

HYDRAGYRUM – AMBIENTE DE SIMULAÇÃO DE REDES A EVENTOS DISCRETOS

Ernesto L. Andrade Neto, Antônio M. Alberti, Leonardo S. Mendes

Departamento de Comunicações – Faculdade de Engenharia Elétrica e de Computação – UNICAMP
Caixa Postal 6101, CEP: 13081-970, Campinas SP – Brasil

Telefone: 55 0 xx 19-7883703, Fax: 55 0 xx 019-2891395, ernestoal, alberti, lmendes@decom.fee.unicamp.br

Resumo

Este artigo descreve a arquitetura do ambiente de simulação de redes Hydragyrum. O Hydragyrum emprega em sua arquitetura os conceitos de modelagem orientada a objetos, implementados através da linguagem C++, para criar um ambiente de simulação baseado em eventos discretos. A arquitetura do Hydragyrum foi projetada para ser modular e expansível como forma de permitir a adição de novos modelos ao ambiente através do uso de *dlls* para a representação dos modelos. A arquitetura desenvolvida para o Hydragyrum tornou-se uma plataforma aberta para o desenvolvimento de simulações baseadas na técnica de eventos discretos e troca de mensagens. Abrangendo uma vasta gama de problemas de simulação, incluindo sistemas baseados em fluxo de dados ou sinais. Na sua implementação atual, o Hydragyrum suporta o sistema operacional WindowsTM (9x/NT/2000) fazendo uso de seus recursos de interface gráfica e *dlls*. O Hydragyrum é empregado atualmente na simulação de desempenho de redes ATM e mostramos os resultados obtidos com simulações de modelos de rede ATM desenvolvidos para a versão atual do Hydragyrum.

1. INTRODUÇÃO

A implementação de grandes redes de comunicação envolve uma vasta gama de tecnologias e protocolos com diferentes graus de compatibilidade e interoperabilidade. Neste ambiente heterogêneo o projeto de novas redes, alterações nos protocolos já existentes ou a implementação de novos serviços de rede geralmente demandam a construção de uma rede de testes. Estas redes de teste são usadas inicialmente para a validação de uma nova tecnologia e para a análise de sua aplicabilidade e expansibilidade. Entretanto, as redes de teste podem possuir um custo extremamente elevado dependendo de sua extensão e do tipo de tecnologia empregada. Outro aspecto importante do ponto de vista da engenharia das redes de comunicação é a questão do projeto e da expansão de redes baseadas em tecnologias já existentes que requerem uma análise detalhada do tipo de tráfego e dos requisitos de tráfego que a rede deve suportar. No desenvolvimento de redes de comunicação, podemos empregar a simulação como ferramenta de análise da viabilidade da implementação ou alteração de uma determinada tecnologia, sem a necessidade da construção de uma rede de testes real. A simulação de redes de comunicação também é uma forma econômica de realizar o projeto de uma rede, visando sua instalação ou expansão, sem necessidade de investimento a priori na compra de equipamentos de rede.

No sentido de prover uma ferramenta para a simulação de redes de comunicação, que permitisse atender as necessidades de projetistas e pesquisadores da área de redes de comunicação, iniciamos o desenvolvimento de simuladores a eventos discretos [1] baseados em [2], que evoluíram para o ambiente de simulação de redes denominado Hydragyrum. O ambiente de simulação de redes Hydragyrum consiste basicamente de um *kernel* (núcleo de processamento), o qual implementa o método de simulação de eventos discretos para caracterizar o comportamento de redes de comunicação de dados (protocolos e tráfego de dados). A este *kernel*, são adicionados os conjuntos de modelos dos elementos de rede da tecnologia a ser simulada. Esta estrutura desacoplada é ideal quando consideramos a grande multiplicidade de tecnologias e protocolos presentes nas redes de comunicação e os diferentes níveis de interação existentes. Pois, a estrutura desacoplada permite o refinamento e aprimoramento dos modelos sem a necessidade de alterações no ambiente de simulação. Somando-se a estrutura modular do *kernel* o ambiente de simulação possui uma interface gráfica que permite a manipulação de parâmetros dos modelos de forma simplificada e a definição das topologias de rede para simulação. O ambiente de simulação conta também com uma interface de linha de comando que é ideal para o processamento de simulações através de roteiros (*scripts*), usados principalmente em simulações de múltiplos cenários de rede (variações de parâmetros de modelos e/ou topologias de rede).

