

Um Ambiente de Simulação de Fluido para Redes Multimídia *

Kelvin F. Reinhardt^{1†}, Edmundo de Souza e Silva¹, Rosa M. M. Leão¹

¹COPPE - Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Caixa Postal 68511, CEP: 21941-972–Rio de Janeiro, RJ
kelvin,edmundo,rosam@land.ufrj.br

Abstract. *An important issue to solve computer network models using traditional simulation methods is the computational cost to obtain the measures of interest. This is due to the very large number of events that have to be executed till the model reaches steady state. To reduce the simulation costs new techniques have been developed in the last few years. One of the most promising techniques is fluid simulation. In this technique, instead of representing the individual packets in the network, the traffic is viewed as a fluid that flows through a series of reservoirs. Therefore, events associated with the arrival and departure of packets to and from routers are not explicitly executed. Instead, only the resulting actions from changes in arrival and departure rates are executed and the equations that govern the changes in the fluid levels in each reservoir are solved. This can speed up the simulation time to solve a model. In this work, a fluid modeling paradigm based on rate rewards is devised. The paradigm was implemented in a fluid simulator for computer networks. The fluid simulator was built over the TANGRAM-II simulator and as such several new features were added to handle fluids. The new fluid simulator inherited all the facilities and the power of the TANGRAM-II tool. It is generic enough to model complex systems and yet hides the complex fluid equations from the end user.*

Resumo. *Os métodos tradicionais de simulação, utilizados para modelagem de redes de computadores, enfrentam problemas em relação ao custo computacional, já que as redes atuais possuem uma grande capacidade de transmissão e em geral grandes dimensões. Para minimizar este problema surgiram diversas técnicas, entre elas, a simulação de fluido, onde o tráfego entre os nós de rede é tratado de forma contínua, em vez de unidades discretas. Sua principal vantagem está no fato de que somente as mudanças de taxas precisam ser tratadas. Neste trabalho, foi criado um paradigma de modelagem de fluido baseado em recompensas de taxa. De acordo com este paradigma, foi idealizado, projetado, construído e validado, um simulador de fluido com foco na área de redes. Este foi construído sobre o simulador da ferramenta TANGRAM-II, onde foram adicionadas diversas funcionalidades. O simulador de fluido herdou as características e o poder de modelagem do TANGRAM-II e de acordo com seus princípios, é genérico o suficiente para modelar praticamente qualquer tipo de sistema, ao mesmo tempo que possui objetos de rede prontos para o uso de forma rápida e simples.*

*Este trabalho é parcialmente financiado pelo CNPq e FAPERJ.

† A CAPES forneceu bolsa de mestrado para K. Reinhardt

1. Introdução

A simulação é uma das ferramentas mais usadas na avaliação de mecanismos e algoritmos desenvolvidos para as redes de computadores. Entretanto, o drástico aumento da complexidade e velocidade de transmissão resulta em um aumento significativo no tempo de simulação necessário para se obter as medidas de interesse. O paradigma mais usado na simulação é a orientação a eventos, ou seja, o sistema reage sempre que há a ocorrência de um evento e o tempo de simulação salta a cada execução de um evento. Em redes de computadores o conjunto de eventos inclui a chegada de um pacote na fila de um canal de um roteador, a transmissão de um pacote por um canal, a mudança de taxas de transmissão de pacotes devido a entrada ou saída de usuários da rede, entre outros.

Não é difícil observar o problema causado pelo simples aumento de velocidade de transmissão em relação a outros eventos na rede. Consideremos um modelo simples de uma fonte de pacotes alimentando uma fila. A fonte gera pacotes a uma taxa variável de zero a 1 Gbps. As mudanças de taxa ocorrem na ordem de segundos. A capacidade de transmissão de pacotes da fila é de 200 Mbps, e os pacotes tem tamanho médio de 2K bits. Devido as altas taxas de chegada e saída de pacotes, 15 minutos de simulação requerem a execução da ordem de 10^8 a 10^9 eventos. Esse tempo pode ser necessário para se obter medidas tais como a probabilidade de descarte de pacotes devido ao enchimento da fila, com um valor razoável de intervalo de confiança. A partir deste exemplo, não é difícil imaginar o custo computacional quase proibitivo para simular um número grande de fontes TCP emitindo pacotes a uma taxa de centenas de megabits através de uma rede com vários roteadores.

