implementação e a redução do tempo investido.

VNs possuem a necessidade de um gerenciamento eficiente, tal como as redes tradicionais. No entanto, as características dessas redes (*e.g.,* alta mobilidade dos nós e topologia dinâmica) oferecem desafios para o gerenciamento efetivo das mesmas. A utilização de tecnologia Par-a-Par (*Peer-to-Peer* – P2P) no gerenciamento de redes pode ser uma alternativa viável para VNs. [Nobre et al. 2013]. Esta alternativa, no entanto, precisa ser avaliada.

Este trabalho está organizado da seguinte maneira. Na seção 2 é apresentada uma revisão teórica sobre VNs e seus diferentes tipos de simuladores. A seção 3 caracteriza as diferentes abordagens de gerenciamento de VANETs. Nas seções 4 e 5 será descrito como se deu a escolha dos simuladores, bem como as atividades práticas realizadas. Por fim, a avaliação dos resultados é apresentada na seção 6.

2. Fundamentação Teórica

Rede veicular (*Vehicular Network – VN*) é uma tecnologia emergente que visa integrar as capacidades de redes sem fio de nova geração com veículos inteligentes [Wang 2012]. Seu objetivo, portanto, é facilitar a troca de informação entre veículos ou entre veículo e infraestrutura (ponto de acesso).

VNs podem ser divididas em três tipos de arquitetura, de acordo com sua utilização: *ad hoc*, com infraestrutura e híbrida [Alves et al. 2009]. Em redes *ad hoc*, cada nó da rede é caracterizado por um veículo. Este utilizará mensagens *multicast* e *broadcast* para transmitir informações para veículos próximos [Zeadally et al. 2012]. As redes com infraestutura contam com um terminal centralizador de mensagens, responsável pelo roteamento de informações. Por fim, redes híbridas são um meio termo entre as duas já citadas, possuindo alguns pontos com infraestrutura e alguns pontos com roteamento *ad hoc*.

A realização de testes de desempenho em VNs pode se tornar inviável por exigir um grande número de pessoas, condições climáticas favoráveis e possuir custos elevados. Como consequência, em muitos casos são utilizados simuladores de redes para a realização de testes e avaliações. Embora a utilização de simuladores traga mais facilidade e utilize menos recursos, seus resultados servem apenas como indicativos de solução, podendo divergir dos resultados reais de uma rede veicular [Alves et al. 2009].

3. Gerenciamento de VANETs

O gerenciamento de uma rede veicular passa pela sua arquitetura de comunicação. Em 2004, iniciou-se a padronização da comunicação em VNs pelo IEEE. O padrão criado ainda está em fase de desenvolvimento, e é conhecido como IEEE 802.11p WAVE. A arquitetura WAVE trabalha com uma nova pilha de protocolos de comunicação, baseada no protocolo *Wave Short Message Protocol* (WSMP). Mensagens WSMP são normalmente mais utilizadas em VANETs. Isto ocorre pelo fato de terem sido criadas especificamente para este tipo de rede, podendo trazer benefícios para esta comunicação. A arquitetura WAVE, no entanto, suporta também o envio de datagramas IP [Alves et al. 2009].

VANETs são caracterizadas pela grande mobilidade e o alto dinamismo dos nós. Dado o fato de que os veículos trafegam em alta velocidade em rodovias, é importante que eles possam se comunicar em tempo real. Caso ocorra uma situação de emergência, todos os veículos afetados poderão ser informados sobre o evento. Outro desafio encontrado em VANETs é o alto número de nós. Em áreas metropolitanas, o número de veículos que trafega em um dado horário pode chegar a ordem de milhões, o que pode significar um congestionamento na rede. O fato de grande parte das mensagens entre veículos ocorrer em forma de *broadcast* contribui para a ocorrência deste problema [Zhang 2012]. A importância destas redes, somada as suas características e particularidades, gera um aumento na busca de formas de um gerenciamento efetivo.

3.1. Gerenciamento de VANETs através de sistemas de gerenciamento P2P

O gerenciamento de rede baseado em P2P (*P2P-Based Network Management* – P2PBNM) surgiu a fim de integrar modelos de gerenciamento de redes tradicionais com os novos serviços introduzidos pelas redes P2P. Sistemas P2PBNM possuem um alto grau de descentralização em tarefas de gerenciamento. Desta forma, os próprios pares de gerenciamento são responsáveis por proverem os recursos necessários para tais tarefas. Também devido a esta descentralização, o uso de informações locais para a tomada de decisões aumenta, o que promove a autonomia dos pares de gerenciamento [Nobre et al. 2013]. Outra característica presente nos pares de gerenciamento é a de possuírem um papel duplo, operando não somente como pares de gerenciamento, mas também participando da comunicação entre os demais *peers* [Granville et al. 2005].

