

# QNet – Um Simulador Gráfico de Tráfego IP para Redes Convergentes

Joseane Farias Fidalgo, Carlos Alberto Kamienski, Thiago Souto Maior Cordeiro de Farias, Kelvin Lopes Dias, Fábio Guerra de Almeida, Djamel Fawzi Hadj Sadok, Judith Kelner  
Centro de Informática – UFPE

{jbf,cak,tsmcf,kld,fga,jamel,jk}@cin.ufpe.br

**Abstract.** *Simulation is an essential technique for performance evaluation of computer networks, both for research and business environments. Particularly, converged networks require careful resource configuration and provisioning for QoS and Traffic Engineering technologies. Among the most common existing simulators, none offers the features of friendly user interface, low cost e suitable functionalities. This paper presents QNet, an IP network simulator that has been developed for helping research, design and management of converged networks.*

**Resumo.** *Simulação é uma técnica fundamental para a avaliação de desempenho de redes de computadores, tanto na área de pesquisa, quanto no ambiente corporativo. Particularmente, redes convergentes necessitam da configuração e provisionamento de recursos utilizando tecnologias de QoS e Engenharia de Tráfego. Entre os simuladores mais utilizados atualmente, nenhum combina as características de interface amigável ao usuário, custo baixo e funcionalidades adequadas. Este artigo apresenta o QNet, um simulador de redes IP que está sendo desenvolvido com o objetivo de auxiliar trabalhos de pesquisa, projeto e gerenciamento de redes convergentes.*

## 1. Introdução

O simulador QNet é destinado à simulação de tráfego em redes IP, em um novo contexto de convergência do modelo de negócios em telecomunicações e da premente necessidade de integração de tecnologias para usar eficientemente a infra-estrutura existente. O QNet disponibiliza tecnologias de rede capazes de tratar de tráfego heterogêneo, utilizando mecanismos para provisionar corretamente os recursos com o objetivo de oferecer garantias de desempenho específicas para cada tipo de tráfego. A contribuição do QNet está em integrar numa única plataforma de simulação recursos para tratar de redes convergentes, tecnologias de QoS e engenharia de tráfego para o protocolo IP.

Existem diversos simuladores de tráfego IP disponíveis, os que mais se destacam comercial e academicamente são o OPNET [12] e o *Network Simulator (ns)* [11]. O OPNET é um simulador comercial largamente utilizado no âmbito corporativo, devido às suas funcionalidades e precisão nos resultados. Ele é mais utilizado em grandes empresas e operadoras de telecomunicações, mas restrito em outros ambientes, principalmente devido ao seu custo elevado.

O *ns* é utilizado principalmente por pesquisadores, por ter distribuição gratuita e código aberto. Tal fato o torna adequado a situações onde é necessário desenvolver novas funcionalidades, como em teses e projetos de pesquisa aplicada. No entanto, a sua interface

(textual) não é amigável ao usuário. A execução de um experimento de simulação no *ns* requer a elaboração de *scripts* em Tcl e grande trabalho adicional para obter e visualizar os resultados. Também é comum o usuário necessitar programar em C++ para que as funcionalidades desejadas estejam disponíveis. Além disso, os protocolos e tecnologias no *ns* em geral são desenvolvidos para uso isolado, para resolução de problemas específicos. Integrá-los normalmente é uma tarefa árdua.

O QNet foi desenvolvido a fim de solucionar estes problemas. Ele fornece uma interface gráfica ao usuário, como o OPNET, para configurar os cenários e visualizar os resultados. Além disso, beneficia-se do potencial e flexibilidade proporcionados pelo *ns*. Na seqüência deste artigo, a seção 2 apresenta a estrutura interna do Qnet e a seção 3 mostra as principais funcionalidades atualmente disponíveis no QNet. Por fim, a seção 4 apresenta alguns comentários finais e trabalhos futuros.

## 2. Estrutura Interna do Qnet

O QNet é composto por quatro módulos principais (Figura 1), o gerenciador de simulação, o gerenciador de projetos, o gerador de *scripts* e o simulador de retaguarda *ns*. O gerenciador de simulação gerencia os módulos restantes, representa a principal interface do usuário com o sistema e possibilita a edição de topologias, gerenciamento de tráfego e de tecnologias de QoS, execução da simulação e visualização de resultados.