Nos últimos anos algumas técnicas foram desenvolvidas visando diminuir o custo computacional para se obter as medidas de interesse. Quando a medida em questão é relacionada a um evento que ocorre com pouca frequência em relação a outros eventos no sistema (por exemplo o evento descarte de pacotes em relação aos eventos de transmissão e geração), técnicas como a denominada de *importance sampling* e *splitting/RESTART* podem ser empregadas. (Ver [Figueiredo, 1999] para referências destas técnicas e um estudo do método de *splitting*.) Outra técnica para reduzir o custo computacional da simulação consiste em um tipo de simulação híbrida chamada de *Time-stepped Hybrid Simulation* [Guo et al., 2000]. Técnicas adicionais como: cálculos preliminares, agregação de tráfego, métodos de extrapolação, estão brevemente descritas em [Kesidis and Singh, 1995].

Mesmo que a medida a ser calculada não esteja diretamente relacionada à ocorrência de um evento raro, é possível que um número muito elevado de eventos tenha de ser gerado para que se possa obter a medida de interesse com um intervalo de confiança razoável. Quando no modelo existirem taxas ordens de grandeza distintas entre a ocorrência de eventos, a chamada simulação de fluido pode se tornar atrativa [Kumaram and Mitra, 1998].

Um modelo estocástico de fluido é um sistema onde a entrada é um fluido contínuo que entra e deixa um dispositivo de armazenamento de acordo com taxas variáveis e aleatórias. Retornemos ao exemplo da fonte de pacotes alimentando uma fila descrito acima. Em um modelo de fluido a geração de pacotes é representada por um fluido e o número de pacotes na fila é obtido por equações que regem a variação do fluxo em um reservatório. Utilizando o paradigma de fluido neste exemplo, o número de eventos a serem

executados seriam reduzidos para $O(10^3)$, para os mesmo 15 segundos de simulação, um ganho de várias ordens de magnitude em tempo de execução.

Trabalhos na área de simulação de fluido são recentes, sendo que o primeiro data de 1996 e do nosso conhecimento apenas duas ferramentas foram desenvolvidas: NETSIMUL [Liu et al., 1999, Liu et al., 2001] e FluidSim [Marie et al., 2000]. Destes apenas o NETSIMUL está disponível para *download*. A construção de modelos em NETSIMUL é bastante rudimentar e as medidas de interesse obtidas limitam-se a médias. Pela descrição apresentada em [Marie et al., 2000] o FluidSim é também limitado aos objetos de rede previamente construídos. O Simulador de fluido do TANGRAM-II é geral, e fornece ao usuário uma flexibilidade bem maior de atributos, de forma a que modelos mais complexos possam ser construídos. É também rico em termos de estatísticas e traces que podem ser gerados permitindo inclusive a visualização dos fluidos a medida que a simulação ocorre. Além disso é possível construir filas de fluido cujo comportamento pode ser especificado pelo usuário, baseando-se nos componentes do simulador descrito neste trabalho. Em [de Freitas Reinhardt, 2002] são apresentadas comparações entre o NETSIMUL e o simulador de fluido do TANGRAM-II.

O objetivo deste trabalho é apresentar um simulador de redes de computadores baseado na técnica de fluido, evidenciando as técnicas desenvolvidas por nós para lidar com os problemas inerentes ao método. As principais contribuições deste trabalho são: (i) criou-se um paradigma de modelagem de fluido baseado no conceito de recompensas de taxa associado a estados do sistema. Este paradigma é geral o suficiente tornando o simulador mais flexível do que outros encontrados na literatura (e.g. [Marie et al., 2000, Liu et al., 2001]); (ii) vários recursos foram inseridos na ferramenta TANGRAM-II [R. M.M. Leão and E. de Souza e Silva, 2000] de forma a que o simulador pudesse operar de acordo com a técnica de fluido convivendo com a técnica tradicional e herdando as facilidades desta; (iii) diversos elementos de uma rede foram estudados e modelados.

O trabalho está organizado da seguinte forma. A seção 2. apresenta conceitos de recompensa e introduz o uso de recompensas na ferramenta TANGRAM-II. Na seção 3. são descritas as características básicas de um simulador de fluido. A seção 4. apresenta o paradigma utilizado e os recursos implementados para a construção do simulador de fluido. Exemplos que ilustram a aplicabilidade e eficiência do simulador são descritos na seção 5. e as conclusões são apresentadas em 6..