O Hydragyrum foi desenvolvido a princípio com o objetivo de simular o comportamento da tecnologia de redes ATM [4][5][6]. Desenvolvemos inicialmente um conjunto de modelos para avaliar o comportamento do desempenho de uma rede ATM. Estes modelos, analisam o tráfego de células na rede e a influência das disciplinas de serviço presentes na rede ATM (baseadas em classes de prioridade) sobre este tráfego. Novos modelos estão sendo desenvolvidos para incorporar os aspectos de gerenciamento de tráfego da rede ATM, como forma de avaliar o impacto dos mecanismos de controle da rede sobre o tráfego oferecido e representar de forma mais precisa o atendimento de requisitos de qualidade de serviço (QoS) de aplicativos que utilizam a rede ATM como meio de transporte. Existem vários softwares para o modelamento e a simulação de sistema de comunicação [3], entretanto o Hydragyrum se destaca por ser um software não comercial disponível para a utilização no ambiente WindowsTM.

Este artigo está organizado como segue. Na segunda seção, apresentamos a arquitetura do ambiente de simulação e os detalhes de implementação. Na terceira seção, mostramos os aspectos da construção de conjuntos de modelos para o ambiente

de simulação. Na quarta seção, são mostrados os modelos de tráfego de células construídos para a simulação de redes ATM. Na quinta seção, é mostrado um exemplo de simulação de redes ATM utilizando o ambiente de simulação. Na sexta seção, são apresentadas as conclusões e as considerações sobre os novos rumos do desenvolvimento.

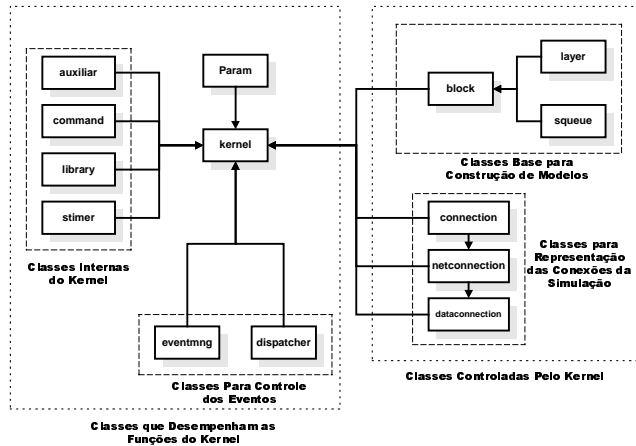


Figura 1. Arquitetura do ambiente de simulação.

2. ARQUITETURA DO AMBIENTE DE SIMULAÇÃO

A modelagem por objetos foi empregada na implementação do Hydragyrum para facilitar o desenvolvimento da estrutura complexa e rica em detalhes dos elementos do ambiente de simulação. O emprego da modelagem por objetos permite o encapsulamento dos diferentes comportamentos dos elementos do *kernel* dentro de classes específicas. O *kernel* do ambiente de simulação foi desenvolvido em linguagem C++ [8] nos compiladores Borland C++ 5.02 e Microsoft Visual C++ 6.0. O método de simulação a eventos discretos empregado no ambiente de simulação é implementado através de uma estrutura de fila de prioridades onde o evento que está no topo da fila sempre possui o menor tempo agendado. O *kernel* retira este evento do topo da fila de prioridades e o encaminha ao destino apropriado. Assim, os modelos podem agendar e processar eventos utilizando intervalos de tempo discretos, determinados pela lógica dos modelos, para representar o comportamento de um elemento de rede real. O processamento de eventos na implementação do Hydragyrum é realizado de forma seqüencial para tornar mais simples a arquitetura do ambiente e aumentar sua portabilidade. Na Figura 1, podemos observar um esquema da estrutura de classes do *kernel* do Hydragyrum, notando os vários elementos presentes no ambiente de simulação e sua interação com o *kernel*. O *kernel* é subdividido em várias classes auxiliares que desempenham funções específicas como: interpretar comandos da interface, gerenciar e distribuir eventos, carregar e descarregar as dlls dos modelos da memória, entre outras funções. A maior parte das tarefas próprias do *kernel* tratam principalmente com o armazenamento e gerenciamento dos objetos que representam a topologia da rede simulada,