As características de gerenciamento encontradas em redes P2P podem ser muito úteis se aplicadas em redes onde a mobilidade dos nós é alta e constante, como é o caso de VANETs. A utilização de sistemas P2PBNM pode ser uma alternativa viável para seu gerenciamento efetivo. Este tipo de gerenciamento pode ser testado utilizando-se simuladores VANETs juntamente com aplicações P2P, a fim de se obter um ambiente funcional e de baixo custo.

4. Escolha dos simuladores VANET

O presente trabalho contou com a utilização de dois simuladores: O NS-3 e o CORE. A seguir serão citadas as principais características dos simuladores trabalhados.

4.1. NS-3

O Network Simulator 3 (NS-3[1]) pode ser instalado em sistemas UNIX em geral e foi escrito em linguagem C++ e Python. O simulador não possui suporte nativo a sistemas Windows, porém pode ser instalado através do *cygwin*[2]. O NS-3 não possui interface gráfica para criação de simulações. No entanto, após a criação de *scripts*, pode-se fazer uso do *NetAnim*[3] para melhor visualizar as simulações criadas.

Alguns simuladores habilitam os nós da simulação a executar diferentes tipos de sistemas. Isto é feito para o caso de testes de aplicações entre os nós, e normalmente é realizado através de emulações de outros sistemas Linux. No NS-3, podem ser utilizadas algumas abordagens para este fim. Dois exemplos são o *Linux Container* (LXC) e *Direct Code Execution* (DCE).

LXC é utilizado para a criação de túneis de comunicação entre dois ou mais *hosts*. Portanto, o LXC pode ser útil se utilizado em simulações do NS-3, de forma que cada nó simulado represente um *container* diferente na ferramenta. O DCE é um módulo para o NS-3 que provê a implementação de protocolos de rede e aplicações, sem que seja necessário alterar seu código fonte [Camara et al. 2014].

4.2. CORE

O *Common Open Research Emulator* (CORE[4]) [Ahrenholz 2010] é também conhecido por ser um emulador. Ou seja, ele é capaz de realizar uma representação de redes de

[1]https://www.nsnam.org/
[2]http://www.cygwin.com/
[3]http://www.nsnam.org/wiki/NetAnim
[4]http://www.nrl.navy.mil/itd/ncs/products/core

computadores como se cada computador fosse uma instância real. A virtualização do *kernel*, ou implementação LXC, possibilita que cada nó da rede seja uma instância virtual Linux, de forma que cada instância possua uma pilha de protocolos de rede independente (*i.e.,* diferente da máquina original).

O CORE não possui suporte a simulações específicas sobre VANET. Desta forma, foram utilizadas redes sem fio, adicionando-se padrões de mobilidade entre os nós. Para a simulação de redes sem fio é recomendado o uso do *Extendable Mobile Ad-hoc Network Emulator* (EMANE). O simulador também não possui modelos nativos de mobilidade dos nós, porém é possível importar padrões externos. Em função disso, utilizou-se a ferramenta BonnMotion[5], capaz de implementar diversos padrões de mobilidade.

É possível fazer com que as topologias simuladas interajam com um ambiente real. Para tanto, é necessária a instalação de uma *Application Programming Interface* (API), que pode ser obtida da página principal do CORE. No caso de VANETs, isto pode ser útil à medida em que experimentos reais começam a ser criados, de forma a utilizar ambientes reais e virtuais para a criação de um experimento com grandes topologias. Esta função pode ainda ser utilizada a fim de se obterem recursos computacionais adicionais (*e.g., clusters*).

5. Simulação realizada

Foram realizadas simulações em ambientes físicos e virtuais. Em ambos os casos, utilizou-se o sistema Ubuntu 12.04 de 64 bits.

A implementação original do simulador NS-3 possui exemplos de suporte a *Linux Containers* (LXC). Portanto, torna-se uma alternativa o uso de códigos funcionais de outros *scripts*, modificando-os para que se enquadrem na simulação em questão. Como resultado, obteve-se um *script* de dois nós em uma rede sem fio com um padrão de mobilidade nativo do NS-3. A medida em que a simulação cresce, torna-se necessária a criação de mais nós com suporte a LXC no *script* de simulação.