2. Uso de Recompensas na Ferramenta TANGRAM-II

A ferramenta TANGRAM-II possui um poderoso ambiente de modelagem desenvolvido para fins de pesquisa e ensino que possibilita a descrição de sistemas gerais através de uma interface amigável [Carmo et al., 1998, Leão et al., 2000, LAND, 2002].

O paradigma de descrição de modelos é orientado a objetos. De acordo com o paradigma da ferramenta, os modelos são representados por um conjunto de objetos que podem interagir entre si através da troca de mensagens. O estado interno de um objeto é representado por um conjunto de variáveis inteiras e o comportamento de cada um é definido por eventos que possuem uma condição e um conjunto de ações, e também por ações associadas à chegada de mensagens. Após a especificação de um modelo, este pode

ser resolvido através de simulação ou usando um método analítico.

Estados do modelo podem ser associados a recompensas de taxa e transições a recompensas de impulso. O conceito de recompensas, bem como seu uso na ferramenta, é de vital importância para o entendimento deste trabalho, uma vez que os fluidos manipulados pelo simulador são representados através destas recompensas. A seguir apresentaremos algumas definições.

Seja um modelo com espaço de estados finito $S = \{s_i, i = 1, \dots, M\}$. A cada estado $s \in S$ está associada uma taxa de recompensa de um conjunto $R = \{r_{c(s_i)}, i = 1, \dots, M\}$, onde $r_{c(s_i)}$ representa a recompensa ganha por unidade de tempo em que o modelo se encontra no estado s_i e $c(s_i)$ é uma função que mapeia o índice i do estado s_i no índice de recompensa associada a s_i . Seja $IR(t)$ a variável aleatória que representa a recompensa instantânea no tempo t . Temos:

$$CR(t) = \int_0^t IR(\tau) d\tau \quad ACR(t) = \frac{CR(t)}{t} \quad ATC(t) = \frac{\int_0^t CR(\tau) d\tau}{t} \quad (1)$$

onde $CR(t)$ é a variável aleatória denominada de recompensa acumulada durante o intervalo $[0, t]$, $ACR(t)$ é a taxa média da recompensa acumulada em um intervalo de tempo $[0, t]$ e $ATC(t)$ é a média no tempo da recompensa acumulada.

Diversas medidas de interesse podem ser calculadas a partir do conceito de recompensa. Por exemplo, o valor médio de uma fila em $[0, t]$ pode ser obtido, através da criação de uma recompensa de taxa que acumule o tamanho da fila, ou seja, IR igual ao tamanho atual da fila. Neste exemplo a medida ACR seria a variável aleatória desejada.

Tendo em vista os conceitos apresentados acima, sabe-se que para se obter medidas de interesse através do uso de modelos com recompensas de taxa, precisa-se atribuir a cada estado do sistema um valor que indique a quantidade de recompensa que a medida está acumulando por unidade de tempo em que permanecer no estado. No TANGRAM-II as recompensas de taxa estão associadas aos estados dos objetos através de condições. Um objeto pode ter diversas recompensas definidas, de forma que várias medidas possam ser obtidas simultaneamente. Uma recompensa acumulada ($CR(t)$) pode ser limitada superiormente e inferiormente.

3. Características Básicas de um Simulador de Fluido

Os modelos de fluido são uma abstração dos modelos convencionais discretos, onde os dados que trafegam na rede são tratados de forma contínua. Em modelos de redes de computadores que usam esta abordagem, o fluxo de unidades discretas (pacotes) que viaja através dos canais de comunicação e fica armazenado nas filas é substituído por fluidos que se movem de um recipiente para outro. Deste modo classificam-se como eventos apenas as mudanças de taxa na transferência dos fluidos. Assim sendo, em vez de um evento ser gerado cada vez que um pacote é transmitido, é necessário representar apenas os eventos relativos a mudanças das taxas de transmissão. Para certos modelos, como por exemplo aqueles que descrevem redes de alta velocidade, onde os intervalos de mudanças de taxas de transmissão são ordens de grandeza menores que o intervalo entre geração ou término do serviço de pacotes, fica evidente o ganho desta abordagem que necessita tratar

apenas eventos de mudanças de comportamento dos fluxos, em vez da enorme quantidade de eventos que representam a atividade dos pacotes. Cabe ressaltar que os modelos de fluido também possuem estados discretos, além dos contínuos que modelam os fluidos.