basicamente blocos e conexões, mostrados na Figura 1. Os modelos de elementos de rede são representados dentro do ambiente de simulação através de blocos (*block*), que por sua vez podem ainda conter instâncias de filas (*squeue*) ou camadas (*layer*), para facilitar a subdivisão lógica do modelo de elemento de rede. Os modelos de elementos de rede possuem uma estrutura de dados denominada parâmetro, que pode conter diferentes tipos de dados. Estes parâmetros, podem ser manipulados confortavelmente através da interface gráfica para alterar o comportamento dos modelos. Os objetos que representam as conexões no ambiente de simulação foram implementados de modo a prover uma abstração de conexões com uma hierarquia de camadas, como as que fazem parte do modelo de referência ISO/OSI. Existem três objetos de conexão no ambiente de simulação: conexão de camada física ou de enlace (*connection*), conexão de camada de rede (*netconnection*) e conexão de camada de transporte ou superior (*dataconnection*). Na Figura 2, é mostrado o inter-relacionamento dos três objetos de conexão, as conexões de rede são compostas de conexões físicas ou de enlace que descrevem um caminho “físico” através da rede. As conexões de dados podem utilizar uma ou mais conexões de rede disponíveis para enviar dados de um modelo a outro na topologia de rede simulada. Este objetos são facilmente manipuláveis na interface gráfica e podem ser usados de diferentes formas em conjuntos de modelos que necessitem implementar uma hierarquia de conexões.

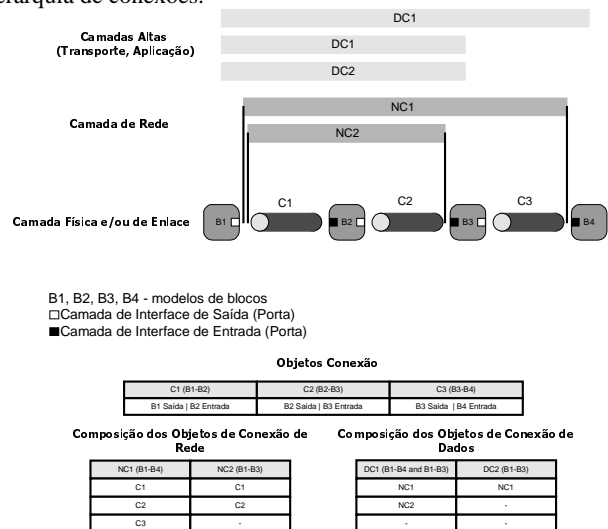


Figura 2. Estrutura hierárquica de conexões.

Além dos modelos e das conexões, outros elementos presentes no ambiente de simulação são as estruturas de dados utilizadas pelo *kernel* para troca de informação entre o *kernel* e os modelos. Tendo como exemplo, as estruturas que definem os eventos e as mensagens trocadas entre os modelos e as listas e vetores de objetos controlados pelo kernel (modelos (blocos), camadas, filas e conexões) que podem ser manipulados pelos modelos quando for necessário.

2.1 Interfaces do Ambiente de Simulação

O ambiente de simulação conta com uma interface gráfica mostrada na Figura 3. A interface gráfica foi desenvolvida

utilizando-se o Borland C++ Builder 3.0. Esta interface gráfica com o usuário (GUI), permite a construção de topologias de rede para simulação utilizando-se os modelos presentes na biblioteca de modelos, que são acessados através do formulário de seleção de modelos. Podemos também alterar os parâmetros dos modelos da topologia utilizando o formulário de manipulação de parâmetros ou clicando duas vezes sobre o ícone do modelo para ter acesso a seus parâmetros. A interface gráfica permite gravação e leitura de arquivos de topologia e a execução da simulação com apresentação dos resultados gerados pelos modelos na forma de gráficos como mostrado na Figura 4.

