

# GENESI – Gerador Gráfico de Código para o Network Simulator

Marilia Paulo Teles<sup>1</sup>, Mario Antonio Soares Ribeiro Junior<sup>1</sup>, Mauro Margalho Coutinho<sup>2</sup>

<sup>1</sup>CESUPA – Centro Universitário do Pará  
localizado na Av. Gov. José Malcher, 1963 - CEP 66.060-230 - Belém

<sup>2</sup>UNAMA – Universidade da Amazônia

mariliateles@terra.com.br, mario\_jr@amazon.com.br, margalho@unama.br

**Abstract.** *The GENESI project is an application development that uses object oriented concepts and libraries like javax.swing and java.awt.Graphics2D. It's objective is to create a graphical interface to facilitate, both with Network Simulator(NS) the generation of "TCL" codes. In a summary way, will be described GENESI in this article.*

**Resumo.** *O projeto GENESI é o desenvolvimento de um aplicativo que utiliza conceitos de orientação a objeto e as bibliotecas javax.swing e java.awt.Graphics2D. Seu objetivo maior é a criação de uma interface gráfica para facilitar, em conjunto com o Network Simulator (NS), a geração de códigos "TCL". De forma sucinta, descreveremos o GENESI no presente artigo.*

## 1. Introdução.

A área de simulação de redes, quer em instituições de ensino, quer em empresas que disputam o mercado de trabalho, representa um ponto de destaque. Pesquisas são direcionadas para essa área, sempre com o fim de aumentar a produtividade e diminuir a probabilidade de erro no dimensionamento dos recursos de rede.

Ferramenta de simulação muito utilizada no estágio atual da informática, o *Network Simulator (NS)*, um simulador discreto de redes, funciona a partir da interpretação de um script na linguagem TCL/OTCL (*Tool Command Language/Object Tool Command Language*). Não contempla em si mesmo, entretanto, um gerenciador gráfico que auxilie na criação da simulação.

O GENESI, software cujo desenvolvimento se trata neste artigo, tem como proposta central a geração de código fonte do NS (em TCL). Um gerenciador gráfico possibilitará essa geração de código fonte, com o retorno de dados relevantes em forma de gráficos. Isso facilitará a análise de desempenho através de métricas, como vazão, atraso, *jitter* e probabilidade de bloqueio.

Parte de uma proposta mais ampla, tem-se neste trabalho três tipos de rede: wans ou redes de grande arrasto, lans ou redes locais e WAN/LAN. Futuramente pretende-se expandir o projeto GENESI, de formas a possibilitar simulações de outras redes atualmente suportadas pelo NS.

## **2. Avaliação de Desempenho.**

### **2.1 Via experimentação.**

Já existindo o sistema a ser avaliado, deve-se utilizar a técnica de experimentação. Dessa maneira o que for avaliado será comprovado.

Inexistindo o sistema, deve-se criar um protótipo, uma simplificação representativa que mantém as mesmas funcionalidades e descarta os pontos não relevantes do sistema a ser modelado.

Pode-se utilizar essa técnica através de programas conhecidos como *Benchmarks*[1], os quais constituem *softwares* de análise para avaliação de um dado sistema ou hardware e apresentam um demonstrativo de desempenho. O uso de *benchmarks* exige conhecimento das propriedades objeto da simulação.

Melhores resultados podem ser obtidos a partir de uma coleta de dados mais eficiente. Entretanto, fazer essa coleta em tempo real é tarefa complexa, principalmente se for levado em consideração que os dados obtidos não devem ser afetados por outros processos que estejam sendo executados paralelamente.

### **2.2 Via simulação.**

Assim como na avaliação de desempenho via experimentação, a via simulação tende a se aproximar mais da realidade, caso os dados de entrada sejam reais. Mas essa também não é tarefa simples.

A simulação pode ser rica em detalhes, pois não impõe restrições. Todavia, o tempo necessário para realizá-la cresce em proporção direta a sua complexidade. Há ocasiões que se torna imprescindível o consumo bem elevado de tempo de CPU. Por vezes esse consumo chega a ser proibitivo. Nesses casos, geralmente se utiliza um ambiente de computação paralela e distribuída, o que diminui o tempo gasto na simulação [1].

Uma das principais técnicas de avaliação de desempenho via simulação é baseada na utilização de eventos discretos. Sistemas desse tipo mudam de estado em intervalos pré-definidos e não de forma contínua. Os dados que são passados na entrada do programa podem ser obtidos através de lotes conhecidos como *traces*, a partir de um sistema real, ou através de análises probabilísticas. O *Network Simulator* possui módulo denominado de redes vivas, o que permite ao simulador trabalhar com dados de entrada provenientes de um sistema real [5].