Para exemplificar a dinâmica de funcionamento destes modelos, serão apresentados alguns dos principais componentes lógicos que compõem uma rede modelada sob o paradigma de fluido. Obviamente, estes componentes são abstrações dos componentes reais que compõem as redes multimídia atuais.

3.1. Fontes de Tráfego

Estes objetos representam os geradores de tráfego da rede. Por exemplo, tráfego proveniente dos mais variados tipos de servidores, onde cita-se *www*, e-mail, troca de arquivos, até os mais modernos serviços como distribuição de vídeo sob demanda, teleconferências e transmissão de voz sobre IP. As fontes de fluido utilizadas neste trabalho são denominadas MMFS (*Markov-Modulated Fluid Sources*), ou seja, fontes de fluido moduladas por Markov [G. Kesidis and Kwok, 1996, Liu et al., 2001, Liu et al., 1999]. A cada estado associa-se a emissão de fluido a uma taxa constante γ_i , em vez da geração de pacotes individuais.

De forma a ilustrar a diferença entre uma fonte que gera pacotes e uma fonte de fluido é mostrado na Figura 1 o comportamento de duas fontes On-Off: uma que gera pacotes e a outra que é uma fonte de fluido. Na Figura 1.b é possível notar que a quantidade de informação transmitida enquanto a fonte está em *On* salta a intervalos de tempo, no caso da fonte ser de pacotes. Por outro lado, a quantidade de informação transmitida aumenta continuamente para a fonte de fluido.

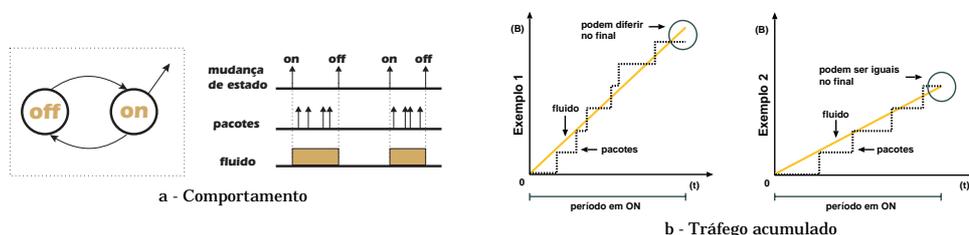


Figura 1: Fontes On-Off de Pacotes e Fluido

3.2. Representação das Filas de uma Rede de Computadores

Numa rede de fluido não existem pacotes chegando e sendo armazenados, e sim fluidos escoando em direção a um nó e sendo acumulados em um reservatório. Em cada reservatório existe uma saída para o líquido armazenado, que escoar, em direção a outro nó, a uma determinada taxa.

O *buffer* representa o espaço disponível para o armazenamento dos fluidos. Ele é representado por sua capacidade B que é uma medida de volume.

As informações armazenadas no buffer podem ser atendidas de diversas maneiras. Duas disciplinas de atendimento são particularmente importantes no ambiente de redes: FIFO e GPS. A seguir será descrito através de equações o comportamento de fluido definido para estas filas.

Seja um nó com uma fila finita de capacidade B , uma taxa de serviço constante c e uma disciplina de atendimento conservadora de trabalho FIFO (*First In First Out*).

Considere $a_i(t) \in [0, \infty)$ a taxa do i -ésimo fluido de entrada no nó e $\vec{a}(t) = \langle a_1(t); a_2(t); \dots; a_N(t) \rangle$ o vetor que representa os fluxos de todos os fluidos de entrada. Desta forma, a taxa total é dada por $A(t) = \sum_{i=1}^N a_i(t)$. Se cada fluxo $a_i(t)$ for descrito por uma função degrau contínua à direita, continua-se a respeitar a condição de que $A(t)$ também seja degrau contínua à direita. De forma semelhante, $\vec{d}(t) = \langle d_1(t); d_2(t); \dots; d_N(t) \rangle$ representa o vetor de saída e $D(t) = \sum_{i=1}^N d_i(t)$ denota a taxa do fluxo total de saída do nó.