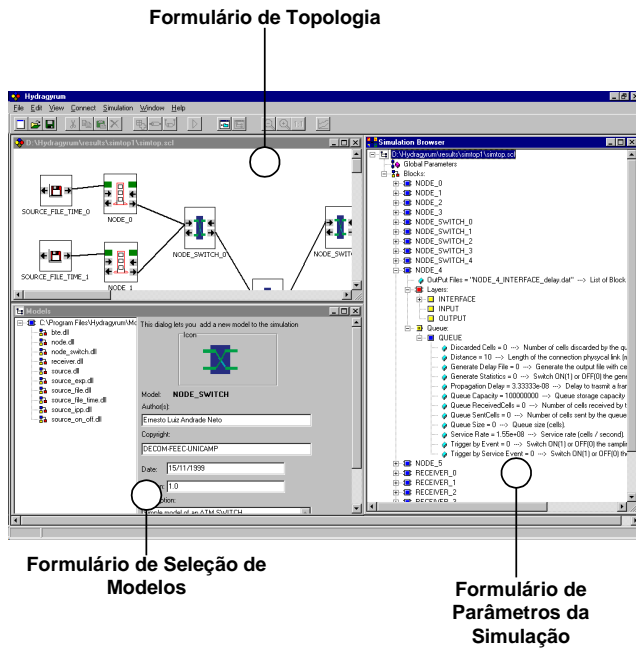


Figura 3. Interface gráfica do ambiente de simulação.

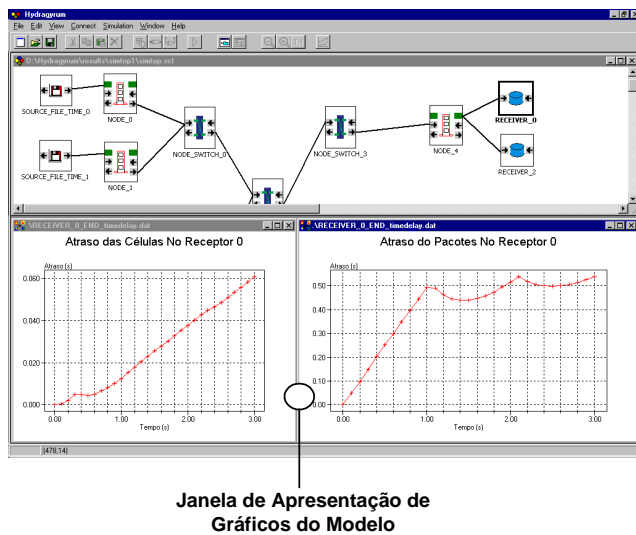


Figura 4. Apresentação de gráficos no ambiente de simulação.

Além da interface gráfica o ambiente de simulação provê uma interface de linha de comando que permite a configuração e

execução das simulações utilizando arquivos de roteiro (*scripts*). Este recurso é muito útil ao realizarmos várias simulações complexas e demoradas, quando é necessária a alteração de parâmetros dos modelos para cada execução da simulação.

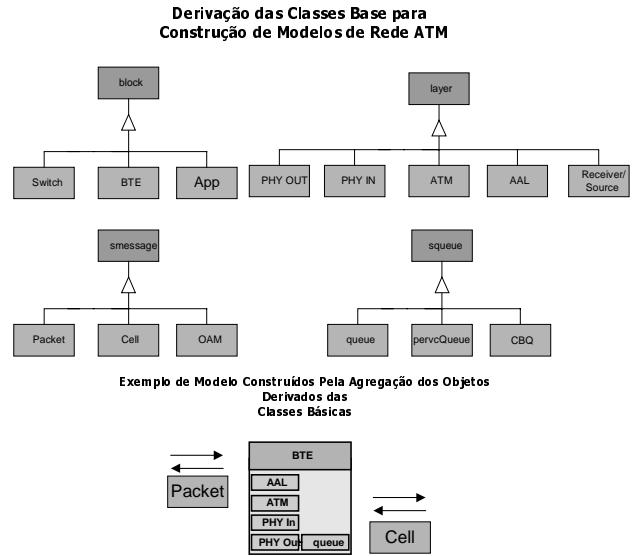


Figura 5. Classes para a composição de conjunto modelos ATM.