O gerenciador de projetos é ativado a partir do menu principal do sistema e permite criar, salvar e recuperar projetos, além de gerar e interpretar o arquivo de descrição do QNet. O arquivo de descrição é um arquivo texto com formato interno e bem definido que armazena as informações de configuração de cada projeto criado no QNet.

O gerador de script processa o arquivo de descrição e gera o script de simulação a ser processado pelo *ns*. Como o ambiente nativo de desenvolvimento do *ns* é baseado em sistemas operacionais compatíveis com o Unix, o QNet foi desenvolvido inicialmente para Linux. Sempre que possível, o QNet utiliza as funcionalidades oferecidas diretamente pelo *ns* ou por implementações independentes disponibilizadas pelos seus desenvolvedores. Algumas funcionalidades, no entanto, foram desenvolvidas especialmente para o QNet.

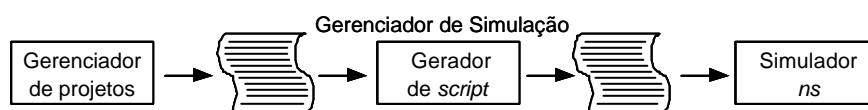


Figura 1. Componentes do QNet

## 3. Funcionalidades do QNet

As funcionalidades atualmente oferecidas pelo simulador Qnet são divididas em topologia, geração de tráfego, tecnologias de Qualidade de Serviço e resultados. Ressalta-se que muitos detalhes são omitidos, devido à limitação de espaço.

### 3.1. Topologia

A topologia de um projeto no Qnet pode ser composta por elementos de backbone e redes de acesso. Uma rede de backbone consiste de um conjunto de roteadores interligados por

enlaces de comunicação e pode ser configurada de duas maneiras distintas, dadas por passo a passo ou nuvem de rede. No modelo passo a passo, o usuário inclui manualmente os roteadores, sistemas finais e os enlaces um a um. Os roteadores possuem um tipo, o qual indica o processamento que o nó é capaz de realizar (melhor esforço, *DiffServ* e *MPLS*). O modelo de nuvem de rede é baseado na agregação de roteadores e enlaces que os interconectam, em uma rede de backbone. Para configurá-la o usuário informa o número de roteadores (nós) e habilita os enlaces entre eles.

Em relação às propriedades configuráveis dos enlaces, tem-se a capacidade ou largura de banda, atraso, tipo e tamanho da fila. Os tipos de filas suportados pelo Qnet variam de acordo com a tecnologia de QoS utilizada – *RED* [6], *WRED* [3], *RIO* [4], *DropTail* e *SFQ* [10] para *DiffServ* e *DropTail* e *SFQ* para melhor esforço.

Os tipos de redes de acesso suportados pelo QNet são *Ethernet*, *xDSL*, *WLAN*, *GSM/GPRS* e *PSTN* (rede de telefonia). A rede de acesso *Ethernet* subdivide-se em compartilhada e comutada. *Ethernet* compartilhada representa uma rede *Ethernet* tradicional com o algoritmo *CSMA/CD*, onde todos os nós estão em um mesmo domínio de colisão interligados por um *hub*. A *Ethernet* comutada representa uma rede onde os nós são interligados por um *switch*, ou seja, não há ocorrência de colisões.

*xDSL* utiliza as tecnologias de linha digital do assinante e subdivide-se em *ADSL*, *HDSL* e *RDSI*. A tecnologia *ADSL* permite a configuração de redes com enlaces assimétricos nos seus enlaces ascendentes (*uplink* – entre 64 e 600 Kbps) e descendentes (*downlink* – entre 64 Kbps e 6 Mbps). A *HDSL* permite a configuração de redes com capacidade simétrica de 2 Mbps. E, por fim, a *RDSI* permite a configuração de redes de acesso digitais com acesso discado com capacidade de 64 ou 128 Kbps.

*WLAN* suporta os padrões *IEEE 802.11* e *IEEE 802.11b* de redes locais sem fio. A capacidade da rede pode ser 1 ou 2 Mbps para *IEEE 802.11* e 1, 2, 5.5 ou 11 Mbps para *IEEE 802.11b*. As redes de acesso *GSM* e *GPRS* representam na realidade dois subtipos de uma mesma rede de acesso. *GSM* é caracterizada por comutação de circuitos e capacidade máxima de 9,6 Kbps. Em uma rede *GPRS*, a capacidade da rede pode atingir até um máximo de cerca de 110 Kbps, dependendo dos parâmetros configurados.