A opção pela utilização de poucos nós na simulação se deu por dois motivos: primeiramente para que a topologia seja melhor compreendida; o segundo motivo está relacionado à melhor utilização dos recursos computacionais envolvidos. Cada nova implementação LXC significa uma nova camada de virtualização no sistema Linux. Desta forma, é necessário cuidado com o número de implementações simultâneas. Para que a simulação aconteça, o ambiente Linux deverá poder suportar duas novas virtualizações.

A criação de uma topologia no CORE é feita rapidamente através de uma interface gráfica. Foram criadas simulações com dois e três nós. Dessa forma, gerou-se dois arquivos de mobilidade randômica no BonnMotion. Os roteadores foram configurados como unidade sem fio, a fim de melhor representarem veículos em movimento.

Por fim, ambos os simuladores foram configurados com padrões de mobilidade e funcionalidade LXC em seus nós. Neste momento, o ManP2P-ng [6][Duarte et al. 2011], um sistema de monitoramento P2P, foi executado nos nós da rede, criando um *overlay* P2P. Uma vez que o módulo de monitoramento foi iniciado, observou-se que os nós criados iniciaram a troca de mensagens entre si. Não ocorreram incompatibilidades na

[5]http://www.bonnmotion.net

[6]https://github.com/ComputerNetworks-UFRGS/ManP2P-ng

execução do ManP2P-ng. De forma análoga, é possível que o monitoramento P2P obtenha resultados positivos, se implementado em VANETs reais. A Figura 1 ilustra o início da comunicação entre dois nós, chamados *n1* e *n2*, no *overlay* criado no CORE.

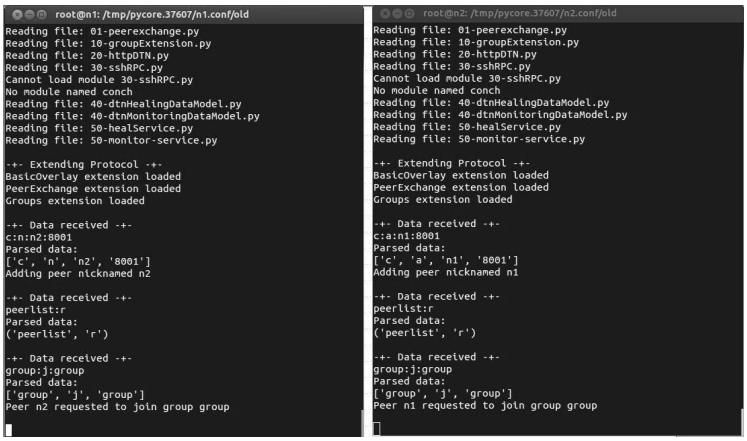

Figura 1. *overlay* criado pelo ManP2P em nós do CORE

6. Avaliação

Inicialmente, foram avaliadas questões relacionadas à instalação das ferramentas. Não existe um arquivo de construção (*e.g., script*) que realize a instalação completa das aplicações. No entanto, em ambos os casos, a documentação de instalação foi clara e precisa.

A próxima característica avaliada foram as implementações LXC apresentadas por ambos os simuladores. Os túneis LXC são uma importante opção para a execução de códigos diretamente nos nós. O CORE apresenta implementações nativas dos túneis, enquanto o NS-3 necessita uma série de configurações adicionais para poder utilizá-los. O NS-3, no entanto, possui outra alternativa para esta função, chamada *Direct Code Execution* (DCE).

Recomenda-se a utilização de sistemas Linux para a instalação dos simuladores. O sistema Mac OSX também suporta a instalação deste simulador. Simulações realizadas em ambientes Windows devem ser feitas através da ferramenta *Cygwin*[7] ou similar. Neste trabalho foi utilizado o sistema operacional Ubuntu Linux 12.04 de 64 bits, sendo que, em nenhum momento foram encontradas incompatibilidades com as aplicações avaliadas.

O estudo anterior mostrou que os simuladores poderiam ser utilizados de forma conjunta com redes reais. Uma vez que foram trabalhados com o uso de máquinas virtuais, verificou-se também se poderiam trabalhar com equipamentos (*i.e., hosts*) reais. Como resultado, constatou-se que ambos os simuladores suportam este tipo de implementação. O CORE possui uma ferramenta própria para conexão com dispositivos externos. O NS-3 pode fazer esta conexão através das implementações LXC.