3. CONSTRUÇÃO DE CONJUNTO DE MODELOS

Os modelos para o ambiente de simulação são construídos apartir das classes base de derivação de modelos (*block*, *layer* e *queue*). Sendo a classe *block* a classe que contém: (i) a estrutura do modelo (organizada em filas e camadas); (ii) as definições da apresentação do modelo na interface gráfica (tipo e número de portas); e (iii) a interface para a carga da *dll* do modelo através da biblioteca de modelos. Na construção de um conjunto de modelos, inicialmente devemos especificar os vários elementos a serem derivados da classe container bloco, que representarão os modelos em si, e os sub-elementos que comporão os modelos (camadas e filas derivadas das classes base apropriadas). O outro passo importante para a construção do conjunto de modelos é a definição das mensagens que serão transportadas pelos eventos entre os modelos do conjunto. Estas mensagens, são derivadas da classe base *message* e permitem o transporte, através de ponteiros contidos na estrutura do evento, de qualquer tipo de dado derivado de *message* que possa ser definido na linguagem C++. Portanto, empregamos fortemente o conceito de polimorfismo na troca de mensagens no ambiente de simulação. A Figura 5, mostra um exemplo de derivação de classes para a construção de um conjunto de modelos de redes ATM e a composição do modelo de BTE (Equipamento Terminal de Banda Larga – *Broadband Terminal Equipment*) utilizando as camadas e filas derivadas para sua composição e sendo capaz de enviar e receber as mensagens *Packet* e *Cell* através de eventos.

Após a definição da estrutura dos modelos, o passo seguinte no modelamento é a introdução da lógica e dos algoritmos que retratam o comportamento do modelo. Para realizar esta tarefa, as

classes base oferecem ao projetista de modelos uma interface virtual que pode ser personalizada para representar as ações do modelo dentro do ambiente quando recebe mensagens de outros modelos ou quando processa comandos do *kernel*. As funções que compõem esta interface estão descritas na Tabela 1 e permitem ao projetista de modelos a integração e o controle dos modelos dentro do ambiente de simulação.

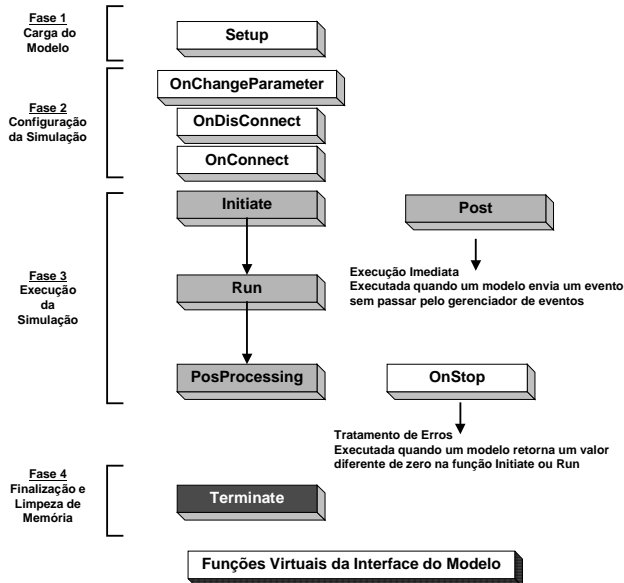


Figura 6. Seqüência padrão das fases da simulação.

Função	Instante ou Motivo da Chamada
Setup	carga da dll do modelo
Initiate	inicialização da simulação
Run	recepção de evento
Post	recepção direta de evento
PosProcessing	término da simulação
OnStop	ocorrência de erro
OnChangeParameter	mudança de parâmetro
Terminate	deleção do modelo
OnConnect/ OnDisconnect	criação/ destruição de conexões

Tabela 1: Funções de interface virtual dos modelos.

As várias funções de interface são chamadas em diferentes fases da simulação ou quando ocorrem ações do kernel sobre o modelo. Podemos observar a seqüência padrão destas fases da simulação na Figura 6.

4. MODELOS PARA REDE ATM

A aplicação inicial do Hydragyrum é a simulação de redes ATM para este fim foram desenvolvidos conjuntos que simulam a rede ATM em nível de células. Os modelos desenvolvidos para a avaliação do tráfego de células na rede ATM foram: (i) NODE,

que representa o terminal de banda larga da rede ATM; (ii) NODE_SWITCH, que representa o nó chaveador (SWITCH) da rede ATM; (iii) RECEIVER, que representa o aplicativo receptor de uma conexão de dados que usa a rede ATM como transporte; e (iv) um conjunto de vários modelos de fonte de tráfego [8] para rede ATM, que incluem fontes on-off, ipp, poissonianas, constantes e de leitura de tráfego armazenado em arquivos.