A rede de acesso *PSTN* é diferente das outras redes de acesso, porque é criada aos pares. Cada *PSTN* só pode estar conectada a uma outra *PSTN* e já embute o seu próprio tráfego (voz). A *PSTN* assume a existência de um *gateway* H.323 responsável pela codificação e decodificação do tráfego de voz entre a rede de circuitos e a rede IP (de pacotes). O QNet disponibiliza cinco codificadores de voz padronizados pela ITU-T (G.711, G.723.1, G.726, G.728 e G.729a) e permite a configuração de codificadores com valores escolhidos pelo usuário (codificadores *ad-hoc*). Existem vários cenários para a utilização de voz sobre IP – o caso explorado pelo QNet envolve dois segmentos de *PSTN* interconectados por uma rede *IP*, comum em uma operadora de telecomunicações.

### 3.2. Tráfego

As aplicações geradoras de tráfego são caracterizadas pelos parâmetros de origem, destino, aplicação, tempos inicial e final de simulação e os parâmetros específicos de cada aplicação. Os tipos de tráfego existentes são sintético, agregado e baseado em traces.

O tráfego sintético é gerado através de modelos de tráfego para a simulação de aplicações normalmente utilizadas em redes reais. O QNet dispõe de modelos de fontes individuais e um modelo de tráfego agregado. Os modelos de fontes individuais são *FTP*, *Telnet*, *HTTP*, voz *CBR*, voz *On/Off*, vídeo *CBR* e vídeo *VBR*. Para as aplicações baseadas no protocolo *TCP*, tem-se as opções de *TCP Tahoe*, *Reno*, *NewReno* e *Sack*.

O tráfego agregado ou auto-similar é gerado através da agregação de várias fontes de tráfego *On/Off* com os períodos de *On* e *Off* especificados por uma distribuição de *Pareto*, com o parâmetro *forma* entre 1 e 2. Nessas condições, a distribuição de *Pareto* apresenta cauda pesada, uma condição para a produção do efeito da auto-similaridade [7].

O QNet é capaz de ler arquivos contendo traces e a partir deles gerar tráfego para a simulação. Os tipos de trace são vídeo em formato H.263, obtidos livremente pela Internet (25 trechos de vídeo com 1 hora de duração já estão inclusos no Qnet) e os traces de tráfego agregado e de fluxo individual gerados pelo capturador de tráfego.

O capturador de tráfego para trace agregado permite coletar informações de tráfego IP de saída em uma base de informações de gerenciamento através do protocolo de gerenciamento SNMP. O usuário inicialmente fornece o nome ou o endereço IP do roteador (ou outro elemento de rede). O sistema apresenta as interfaces disponíveis naquele roteador. O usuário escolhe a interface, o intervalo entre as leituras da MIB, o critério de parada (idêntico à captura de fluxos) e a comunidade (*community*) SNMP.

O capturador para fluxos individuais permite capturar tráfego real em uma rede Ethernet, baseado em vários filtros, como endereços IP e portas TCP/UDP. Essa função implementa uma funcionalidade restrita de um analisador de protocolos ou *sniffer*. Uma vez que a captura de pacotes é uma atividade que ocorre em tempo real, as funções de baixo nível do capturador são implementadas em C<sup>++</sup>. O arquivo de saída contém dois campos para cada pacote, o tempo decorrido entre o último pacote capturado e o seu tamanho. A partir dessas informações, o fluxo original pode ser reconstituído no QNet.

### 3.3. Qualidade de Serviço

No Qnet as técnicas de QoS implementadas são MPLS [13] e DiffServ [2], expostas a seguir.

#### 3.3.1. MPLS

A implementação de MPLS no QNet visa a realização de engenharia de tráfego, oferecendo a criação de caminhos explícitos (ER-LSPs) e a criação de caminhos baseados em restrições (CR-LSPs), utilizando o protocolo LDP [1] para a distribuição dos rótulos. O critério utilizado para o estabelecimento de caminhos dos CR-LSPs é a disponibilidade de vazão em todos os LSRs entre a origem e o destino.