[7]https://www.cygwin.com

Foram utilizados protocolos de comunicação tradicionais nas duas simulações. Isto ocorreu pois os protocolos de comunicação VANET referenciados anteriormente (*e.g.,* WAVE, WSMP) ainda não foram implementados nos simuladores trabalhados. Sendo assim, utilizaram-se as abstrações já implementadas para o roteamento no NS-3. Para o roteamento entre roteadores no CORE, utilizou-se o protocolo OSPFv3.

A próxima característica avaliada refere-se aos arquivos *trace*. A utilização destes arquivos provê um padrão de mobilidade aos nós, simulando a movimentação de veículos. O NS-3 implementa nativamente uma série de padrões de mobilidade. Desta forma, utilizou-se o padrão *RandomWalk2dMobility*. O CORE, porém, não realiza a implementação de padrões de mobilidade nativos. Ambos os simuladores, no entanto, são capazes de importar arquivos *trace* de ferramentas auxiliares. Arquivos *trace* podem ser obtidos tanto de simuladores de tráfego, quando de ferramentas de geração de modelos de mobilidade. A simulação do CORE foi realizada com padrões de mobilidades gerados pela ferramenta BonnMotion.

O NS-3 possui vantagem sobre o CORE, no que diz respeito a quantidade de nós suportados na simulação. O fato de não possui uma interface gráfica pode ajudar neste sentido, aumentando o desempenho da simulação e fazendo com que mais nós sejam suportados na mesma simulação. Uma pesquisa diz que o NS-3 pode suportar algo em torno de 1000 nós na simulação [Stanica et al. 2011]. O número de nós suportados pelo CORE, de acordo com a sua documentação oficial, pode variar em função de diversos outros fatores (*e.g., hardware* e sistema operacional utilizados, número de processos ativos).

Na Tabela 1, é possível identificar algumas das características avaliadas em cada simulador, bem como os resultados obtidos.

Tabela 1. Comparação de funcionalidades apresentadas pelos simuladores

	NS-3	CORE
Suporte a implementações LXC	x	x
Implementações LXC nativas		x
Suporte a *traces* externos	x	x
Suporte completo a protocolos de comunicação VANET (*e.g.,* WAVE)		
Integração a *hosts* reais	x	x
Integração a redes reais	x	x
Suporte a Sistema Operacional Linux	x	x
Suporte a Sistema Operacional Windows		
Suporte a Sistema Operacional Mac OS X	x	
Suporte a Sistema Operacional FreeBSD	x	x
Interface gráfica		x
Não possui custo para implementação	x	x
Código aberto	x	x
Implementa uma forma de execução de comandos alternativa ao LXC	x	

A aplicação P2P pôde ser implementada em ambos os simuladores sem qualquer tipo de incompatibilidade. Entretanto a utilização do CORE trouxe mais vantagens à simulação. Sua interface gráfica ajudou na solução de problemas e maximizou o tempo de criação de novas simulações. A implementação própria da ferramenta LXC também

foi útil na simulação. O NS-3 não possui esta característica nativamente, portanto foi necessária a criação e configurações de túneis manuais. *Scripts* de configuração LXC foram criados a fim de agilizar este processo, e podem ser visualizados no Anexo 2. O tempo de criação de novas simulações no NS-3 foi mais alto que o do CORE. Isto se deu pois os *scripts* de criação de novas simulações devem ser programados com linguagem C++.

O NS-3, por sua vez, também possui outras vantagens, se comparado ao CORE. Primeiramente, trata-se de uma das maiores ferramentas de simulação da atualidade. O NS-3 possui código aberto e uma documentação completa e organizada. Isto facilita e faz com que seja possível a atualização permanente, com o envio de novos projetos, criados por desenvolvedores. Os projetos podem ser dos mais diversos tipos, podendo variar desde a implementação de um novo protocolo de roteamento, até uma simples correção de falha.

O suporte encontrado pelo NS-3 também é superior ao do CORE. Talvez isto aconteça dada a dificuldade inicial apresentada pelo NS-3 aos novos usuários. Porém, este apresenta uma série de manuais, tutoriais, grupos de discussão e canais de comunicação. O CORE, por sua vez, possui um manual de instruções e um canal de comunicação, via e-mail.

A figura 2 foi criada para melhor compreender as avaliações realizadas. Diferentemente da tabela 1, cujo objetivo é apresentar funções implementadas nos simuladores, a figura 2 elenca qual simulador melhor implementa as características citadas, atribuindo notas de 0 a 5. A avaliação foi realizada com base na construções de simulações e implementação do ManP2P-ng na simulação.