Seja τ_n a n -ésima transição do vetor de entrada $\vec{a}(t)$. Desta forma, uma mudança em $\vec{a}(t)$ que ocorra em $t = \tau_n$ levará $Q(\tau_n)/c \geq 0$ unidades de tempo para se propagar até a saída da fila, onde $Q(\tau_n)$ representa o volume de fluido na fila no tempo τ_n . Sendo assim, no tempo $\omega_n = \tau_n + Q(\tau_n)/c$, a proporção dos fluxos de saída deve ser a mesma dos fluxos de entrada no tempo τ_n . Logo, se $A(\tau_n) > 0$, então para cada fluxo i , $d_i(\omega_n)/D(\omega_n) = a_i(\tau_n)/A(\tau_n)$. Cabe ainda salientar que, uma mudança em $\vec{a}(t)$, pode produzir duas transições (não simultâneas) em $\vec{d}(t)$ caso $Q(\tau_n) > 0$, $A(\tau_n) < c$ e a fila esvaziar antes da próxima transição em $\vec{a}(t)$.

O comportamento do vetor de fluxos de saída $\vec{d}(t)$ é descrito pelas seguintes equações [Kumaram and Mitra, 1998]:

$$\vec{d}(\omega_n) = \begin{cases} \frac{D(\omega_n)}{A(\tau_n)} \cdot \vec{a}(\tau_n) & , \text{ se } A(\tau_n) > 0, \\ 0 & , \text{ se } A(\tau_n) = 0 \text{ e } Q(\tau_n) = 0 \text{ (Caso em que } \omega_n = \tau_n), \end{cases} \quad (2)$$

e

$$\vec{d}(t_0(\tau_n)) = \vec{a}(\tau_n), \text{ se } t_0(\tau_n) < \tau_{n+1} \text{ e } Q(\tau_n) > 0. \quad (3)$$

Considere agora a disciplina de atendimento GPS (*Generalized Processor Sharing*) onde a capacidade do canal é dividida entre os fluxos de acordo com parâmetros específicos. A dinâmica da fila é descrita pelas seguintes equações [G. Kesidis and Kwok, 1996]:

$$q_i(T_{n+1}) = q_i(T_n) + [(a_i(T_n) - d_i(T_n))\mathbf{1}(0 \leq q_i(T_n) < B_i) + (a_i(T_n) - d_i(T_n))^- \mathbf{1}(q_i(T_n) = B_i)] \times (T_{n+1} - T_n) \quad (4)$$

$$\frac{d_i}{d_j} = \frac{\phi_i}{\phi_j}, \text{ se } Q_i > 0 \text{ e } Q_j > 0 \quad (5)$$

onde, $i, j \in \{1, 2, 3, \dots, n\}$, representam as filas de cada fluxo do nó GPS e para cada uma delas, tem-se que B_i representa seu tamanho, $a_i(t)$ representa a entrada de fluido, $d_i(t)$ a saída e $q_i(t)$ o volume acumulado no *buffer*. T_n representa a n -ésima transição de $A(t)$, $(y)^- = \min(0, y)$ e $\mathbf{1}$ é a função indicadora.

As disciplinas apresentadas acima, descrevem a maneira pela qual o fluido é servido na fila de um nó de rede. Entretanto para que se possa garantir a equidade de serviço, a robustez, e a qualidade de serviço (QoS), há também, a necessidade de um gerenciamento do espaço da fila, ou em outras palavras, uma política de ocupação de *buffer*. A falta de gerenciamento do espaço pode causar injustiças.

Seja uma fila finita com disciplina de atendimento GPS, de capacidade c que recebe dois fluxos: um amarelo e outro vermelho. Imagina-se um cenário, onde num dado instante a fila esteja cheia, com 50% de fluido de cada cor e o amarelo passe a chegar a uma taxa bastante alta, em relação a capacidade de serviço de sua classe, por exemplo. chegada de $1,25c$ sendo o serviço da classe amarela $0,5c$. Supõem-se também, que a taxa de chegada do vermelho diminua para um valor menor que a capacidade do canal destinada a este, por exemplo $0,3c$. Neste cenário, a taxa do amarelo é maior que a do vermelho, $a_{\text{amarelo}} = 1,25c > a_{\text{vermelho}} = 0,3c$ e a soma das duas taxas de entrada é maior que a capacidade de serviço, $1,55c > c$. Nesta situação, ocorrerão perdas na entrada da fila e apenas parte do fluxo conseguirá lugar no *buffer*. Assim sendo, o fluido amarelo vai entrar em maior quantidade, por causa de sua maior avidéz e tenderá a tomar todo o espaço da fila. O vermelho mesmo chegando a uma taxa menor do que sua banda nominal garantida pela disciplina de serviço GPS, terá perdas e não conseguirá usufruir da banda a que tem direito. Caracteriza-se assim, uma injustiça em relação à qualidade de serviço.