## **3. Trabalhos Correlatos.**

O *Trace Graph*[2] é uma ferramenta que pode ser utilizada em conjunto com o NS. Funciona como analisador dos arquivos de saída do NS, a partir dos quais vários tipos de gráficos poderão ser gerados. Essa ferramenta foi produzida por Pedro Vale Estrela, da universidade de *Wroclaw University of Technology*, na Polônia. Serve para reconhecer os arquivos de *trace* de redes cabeadas (wired), redes baseadas em satélites, redes locais sem fio (*wireless*) e redes mistas (*wired-cum-wireless*).

A partir dessa análise, duzentos e trinta e oito tipos de gráficos 2D e doze tipos de gráficos 3D podem ser gerados e salvos como figura nos formatos *joint photographic expert group* (jpg) ou *Tagged Image File Format* (tiff) ou ainda em um arquivo de texto

com os dados disponibilizados em forma de colunas, para que depois possam ser utilizados na geração dos gráficos em outros editores. Nesses gráficos, informações como atraso, *jitter*, que se constitui na variação estatística do atraso, tempo de processamento, número de nós intermediários e vazão são geradas.

O *Trace Graph* se assemelha a um dos módulos propostos no GENESI, onde também serão gerados gráficos a partir da análise do trace produzido pelo NS. Os gráficos que serão inicialmente gerados pelo GENESI são: atraso, *jitter*, vazão e probabilidade de bloqueio. Entretanto, não se pretende fazer vários modelos do mesmo gráfico.

Outra ferramenta que possui proposta semelhante ao GENESI é o TCLGen [9], um projeto da Universidade de Coimbra, desenvolvido por Vítor Fonseca, Jorge Sá Silva e Fernando Boavida. Assim como o GENESI, se dispõe a desenvolver uma interface gráfica em java para a geração de códigos TCL.

Essa ferramenta gerará código para vários tipos de rede suportadas pelo NS e possuirá uma interligação com o mesmo, para que o TCL possa ser interpretado e o *trace* possa ser gerado. Entretanto, diferentemente do GENESI, este projeto não se propõe a avaliar os dados retornados no trace para a obtenção de dados mais relevantes.

#### 4. O projeto VINT.

O *Network Simulator* (NS) é um software de simulação discreta de redes, concebido na universidade de *Berkeley*, Califórnia [3]. Trata-se de um software livre e com o código fonte aberto. Isso tem contribuído para o incremento de credibilidade junto à comunidade científica internacional. A primeira versão do NS foi lançada em 1989. O projeto cresceu e se transformou em um consórcio de diversas instituições. A partir daí passou a se chamar *Virtual Internetwork Testbed* (VINT)[4]. O NS é capaz de simular vários tipos de redes e prover suporte a uma gama de recursos, dentre os quais se destacam:

- a) Protocolos do modelo de referência TCP-IP.
- b) Políticas de fila, como FIFO (*First In First Out*), RED (*Random Early Detection*), RIO (*Random Early detection with In/Out*), CBQ (*Class-Based Queueing*), SFP (*Stochastic Fair Queueing*) etc.
- c) Algoritmos de roteamento como o de *Dijkstra*.
- d) Protocolos da camada MAC como o *multicast* e o *unicast* etc [5].

As simulações são realizadas a partir de um interpretador de scripts TCL/OTCL, desenvolvido no MIT (*Massachusetts Institute of Technology*). Para simular no NS o usuário deve escrever um script em TCL contendo todas as informações pertinentes à rede, como topologia, fontes do tráfego, configurações do enlace pelo qual passarão os pacotes etc. Como resultado, há o retorno de um arquivo texto que deverá ser analisado pelo usuário e a partir da análise deste arquivo (*trace file*), informações mais concretas sobre a simulação são obtidas. Todo esse procedimento requer um nível de conhecimento relativamente alto do simulador, uma vez que os resultados que o NS gera estão encapsulados no arquivo de *trace* e sua

interpretação não é simples. A simulação, por vezes, exige muita programação, o que desestimula o seu uso em atividades de planeamento nos ambientes comerciais.

A proposta do GENESI foi concebida exatamente para minimizar esse problema. A partir de uma interface amigável (*user-friendly*), o usuário passa a se preocupar apenas com objetos gráficos para a montagem do esquema da simulação e com a análise do resultado em forma de gráficos. A tarefa passa a ser mais atraente e, enquanto tal, mais produtiva e mais fácil inclusive para quem não conhece mas tem a necessidade de trabalhar com o NS, sem precisar se tornar um especialista.

