



Marque a opção do tipo de trabalho que está inscrevendo:

Resumo

Relato de Caso

SIMULAÇÃO DE REDES VEICULARES

Análise de métricas de roteamento para redes veiculares com uso de simulador

AUTOR PRINCIPAL: Jair José Ferronato

ORIENTADOR: Marco Antônio Sandini Trentin

UNIVERSIDADE: UPF – Universidade de Passo Fundo

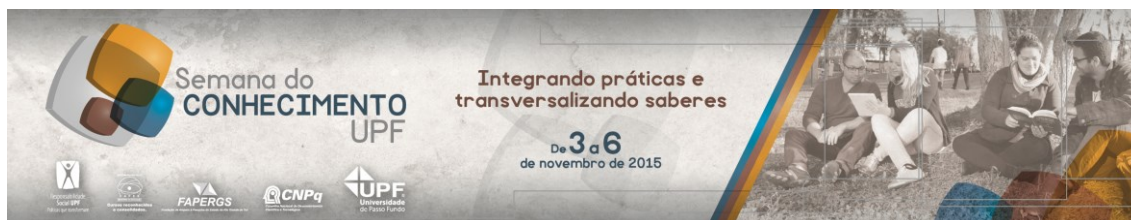
INTRODUÇÃO

As redes veiculares utilizam troca de dados entre os veículos e com a infraestrutura das rodovias para transmissão de dados. Este trabalho apresenta a simulação dos protocolos de redes veiculares através da comparação do comportamento e do desempenho dos protocolos de roteamento para transmissão de dados em ambiente real. Foram utilizados parâmetros como definição de cenário em área central da cidade, velocidade média permitida no local, fluxo de veículos que trafegam em determinados horários, área de abrangência do sinal e protocolos de transporte de dados. Os resultados obtidos demonstram que é possível simular o comportamento das redes veiculares e dessa forma compreender os pontos fracos e fortes de cada protocolo de roteamento simulado. Sendo assim, os experimentos evidenciam a plataforma de simulação e sua arquitetura direcionada para análise de componentes de rede. Os mecanismos criados para automatizar as simulações como scripts em Linux facilitaram o processo em face ao complexo sistema simulador.

DESENVOLVIMENTO:

Segundo RASHVAND (2013), as *Vehicular Adhoc Networks (VANETs)* são consideradas redes dinâmicas e que apresentam impactos conforme a velocidade utilizada por cada veículo e a taxa de deslocamento em áreas urbanas. Cada carro é equipado com dispositivos de comunicação os quais interagem entre si e com a infraestrutura de transmissão de dados existente. As alternâncias de velocidades causam instabilidade nas topologias e aumentam a complexidade e demanda para roteamento e transmissão de mensagens.

A presente pesquisa busca utilizar os recursos de simulação computacional em específico da área de redes de computadores aplicados a um ambiente urbano do Município de Passo Fundo. Com a coleta de dados foi possível obter os parâmetros utilizados no simulador como: fluxo de veículos, velocidade de deslocamento, raio de transmissão de



senal, área total da simulação, tempo de parada, tempo de simulação e tamanho do pacote de dados.

O simulador utilizado nesta pesquisa é o Network Simulator (NS2). Segundo ISSARIYAKUL (2008), este simulador é orientado a eventos e tem código aberto projetado especificamente para testes em redes de comunicação. Este simulador contém módulos para vários componentes como roteamento, protocolos da camada de transporte, a nível de enlace, físico e outras aplicações. Para investigar o desempenho de redes, usa-se a linguagem de *script* para configurar os nodos e observar os resultados gerados pelo NS2. A Figura 1 dos anexos representa a arquitetura do NS2. Este simulador consiste no uso de duas linguagens: C++ e a ferramenta orientada a objetos *OTcl*. O resultado consiste num arquivo *Trace* que pode ser visualizado através do recurso gráfico de animação chamado *Network Animator (NAM)* ou do aplicativo gerador de gráficos *XGraph* para apresentar resultados das simulações.

Esta pesquisa considerou parâmetros como modelo de mobilidade *Manhattan Grid*, o qual usa conceito de grades como ruas e probabilidade de conversão dos veículos. Este modelo é essencial para identificar a forma de locomoção dos veículos e o tráfego gerado entre os agentes da simulação foram *TCP/FTP* e *UDP/CBR*. Para analisar os arquivos gerados pelo NS2 foi utilizado a linguagem de script *AWK* presente no sistema operacional Linux e desta forma é possível analisar cada pacote gerado no nodo de origem incluindo toda sua trajetória até o destino passando por componentes de redes como roteadores e veículos. Esse processo foi automatizado com base em NEVES (2010) através de *scripts* a fim de obter agilidade na alteração de parâmetros do simulador. Também foi utilizado o intervalo de confiança de 95% para obter a confiabilidade dos dados gerados pelo simulador.