Existem várias formas de resolver estas injustiças, dentre as quais destacam-se algumas bastante conhecidas, descritas em [Lizambri et al., 1999, Mutlu Arpacı and John A. Copeland, 2000]. Estas técnicas são aplicadas para os modelos que trabalham com pacotes, mas podem ser aplicadas ao paradigma de fluidos. Por exemplo, na técnica de Particionamento Completo (CP - Complete Partitioning) a fila é permanentemente particionada em N filas menores, uma para cada fluxo de serviço. Não existe compartilhamento do recurso espaço em fila, entretanto nenhum fluxo é prejudicado por ações dos demais. Esta técnica não faz um bom aproveitamento do espaço total do *buffer*. Entretanto garante justiça, robustez e com isto a qualidade de serviço. No esquema de Compartilhamento Completo (CS - Complete Sharing) a fila é integrada e um pacote que chega é aceito se houver espaço disponível. Nesta disciplina existe o máximo aproveitamento do recurso espaço em fila, no entanto um fluxo pode monopolizar o serviço se for mais ávido.

3.3. Retardo de Propagação e Reguladores de Tráfego

O retardo de propagação é uma característica importante das redes de dados. O efeito do retardo de propagação entre nós é a inserção de um tempo de retardo no fluxo que está por este passando. Em outras palavras, uma mudança de taxa ocorrida em um tempo t na entrada de um canal, só será observada em seu outro extremo no tempo $t + t_p$.

Outro componente de importância na estrutura de uma rede de computadores é o regulador de tráfego (*Leaky Bucket*). Existem 3 parâmetros associados a um FLB (Fluid Leaky Bucket - regulador de tráfego de fluido): (a) A capacidade do *buffer* de fluido - Equivalente ao *buffer* de pacotes em um regulador tradicional. Representada pela variável B_C ; (b) A capacidade do *buffer* de fichas - Representada pela variável B_T ; (c) A taxa de geração de fichas - Representada por r . Além das variáveis acima apresentadas, considera-se $a(t)$ como a taxa de fluido que entra no regulador, $d(t)$ como a taxa de fluido que sai, $q(t)$ a quantidade de fluido armazenada no *buffer* e $f(t)$ a quantidade de fichas armazenada no *buffer* de fichas.

A dinâmica do regulador de tráfego de fluido é dada de forma precisa, pelas seguintes equações:

$$q(T_{n+1}) = [(a(T_n) - r)\mathbf{1}(0 < q(t) < B_C) + (a(T_n) - r)^+\mathbf{1}(q(t) = 0) +$$

$$+(a(T_n) - r)^- \mathbf{1}(q(t) = B_C)] \times (T_{n+1} - T_n) + q(T_n) \quad (6)$$

$$f(T_{n+1}) = [(r - a(T_n)) \mathbf{1}(0 < f(T_n) < B_T) + (r - a(T_n))^+ \mathbf{1}(f(T_n) = 0) + (r - a(T_n))^- \mathbf{1}(f(T_n) = B_T)] \times (T_{n+1} - T_n) + f(T_n) \quad (7)$$

$$d(t) = a(t) \mathbf{1}(q(t) = 0) + r \mathbf{1}(q(t) > 0) \quad (8)$$

onde, T_n representa a n -ésima transição de $a(t)$, $(y)^- = \min(0, y)$, $(y)^+ = \max(0, y)$ e $\mathbf{1}$ é a função indicadora.

4. Simulação de Fluido no Ambiente de Modelagem do TANGRAM-II

Em uma simulação tradicional, eventos a serem executados são colocados em uma lista ordenada por tempo. A simulação progride através da execução dos eventos, onde as ações associadas a estes são realizadas e novos eventos podem ser habilitados. O intervalo de tempo de um evento, desde o momento em que é habilitado até o seu disparo, é determinado por uma distribuição que é especificada no modelo. De forma a ilustrar este processo, a Figura 2.a mostra um modelo bastante simples, com uma fonte *on-off* e uma fila. Neste modelo tem-se eventos de mudança de estado da fonte, eventos de geração de pacotes e eventos de serviço de pacotes na fila. Na simulação de fluido, devem existir novos eventos relacionados com o comportamento dos fluidos. A Figura 2.b mostra um modelo equivalente ao apresentado acima, onde salienta-se a necessidade da especificação de, por exemplo, eventos como o esvaziamento do *buffer* e o enchimento deste por completo. Estes eventos são diferentes dos tradicionais e são definidos por equações matemáticas (apresentadas na seção 3.). Para o escalonamento destes eventos faz-se necessária uma monitoração do nível de fluido do *buffer* e previsões de tempo de disparo para quando o nível do fluido atingir certos valores.