## 5. Componentes do Network Simulator (NS).

### 5.1. Nós.

São estruturas, encontradas no NS, compostas por dois classificadores: o classificador de endereços e o classificador de portas. O classificador de endereços é responsável por verificar se o pacote recebido, através do enlace de entrada, tem como destino aquele nó ou se o mesmo deve ser roteado mais adiante. Já o classificador de portas tem como finalidade conduzir o pacote até à aplicação correta. Um nó possui a capacidade de se vincular a diversos agentes ou protocolos (UDP,FTP etc.)[6].

### 5.2. Enlaces.

O enlace possui uma seqüência de eventos internos, iniciando-se na fila. Caso o pacote seja descartado, este é encaminhado para o agente nulo; caso contrário o pacote será atrasado e enviado para o controle de TTL (*Time to Live*).

Um pacote ao entrar no Nó, será encaminhado inicialmente para uma fila. Esta poderá ser parametrizada utilizando várias políticas, como a política *DROP TAIL*, que é política padrão, a *SFQ (Stochastic Fair Queueing)*, a *WFQ (Weight Fair Queueing)*, a *RED (Random Early Discard)* etc. O pacote, ao sair da fila, poderá ser descartado, o que geralmente ocorre em situações de congestionamento, como também poderá ser atrasado, de acordo com a configuração do usuário[6].

## 6. Definindo o Cenário e o Tráfego.

O gráfico da figura 1 representa um exemplo composto por quatro nós e três enlaces.

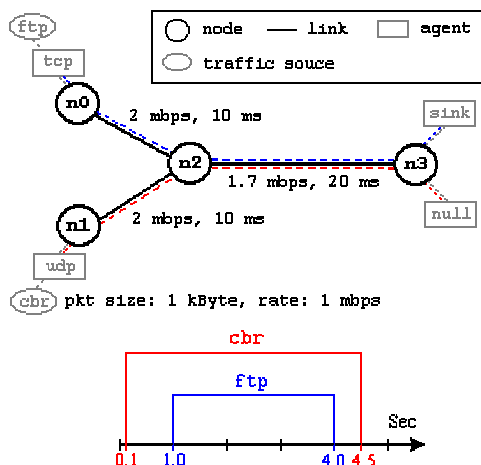


Figura 1 - Modelagem para um Script TCL

Acima de cada nó encontram-se os agentes, que é a forma como são definidos os protocolos de transporte. Na figura 1 pode-se observar um agente TCP (*Transmission Control Protocol*) ligado ao nó n0, um agente UDP (*User Datagram Protocol*) ligado ao nó n1 e dois agentes de recepção ligados ao nó n3 (SINK e NULL). Esses dois agentes são os responsáveis por receber os pacotes encaminhados pelos protocolos de transporte. Os Agentes de recepção **NULL/ SINK**, recebem os pacotes e, caso necessário, geram os pacotes de reconhecimento (ACK).

## 7. O projeto GENESI.

O projeto GENESI consiste no desenvolvimento de um aplicativo orientado a objeto, com interface gráfica. Isso facilita, em conjunto com o *Network Simulator* (NS), a geração de códigos TCL e, conseqüentemente, a validação de projetos de redes.

O objetivo é auxiliar no processo de simulação. A partir do momento em que o usuário passa a utilizar o GENESI, muitos detalhes de código começam a ser abstraídos, uma vez que recursos de construção baseados em arrastar e soltar facilitam a concepção e visualização do modelo, sem a necessidade de preocupação com o código.

É de destacar no GENESI a portabilidade, uma vez que o funcionamento independente do sistema operacional se dá em face da plataforma adotada: JAVA. A primeira etapa do projeto consistiu no desenvolvimento de uma interface, onde foram aplicados os conceitos de *user-friendly* com base na biblioteca *javax.swing* e *java.awt.Graphics2D* [7], [8]. Essa interface está dividida em três partes:

a) A primeira composta por uma área de trabalho onde o usuário construirá sua topologia e inserirá os diversos elementos de simulação, como protocolos, parâmetros do enlace etc. A área de trabalho é subdividida em área de desenvolvimento gráfico, onde serão adicionados todos os componentes da rede projetada pelo usuário, e área de apresentação do código gerado pelo GENESI a partir da rede desenhada na área de desenvolvimento gráfico, como pode ser observado nas figuras 2 e 3.