O mapeamento entre classes de equivalência (FEC) e rótulos MPLS é necessário para que o *LSR* possa determinar o rótulo que deve ser inserido nos pacotes para que eles sejam encaminhados por determinado LSP. Os critérios disponíveis no QNet são endereços de sistemas finais (*hosts*) e prefixos de rede (no caso de redes de acesso), aplicação geradora de tráfego e interface de entrada no LSR que inicia o LSP.

Para a criação de ER-LSPs é necessária a configuração do caminho explícito, o mapeamento FEC/rótulo e o tempo de ativação do ER-LSP. Para a criação do CR-LSP, existem duas possibilidades, CR com ou sem caminho explícito. No primeiro caso a configuração é semelhante a do ER-LSP, acrescentando apenas a capacidade em bps que será reservada para o CR-LSP. No caso de CR sem caminho explícito, informa-se os LSRs de origem e destino e a capacidade – o sistema descobre automaticamente um caminho que atenda ao requisito de capacidade especificado. A Figura 2 apresenta a interface do Qnet, em um cenário com LSRs (R2, ..., R6), *hosts* (H0) e redes de acesso (ETH0, ..., WLAN0), bem como o gerenciamento de LSPs e a criação de um CR-LSP com caminho explícito.

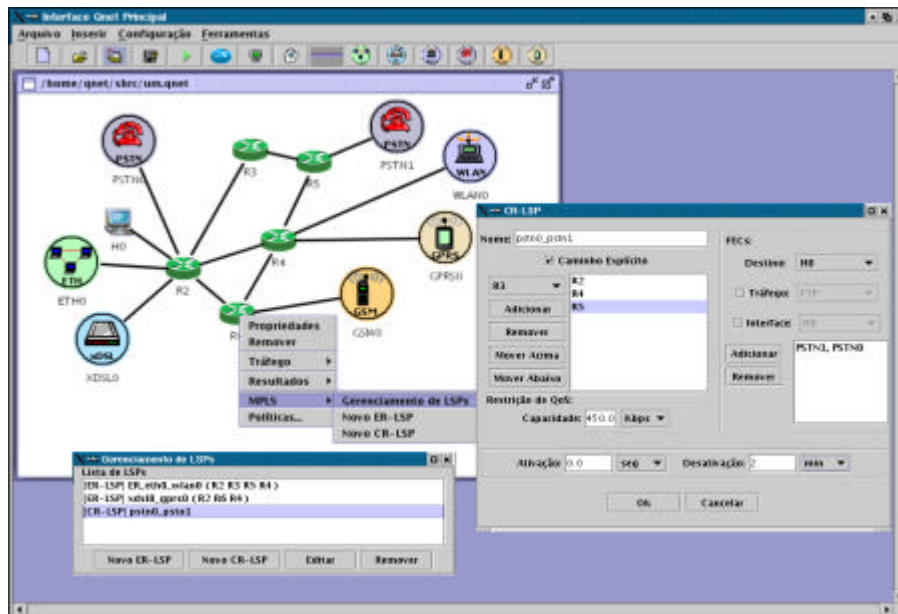


Figura 2. Interface Principal do QNet, Gerenciamento e Criação de *CR-LSPs*.

### 3.3.2. Serviços Diferenciados

A implementação de *DiffServ* no QNet disponibiliza os *PHBs EF* [5], *AF* [9] e *BE*. Cada *PHB* tem uma configuração específica, mas relacionadas entre si, porque utilizam recursos de um mesmo enlace de saída. A configuração dos *PHBs EF* e *AF* é ilustrada na Figura 3.