O modelo do NODE, implementa a camada de adaptação AAL5 [7] e possui um enlace de saída ATM que pode ter sua taxa controlada como parâmetro, portas de saída e portas de entrada. O modelo do NODE_SWITCH, possui portas de entrada e portas de saída com sistemas de filas ATM (enlaces) associados e uma camada de chaveamento que controla o encaminhamento das células entre as portas de entrada e saída. O chaveamento das células é realizado em um intervalo constante de tempo e não existe bloqueio de células no chaveador. Em todos os sistemas de filas implementados nos modelos a disciplina de serviço é baseada em classes de prioridade (*Class Based Queueing*). Os modelos de fontes possuem uma série de parâmetros de controle específicos para cada tipo de fonte. As conexões entre fonte e receptor através da topologia de rede ATM são estabelecidas manualmente utilizando-se os recursos da interface gráfica. Nos sistemas de fila presentes nos modelos e no receptor de tráfego é possível realizar a coleta de estatísticas dos modelos durante a simulação. Nos vários sistemas de filas é possível observar a variação das estatísticas do sistema de filas (número de células recebidas, atraso (instantâneo ou médio) no sistema de fila, tamanho da fila (instantâneo ou médio), utilização, etc. ...), realizando amostragem destas estatísticas para arquivos a intervalos de tempo pré-determinados ou toda vez que ocorrer um evento de chegada ou saída no sistema de filas. No receptor, podemos colher as estatísticas para os atrasos de propagação da células e dos pacotes através da rede durante a simulação.

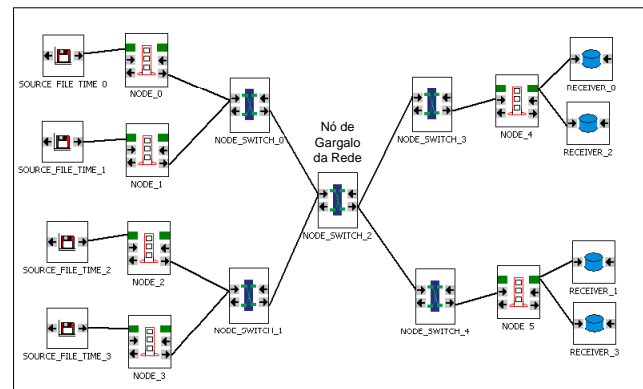


Figura 7. Topologia de rede ATM simulada.

5. SIMULAÇÃO DE REDE ATM

A análise de desempenho de uma rede ATM pode ser empregada para dimensionar os enlaces da rede. Na simulação apresentada realizamos uma análise da carga que pode ser fornecida a um nó de gargalo da rede por quatro aplicativos distintos sem que o atraso destes aplicativos comece a tornar-se excessivo (no caso superior a 10µs de atraso médio de transmissão por célula). A configuração da topologia de rede simulada é mostrada na Figura 7. Na rede as fontes numeradas de 1 a 4 transmitem para os

receptores com o mesmo número da fonte. Esta configuração das conexões provoca a concorrência do tráfego pelos recursos da rede. Para analisar o desempenho da rede, utilizamos quatro fontes de dados que lêem de arquivo traços de tráfego com diferentes graus de correlação gerados com o método descrito em [10], representado o tráfego de uma rede Ethernet. A taxa de média de bits de cada fonte foi variada de modo a variar a carga agregada nos enlaces de saída do nó de gargalo (NODE_SWITCH_2). Todos os enlaces de rede ATM simulados possuem taxa de 155 Mbps. Os valores medidos durante a simulação para a utilização média dos enlaces de saída do nó de gargalo são mostrados na Tabela 2.

Taxa por fonte (Mbps)	Utilização dos Enlaces de Saída do Nó de Gargalo (SW2) (155Mbps)	
	Enlace B_6	Enlace B_7
3,4	0,052540	0,052261
10,2	0,155960	0,155868
17,0	0,259933	0,259780
23,8	0,363906	0,363692
30,6	0,467878	0,467603
34,0	0,519864	0,519558
37,4	0,571848	0,571510
44,2	0,675813	0,675411
51,0	0,779778	0,779309

Tabela 2. Valores simulados da utilização média dos enlaces de saída do nó de gargalo em função da taxa individual das fontes de tráfego.