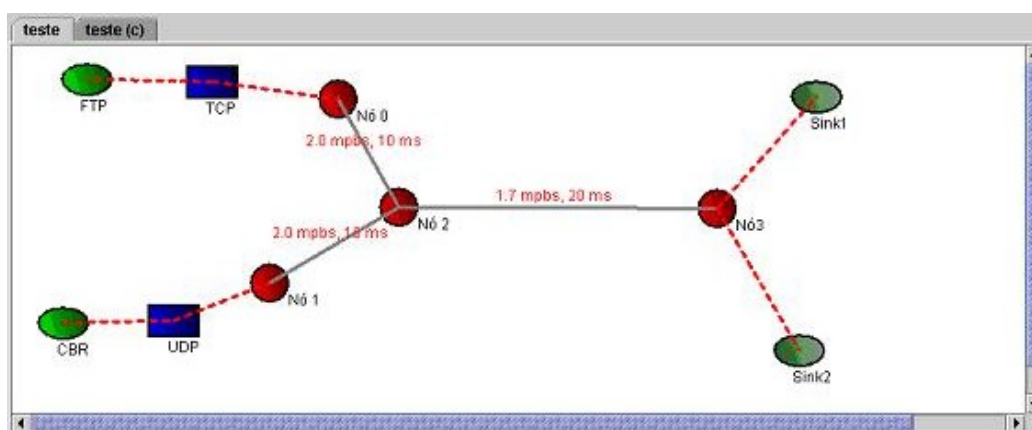


Figura 2 - Área de desenvolvimento gráfico

```
teste teste (c)
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

$ns color 1 orange
$ns color 2 blue
```

**Figura 3 - Área de geração de Código**

b) A segunda parte é composta por uma área de definição de elementos de rede do sistema, na qual o usuário escolherá qual dos objetos deseja adicionar na área de desenvolvimento gráfico. Essa barra contém todos os botões essenciais ao projeto da rede. A barra de ferramentas pode ser visualizada na figura 4.



**Figura 4 - Barra de ferramentas (da esquerda pra direita: nó, enlace, ligação, transporte, aplicação, sink, excluir objeto e excluir tudo).**

c) A terceira parte é composta por uma barra de propriedades que contém a configuração de dados e a régua de evolução da simulação. A barra de dados é o local no qual o usuário poderá estabelecer parâmetros dos valores da simulação. Trata-se de um objeto dinâmico que altera seu conteúdo de acordo com o item selecionado. A barra de configuração de dados possui cinco variações distintas para cada objeto selecionado: a do objeto nó (figura 5); a do objeto enlace (figura 6); a do objeto transporte (figura 7); a do objeto aplicação (figura 8); e a do objeto sink (figura 9).

Nome:

Tipo:

**Figura 5 - Barra de configuração de dados do objeto NÓ**

Nome:

Vazão:

Atraso:

Origem:

Destino:

Fila:

Buffer:

Tipo:

Vazão volta:

Atraso volta:

**Figura 6 - Barra de configuração de dados do objeto ENLACE**

Nome:

Tipo:

**Figura 7 - Barra de configuração de dados do objeto TRANSPORTE**

Nome:

Tipo:

Cor:

**Figura 8 - Barra de configuração de dados do objeto APLICAÇÃO**

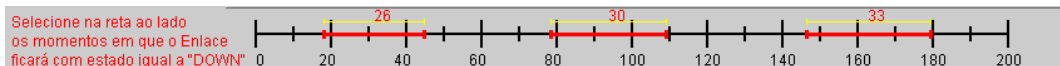
Nome:

Tipo:

Transporte:

**Figura 9 - Barra de configuração de dados do objeto SINK**

A régua (figura 10) mostra, um componente que será ativado quando os objetos do tipo enlace ou do tipo aplicação forem criados ou seleccionados, sendo que esta barra é independente para cada objeto, ou seja, as marcações feitas para um objeto do tipo enlace não irão interferir na instância do objeto do tipo aplicação ou mesmo para outro objeto do tipo enlace.



**Figura 10 - Régua**

Após o projeto da rede, o usuário, através do menu “Executar” (figura 11), poderá realizar a simulação da rede desenvolvida. Porém, antes de executar a simulação em conjunto com o NS versão 2.24, o GENESI fará uma validação do modelo de rede criado pelo usuário, onde possíveis erros serão verificados. Caso seja detectada alguma inconsistência no modelo, o GENESI não iniciará a simulação e exibirá (figura 12), os erros até então encontrados durante a validação da rede simulada.