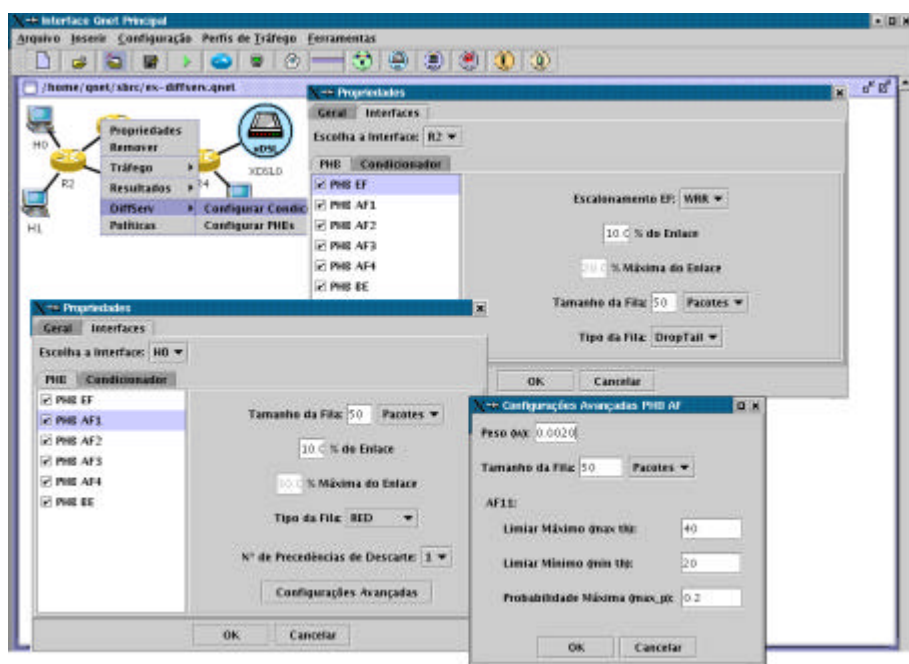


Figura 3. Configuração dos *PHBs EF e AF*.

Os mecanismos para implementar o *PHB EF* podem ser baseados nas disciplinas de serviço *PQ* ou *WRR*. No caso de *WRR*, deve-se também configurar um valor percentual da capacidade do enlace que é atribuída ao tráfego *EF* no escalonamento de pacotes. No caso de *PQ*, o tráfego *EF* tem prioridade total sobre os demais *PHBs*, que irão dividir o restante da capacidade do enlace através de *WRR*. A política de gerenciamento de filas para o *PHB EF* pode ser *DropTail* ou *SFQ*. A configuração do *PHB AF* deve ser realizada para todas as classes *AF* que estiverem disponíveis no roteador. O mecanismo de escalonamento utilizado para dividir a capacidade do enlace entre as classes *AF* e o *PHB BE* é o *WRR*. Deve-se configurar o percentual do enlace para *WRR*, o tipo da fila e o número de precedências de descarte que são utilizados. Para um nível de precedência, os tipos de fila são *RED*, *DropTail* e *SFQ*. Para dois ou três níveis de precedência, os tipos de fila são *RIO* e *WRED*. As configurações avançadas estão relacionadas a parâmetros do *RED*, dados pelo peso, limiar máximo, limiar mínimo e probabilidade máxima para cada nível de precedência ativo (ex. *RED*, só um). O peso refere-se ao tamanho instantâneo que a fila tem no cálculo do tamanho médio, seguindo o modelo de média móvel.

O condicionamento de tráfego é implementado somente na interface de entrada do roteador e implementa os cinco elementos que em geral compõem um condicionador, o classificador, o medidor, o marcador, o descartador e o suavizador, cujas pertinências e configurações são de acordo com o *PHB*. Para uma determinada interface podem ser instalados vários condicionadores, um para cada *PHB*.

O condicionamento de tráfego de um serviço baseado no *PHB EF* é implementado através de um policiador (medidor + descartador), com medição baseada no mecanismo balde de fichas e policiamento das taxas *CIR* (*Committed Information Rate*) e *CBS* (*Committed Burst Size*). Assim, a ação para tráfego fora do perfil é sempre o descarte. O condicionamento de tráfego para um serviço baseado no *PHB AF* pode ser baseada em

balde de fichas ou *SRTCM* (*Single Rate Three Color Marker*) [8]. O condicionamento de tráfego para os *PHBs EF* e *AF* é ilustrado na Figura 4.