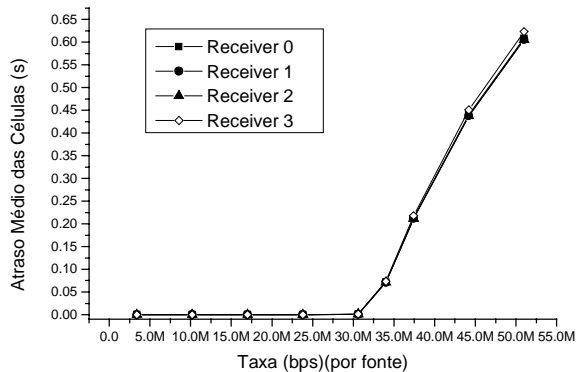


Figura 8. Atraso total de transmissão de fim a fim dos aplicativos em função da taxa de transmissão.

Nesta simulação, sempre duas fontes de tráfego concorrem pelo mesmo enlace de saída no switches da rede. Em cada enlace de saída do nó de gargalo, observamos a concorrência dos tráfegos pela utilização do enlace de saída, para determinarmos o desempenho da rede como um todo. Na Figura 8, observamos que o limite de atraso desejado para a transmissão das células na rede (10µs) começa a ser violado quando as fontes passam a transmitir a uma taxa média individual superior a 30,6 Mbps por

fonte de tráfego, que equivale a uma utilização de 47% dos enlaces de saída do nó de gargalo.

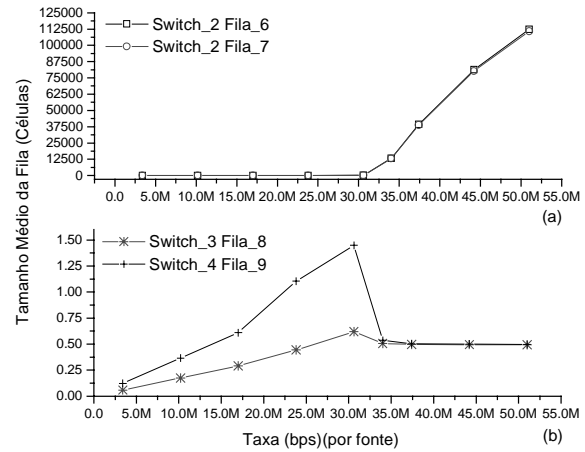


Figura 9. Tamanho médio da fila nos switches em função da taxa de transmissão dos aplicativos.

Na Figura 9 (a), podemos observar a variação do tamanho médio das filas dos enlaces de saída do nó de gargalo que começam a aumentar a partir de 30,6 Mbps por fonte, quando os enlaces passam a ficar congestionados aumentando o atraso das células na rede. Comparando a Figura 9 (a) com a Figura 9 (b), que mostra o tamanho médio das filas nos enlaces de saída dos nós posteriores ao nó de gargalo da rede (NODE_SWITCH 3 e 4), observamos que o nó de gargalo domina o congestionamento da rede. Pois, os tamanhos médios das filas no nó de gargalo (Figura 9 (a)) são superiores aos tamanhos médios das filas dos nós posteriores (Figura 9 (b)). Quando atingimos a situação de congestionamento no nó de gargalo, o comportamento do tamanho médio das filas dos nós posteriores ao nó de gargalo torna-se uniforme a partir de aproximadamente 37,4 Mbps. Isto, é devido a fato da congestão no nó de gargalo provocar um fluxo contínuo de células na taxa nominal do enlace (155 Mbps) para cada uma das switches NODE_SWITCH 3 e 4.

Na topologia simulada, outra análise que pode ser realizada é a variação das classes de prioridade de uma das fontes de tráfego em relação as outras para diminuir o atraso de transmissão da fonte de tráfego. Na Figura 10, observamos o efeito de atribuirmos uma prioridade maior ao tráfego entre a fonte 3 e o receptor 3 em relação as outras classes utilizando o mecanismo de *Class Based Queueing* modelado nos NODE_SWITCHS. Constatamos que o atraso médio de transmissão é bem menor para a fonte 3 quando esta possui uma prioridade maior em relação as outras fontes. E o atraso da fonte 3 também é menor quando o comparamos ao caso em que todas as fontes possuem a mesma prioridade. Nos resultados mostrados na Figura 10, as simulações com mesma prioridade para todas as fontes e em seguida com prioridade maior para a fonte 3, foram realizadas para uma taxa média de 3,4 Mbps por fonte. Em todas as

simulações realizadas não foram impostas limitações para a capacidade de armazenamento dos sistemas de filas simulados.