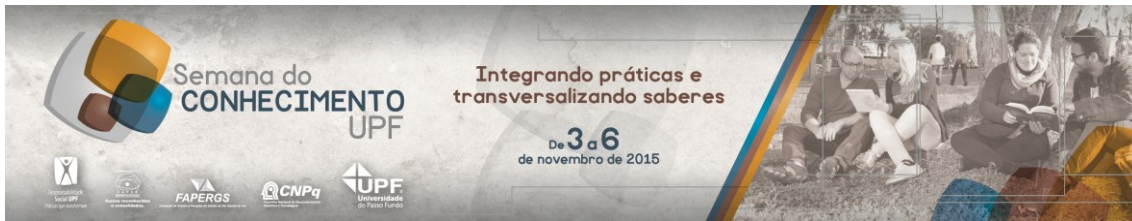
Os resultados evidenciam que é possível com o uso de simuladores analisar protocolos de roteamento de redes veiculares em específico com métricas de desempenho como: latência de rede, taxa de entrega, sobrecarga e transferência. Este cenário possibilita a comparação das técnicas de redes aplicados à um cenário real.

CONSIDERAÇÕES FINAIS:

Esta pesquisa permitiu a simulação de rede através de um ambiente urbano e ferramentas do sistema operacional Linux de forma automatizada para a geração dos arquivos da simulação e análise de dados. A Tabela 1 evidencia a geração da mobilidade de veículos do experimento. Sendo assim, conclui-se que é possível e útil o uso de simuladores de redes para análises e melhorias de protocolos para *VANETs*.

REFERÊNCIAS

RASHVAND, H. F.; CHAO, H. C. Dynamic Ad-Hoc Networks. Institution of Engineering and Technology, 2013. (Telecommunications). Disponível em: <<http://digital-library.theiet.org/content/books/te/pbte059e>> Acesso em: 10 ago. 2015.
ISSARIYAKUL, T.; HOSSAIN, E. Introduction to Network Simulator NS2. 1. ed. Springer Publishing Company, Incorporated, 2008. Disponível em: <<http://www.springer.com/engineering/>



signals/book/978-0-387-71759-3> Acesso em: 25 ago. 2015.

NEVES, Júlio Cezar. Programação Shell Linux. 8.ed. Rio de Janeiro: Brasport, 2010.

ANEXOS

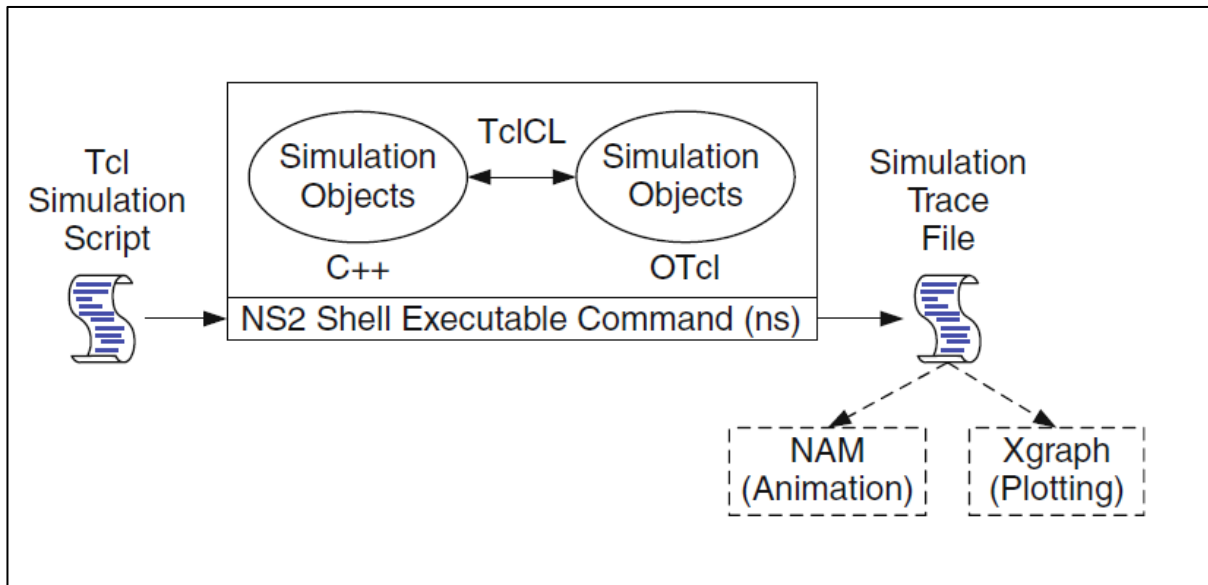


Figura 1 – Arquitetura do simulador NS2 – (ISSARIYAKUL, 2008).

```
#!/bin/bash
i=0
c1=`cat parametros.txt | grep -n ^ | grep 1: | cut -d: -f2`
c2=`cat parametros.txt | grep -n ^ | grep 2: | cut -d: -f2`
t=`cat parametros.txt | grep -n ^ | grep 3: | cut -d: -f2`
a1=`cat parametros.txt | grep -n ^ | grep 5: | cut -d: -f2`
a2=`cat parametros.txt | grep -n ^ | grep 6: | cut -d: -f2`
vel=`cat parametros.txt | grep -n ^ | grep 7: | cut -d: -f2`
while [ $i -le $iter ]
do
#Baixa densidade de nodos
bm -f movimentos/cenario1 ManhattanGrid -d $t -n $c1 -x $a1 -y $a2 -m $vel -u 8 -v 2
bm NSFile -f movimentos/cenario1

#Alta densidade de nodos
bm -f movimentos/cenario2 ManhattanGrid -d $t -n $c1 -x $a1 -y $a2 -m $vel -u 8 -v 2
bm NSFile -f movimentos/cenario2

i=$((i+1))
done
```



Tabela 1 – Código Shell Script para geração da mobilidade. Fonte (autor)