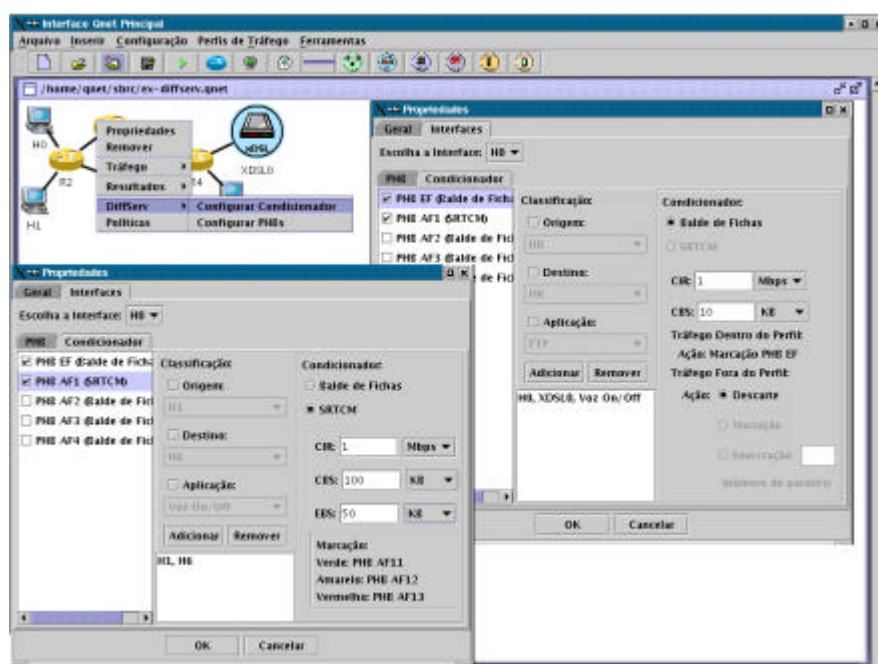


Figura 4. Condicionamento de Tráfego para os *PHBs EF* e *AF*.

### 3.4. Resultados

A configuração de resultados no Qnet permite que o usuário selecione um ponto de medição (sistema final, roteador ou rede de acesso) e então escolha se os resultados desejados se referem a alguma aplicação específica ou a algum elemento de rede (nó).

No caso de resultados por aplicação, o sistema mostra ao usuário quais aplicações estão disponíveis naquele ponto de medição, então o usuário escolhe a aplicação desejada e também a métrica disponível para aquela aplicação. As métricas disponíveis são vazão, perda de pacotes, atraso e variação do atraso. No caso das aplicações baseadas em *TCP*, a métrica “perda de pacotes” não está disponível.

No caso de resultados por elemento de rede, o usuário escolhe a métrica desejada, a direção da medição, a granularidade e um filtro de medição. As métricas para resultados por elemento podem ser vazão, perda de pacotes e tamanho das filas. A direção pode ser referente à entrada ou à saída de pacotes do elemento de rede. Quanto a granularidade, a coleta dos resultados pode ser para uma interface específica (sistema final, rede de acesso ou roteador conectado na outra ponta do enlace), ou global (todas as interfaces). O filtro, opcional, é de acordo com alguma classificação de *QoS* (*LSP – MPLS* ou *PHB – DiffServ*).

Existem duas possibilidades de visualização de resultados, dadas por gráfico e tabela. É importante salientar que no gráfico de vazão, os pontos com dados correspondem ao intervalo de medição que foi selecionado pelo usuário no momento da execução da simulação. Para as métricas atraso e variação do atraso (*jitter*), as amostras são coletadas para todos os pacotes, de modo que o gráfico torna-se mais denso. As métricas atraso e *jitter* podem ser combinadas em uma única visualização, uma vez que sua apresentação conjunta tanto é

necessária com frequência quanto é coerente em termos semânticos. Nenhuma outra combinação de métricas pode ser apresentada simultaneamente no mesmo gráfico, devido à incompatibilidade de unidades. A Figura 5 ilustra a visualização de resultados no QNet.