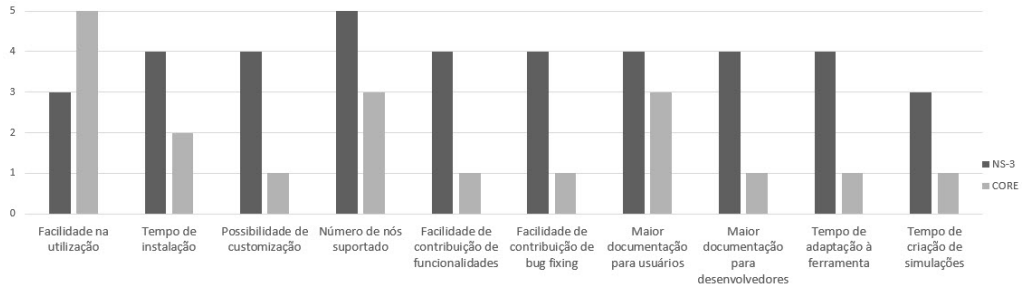

Figura 2. Avaliação de funcionalidades apresentadas por cada simulador

7. Conclusões

O presente trabalho abordou a realização de um estudo com base em simulações e avaliações dos resultados encontrados em simuladores de redes veiculares *ad hoc*. Este trabalho avaliou os simuladores quanto as suas características, considerando os benefícios que as aplicações P2PBNM podem trazer ao gerenciamento de VANETs. Pode-se dizer que, em ambos os casos, os resultados obtidos dos simuladores foram satisfatórios. A aplicação P2P não apresentou incompatibilidades em nenhum dos ambientes trabalhados. Da mesma forma, não houve extrapolação de dados com nenhum dos simuladores. De

forma análoga, portanto, é possível que o monitoramento P2P obtenha resultados positivos, se implementado em VANETs reais. Por outro lado, existem algumas características específicas de VANETs que ainda não estão disponíveis para implementação nos simuladores trabalhados. À medida que protocolos de roteamento utilizados em VANETs começarem a ser modelados também em simuladores, isto deverá trazer benefícios para as simulações.

Embora os resultados obtidos dos simuladores fossem positivos, constatou-se algumas diferenças na sua utilização. Algumas funções podem ser melhor realizadas pelo NS-3 ou pelo CORE. Para melhor visualizar dos resultados obtidos,criou-se uma tabela e um gráfico de comparação na seção 6.

Referências

Ahrenholz, J. (2010). Comparison of core network emulation platforms. In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pages 166–171.

Alves, R., Campbell, I., Couto, R., Campista, M., Moraes, I., Rubinstein, M., Costa, L., Duarte, O., and Abdalla, M. (2009). Redes veiculares: princípios, aplicações e desafios. In *Minicurso do Simpósio Brasileiro de Redes de Computadores*, pages 199–254. SBRC.

Camara, D., Tazaki, H., Mancini, E., Turletti, T., Dabbous, W., and Lacage, M. (2014). Dce: Test the real code of your protocols and applications over simulated networks. *Communications Magazine, IEEE*, 52(3):104–110.

Duarte, P., Nobre, J., Granville, L., and Rockenbach Tarouco, L. (2011). A p2p-based self-healing service for network maintenance. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 313–320.

Granville, L., da Rosa, D., Panisson, A., Melchiors, C., Almeida, M., and Rockenbach Tarouco, L. (2005). Managing computer networks using peer-to-peer technologies. *Communications Magazine, IEEE*, 43(10):62–68.

Nobre, J., Bertinatto, F., Duarte, P., Granville, L., and Tarouco, L. (2013). Gerenciamento oportunístico em redes tolerantes a atrasos/desconexões através da utilização de tecnologia par-a-par na previsão de encontros entre nós. In *XVIII Workshop de Gerência e Operação de Redes e Serviços*. SBRC.

Stanica, R., Chaput, E., and Beylot, A.-L. (2011). Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations. volume 55, pages 3179–3188. Elsevier North-Holland, Inc., New York, NY, USA.

Wang, Y. (2012). Review of vehicular networks, from theory to practice. volume 43, pages 25–29. ACM, New York, NY, USA.

Zeadally, S., Hunt, R., Chen, Y.-S., Irwin, A., and Hassan, A. (2012). Vehicular ad hoc networks (vanets): status, results, and challenges. *Telecommunication Systems*, 50(4):217–241.

Zhang, J. (2012). Trust management for vanets: Challenges, desired properties and future directions. *Int. J. Distrib. Syst. Technol.*, 3(1):48–62.