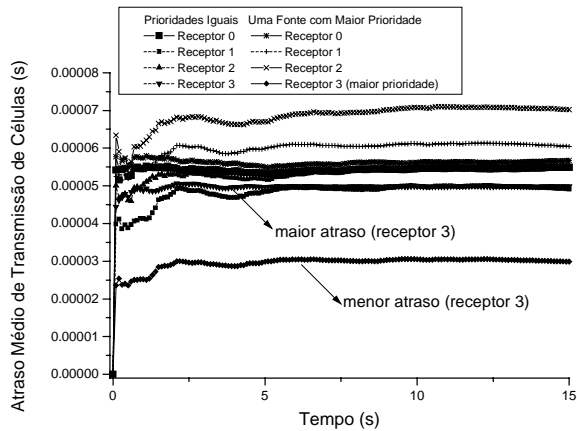


Figura 10. Comparação do uso de classes de tráfego para redução do atraso de transmissão de fim a fim.

6. CONCLUSÃO

O ambiente de simulação desenvolvido é uma ferramenta extremamente flexível para a simulação de redes de comunicação comparando-se em termos de arquitetura a sistemas comerciais de simulação de rede atualmente disponíveis como [11]. O Opnet [11], possui as mesmas premissas de uma arquitetura aberta e expansível, entretanto o desenvolvimento de novos modelos é realizado pela empresa e não pelos usuários da ferramenta na maioria das vezes. O Hydragyrum por sua vez permite que o usuário do ambiente de simulação, respeitando as regras de construção de modelos e possuindo conhecimento de programação em linguagem C++, desenvolva novos modelos e conjuntos de modelos para o ambiente. Esta possibilidade de adição por parte do usuário de novos modelos não está prevista em softwares disponíveis comercialmente como COMNET III [12].

O Hydragyrum tem permitido a análise de desempenho de redes ATM em nosso grupo de pesquisa e está atualmente sendo expandido para incluir os aspectos de gerenciamento de tráfego da rede ATM. O Hydragyrum está disponível para pesquisadores e projetistas interessados em desenvolver simulações baseadas no método de eventos discretos no endereço eletrônico <http://www.mc21.fee.unicamp.br/hydragyrum>.

7. Agradecimentos

Agradecemos a FAPESP pelo suporte dado aos nossos projetos de pesquisa na área de simulação de redes sob os contratos 96/12372-2 e 97/14449-5.

8. REFERÊNCIAS

[1] A. M. Alberti, “SimATM: Um Ambiente para a Simulação de Redes ATM”, Tese de Mestrado, UNICAMP, Abril 1998.

[2] N. Golmie, A. Koenig e D. Su, “The NIST ATM Network Simulator”, Operation and Programming, Version 1.0, NIST Internal Report 5703, 1995.

[3] A. M. Alberti, E. L. Andrade, L. S. Mendes “Simulação de Redes ATM”, XVII SBT, 1999.

[4] A. Alles, “ATM Internetworking”, White Paper, Cisco Systems, Inc., 1995.

[5] U. Black, “ATM: Foundation for Broadband Networks”, Prentice-Hall, 1995.

[6] G. Sackett e C. Metz, “ATM and Multiprotocol Networking”, McGraw Hill, 1997.

[7] ITU-T Recommendation I.363, “B-ISDN ATM Adaptation Layer (AAL) Specification”, Helsinki, 1993.

[8] B. Stroustrup, “The C++ Programming Language”, Addison-Wesley, 3rd Edition, 1997.

[9] Adas, A., “Traffic Models in Broadband Networks”, IEEE Communications Magazine, July 1997.

[10] E. Andrade, A. Alberti, D. Arantes, L. Mendes, “A Realistic Model for Self-Similar Ethernet LAN Traffic in SimATM – an ATM Network Simulator: Design and Performance Implications”, Annals of International Telecommunications Symposium 1998, São Paulo, Brasil.

[11] <http://www.opnet.com>

[12] M. Falkner, “Modeling ATM Networks with COMNET III”, COMNET III Application Notes, Version 1.0, August 1996.