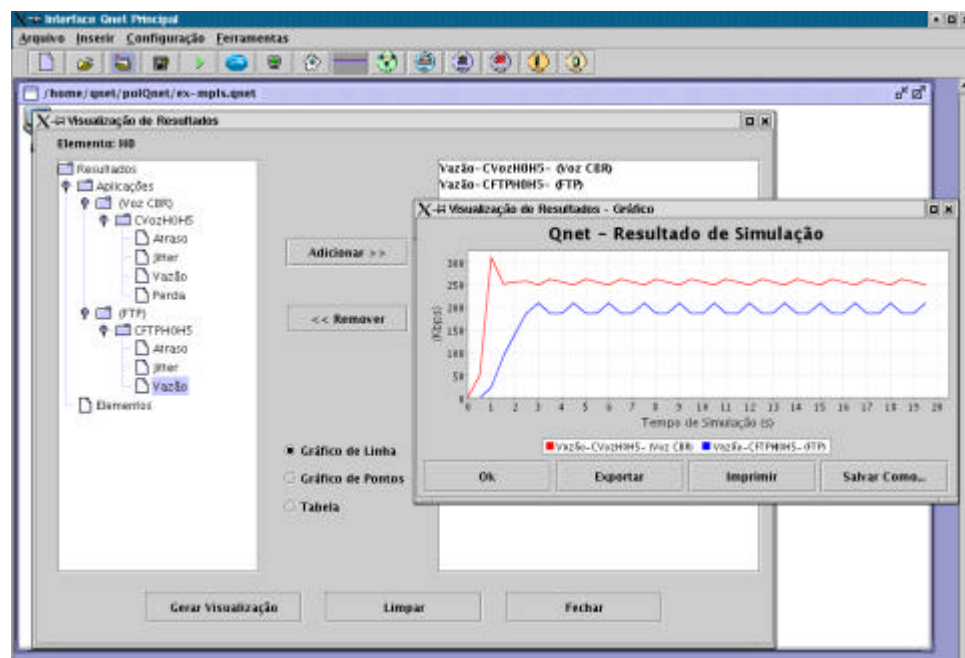


Figura 5. Visualização de Resultados no Qnet.

#### 4. Conclusão

Este artigo apresenta o simulador QNet, que está sendo desenvolvido com o intuito de fornecer uma ferramenta adequada para auxiliar o trabalho de pesquisadores e profissionais da área de redes de computadores. A maior contribuição do QNet é oferecer características como simplicidade de uso, baixo custo e funcionalidades para simular redes convergentes, em uma única ferramenta, com interface gráfica. O QNet permite a configuração de topologias compostas de redes de acesso (diversas tecnologias) e backbone. Roteadores das redes de backbone podem ser configurados com DiffServ ou MPLS, para prover QoS e/ou realizar Engenharia de Tráfego. Os resultados de simulação podem ser facilmente configurados e visualizados, principalmente através de gráficos.

O QNet está ainda em desenvolvimento, mas atualmente já vem sendo avaliado na resolução de problemas reais de operadoras de telecomunicações. Várias funcionalidades serão incorporadas, visando tornar a sua utilização ainda mais adequada ao seu objetivo. Entre essas funcionalidades estão a utilização de gerenciamento baseado em políticas para provisionamento dinâmico de *DiffServ* e *MPLS*, a disponibilidade de técnicas de previsão e modelagem de tráfego e a adição de técnicas de simulação estocástica.

#### Referências

- [1] Andersson, L. *et al.*, "LDP Specification", RFC 3036, Janeiro de 2001.
- [2] Blake, S. *et al.*, "An Architecture for Differentiated Services", RFC 2475, Dezembro de 1998.



- [3] Cisco, “Distributed Weighted Random Early Detection”, White Paper, <http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.pdf>, acessado em 24.03.2004.
- [4] Clark, D. & Fang, W., “Explicit Allocation of Best-Effort Packet Delivery Service”, IEEE/ACM Transactions on Networking, vol. 6, no 4, Agosto de 1998.
- [5] Davie, B. et al., “An Expedited Forwarding PHB (Per-Hop Behavior)”, RFC 3246, Março de 2002.
- [6] Floyd, S. and Jacobson, V., “Random Early Detection gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, vol. 1, no 4, Agosto de 1993.
- [7] Grossglauser, M. & Bolot, J-C, “On the Relevance of Long-Range Dependence in Network Traffic”, IEEE/ACM Transactions on Networking, Outubro de 1999.
- [8] Heinanen, J. & Guerin, R., “A Single Rate Three Color Marker”, RFC 2697, Setembro de 1999.
- [9] Heinanen, J. et al., “Assured Forwarding PHB Group”, RFC 2597, Junho de 1999.
- [10] McKenney, P., “Stochastic fairness queueing”, In Internetworking: Research and Experience Vol.2, 113-131, Janeiro de 1991.
- [11] Network Simulator, “Network Simulator Web Site”, <http://www.isi.edu/nsnam/ns>, acessado em 24.03.2004.
- [12] OPNET, “OPNET Web Site”, <http://www.opnet.com>, acessado em 24.03.2004.
- [13] Rosen, E., Viswanathan, A. & Callon, R., “Multiprotocol Label Switching Architecture”, RFC 3031, Janeiro de 2001.