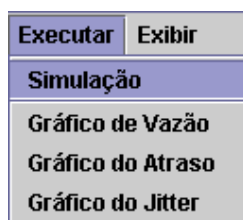


Figura 11 - Menu Executar

 A screenshot of a window titled "OUTPUT (4)". The window header says "Foram Encontrados: 4 erros". Below this is a table with three columns: "Erro", "Descrição", and "Objeto".
 

Erro	Descrição	Objeto
✘ OBJETO SEM ENLACE/LIGAÇÃO	O OBJETO COM O NOME: "Nó1" DO TIPO "NÓ" NÃO PODE FICAR SEM NENHUM ENLACE/LIGAÇÃO	Nó1
✘ APLICAÇÃO INEXISTENTE	É NECESSÁRIO QUE EXISTA PELO MENOS UMA APLICAÇÃO PARA QUALQUER TIPO DE SIMULAÇÃO	SEM APLICAÇÃO
✘ TRANSPORTE INEXISTENTE	É NECESSÁRIO QUE EXISTA PELO MENOS UM TRANSPORTE PARA QUALQUER TIPO DE SIMULAÇÃO	SEM TRANSPORTE

Figura 12 - Barra de OutPut

Caso o GENESI não encontre nenhum erro na rede simulada, uma mensagem informando que a validação foi concluída com sucesso será exibida.

Para que o processo seja encerrado com sucesso, o NS versão 2.24, deverá estar devidamente instalado e configurado, pois realizará a simulação da rede desenvolvida, retornando os *traces* que servirão de entrada para os gráficos com resposta da simulação realizada.

O tratamento dos arquivos de *trace*, permitirá que gráficos de atraso, *jitter*, vazão e probabilidade de bloqueio sejam apresentados. Assim o usuário poderá visualizar de maneira prática e rápida o desempenho da rede desenvolvida. Os gráficos também poderão ser salvos em disco para que depois possam ser reutilizados e melhor analisados.

## 8. Conclusões e Trabalhos Futuros.

O Projeto GENESI, neste primeiro momento se destinou ao desenvolvimento de aplicações para redes *WAN/LAN*. Entretanto as pesquisas e os estudos continuarão para que mais adiante seja implementado em todos os tipos de redes que o NS simula.

Outro objetivo do projeto GENESI, será utilizar a função do Java chamada *applet*, a qual permitirá que o aplicativo seja executado via *browser*, em qualquer dos sistemas operacionais – *Linux* ou *Windows* - sem pormenores de configurações, como identificar se o *Network Simulator* está instalado corretamente e onde o mesmo está instalado.

Após a conclusão do projeto GENESI, um dos objetivos será a disponibilização de uma versão beta para uso geral, principalmente em universidades, para que possam ser identificadas possíveis falhas que serão notificadas para a devida correção em versões posteriores.

Isso possibilitará que o GENESI se torne conhecido e utilizado livremente por todos que precisam simular uma rede, mas que ainda esbarram na dificuldade de estudar uma linguagem de programação.

## 9. Referências Bibliográficas.

- [1] LAW, A.M.; KELTON, W.D. Simulation Modeling and Analysis. Third Edition, McGraw-Hill, 2000.
- [2] Site Oficial do Trace Graph. Disponível em: <http://www.geocities.com/tracegraph>. [Capturado em Junho de 2005].
- [3] Site do Projeto NS. Disponível em: <http://www.isi.edu/nsnam/vint/>. [Capturado em Junho de 2005].
- [4] Site do Projeto VINT. Disponível em: <http://netweb.usc.edu/vint/vint-summary.html>. [Capturado em Junho de 2005].
- [5] Documentação On-line do Projeto NS. Disponível em: <http://www.isi.edu/nsnam/ns/doc/index.html>. [Capturado em Junho de 2005].
- [6] Margalho, Mauro Coutinho. Costa, João Crisóstomo Weil. Francês, Carlos Renato Lisboa. Avaliando o Desempenho de Redes Através do Simulador Network Simulator. SEPAI, 2002, Belém.
- [7] Documentação da Linguagem JAVA. Disponível em: <http://java.sun.com/j2se/1.4.2/docs/API>. [Capturado em Junho de 2005].
- [8] Knudsen, Jonathan. JAVA 2D Graphics. 1º Edição. United States of America. Ed. O'Reilly & Associates, Inc. 2001. 287 p.
- [9] Artigo sobre o TCLGen. Disponível em: <http://www.fcn.pt/crc2003/v3/sessao4/poster09.pdf> [Capturado em Agosto de 2005].