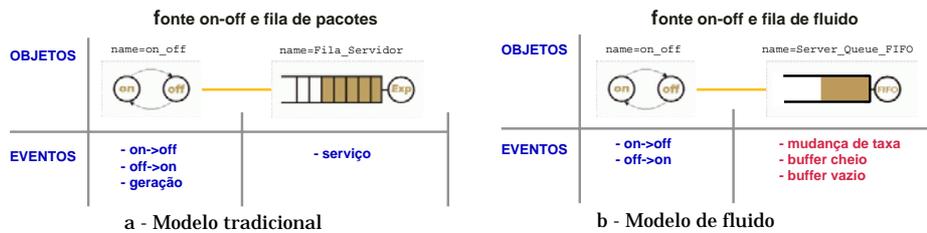


Figura 2: Eventos em um Modelo Tradicional e em um Modelo de Fluido

Objetivou-se neste trabalho, a construção de recursos que possibilitassem a manipulação destes eventos de fluido, de forma genérica e transparente para os usuários. Mesmo sendo perfeitamente possível a implementação de mecanismos que descrevam por completo o comportamento dos objetos de fluido apresentados na seção anterior de forma direta no código do simulador, optou-se pela criação de um paradigma genérico e de recursos que formassem uma base para a implementação de tais objetos. Ou seja, em vez de implementar toda a funcionalidade das equações de fluido (eventos de fluido, disciplinas de atendimento e etc.) para um subconjunto de objetos (por exemplo, diferentes tipos de filas) internamente ao simulador, optou-se pela implementação de um conjunto

genérico de funcionalidades, onde apenas o tratamento de eventos de fluido está embutido no simulador, de forma a prover suporte à criação de modelos de fluido complexos.

O paradigma genérico que serve de base para a construção dos modelos de fluido utiliza a idéia de representar os reservatórios de fluido através de recompensas. Desta forma, recompensas acumulam valores que indicam o volume de fluido armazenado em um determinado recipiente. Através do uso destas recompensas, pode-se modelar os recipientes de fluido, necessários para a construção dos objetos do simulador.

Outro princípio importante para o paradigma é a definição de que as mensagens da ferramenta, devem transportar valores de taxas de fluxos de fluido, e a troca de mensagens, passa a representar mudanças nessas taxas. Assim sendo, uma mensagem enviada de um componente de rede *A* para outro *B*, especifica o quanto de fluido passa a escoar por unidade de tempo de *A* para *B* a partir do recebimento da mensagem. Com o uso das recompensas de taxa acumulada para a descrição dos recipientes de fluido e o uso das mensagens para a notificação de mudanças de taxa, tem-se a base genérica para a implementação de objetos de simulação de fluido.

4.1. Recursos do Simulador

De acordo com os princípios de funcionamento e com a necessidade de representar os componentes de rede apresentados na seção 3., surgiu a necessidade de criação e implementação de diversos recursos. Estes recursos formam a base genérica para a criação de modelos de fluido e implementam diversas funcionalidades dentre as especificadas pelas equações da seção 3.. Em paralelo ao desenvolvimento dos recursos genéricos, foram especificados objetos de fluido, de forma que implementassem as funcionalidades restantes, peculiares aos objetos. A seguir apresentaremos dois dos principais recursos desenvolvidos: *Evento REWARD_REACHED* e *Soma de Recompensas Acumuladas e Instantâneas*.

Evento REWARD_REACHED

De acordo com a teoria dos fluidos, precisa-se saber quando o nível de fluido do reservatório atinge certos limites, representando por exemplo, o esvaziamento e enchimento completo do recipiente. Nestes casos ações devem ser tomadas, como, no momento que um reservatório esvazia, a taxa de saída deve ser alterada, recebendo o valor igual ao da taxa do fluxo de entrada.

Foi definido então um pseudo evento denominado *REWARD_REACHED*. Este evento especial escalona disparos para o instante em que o valor acumulado *CR* de uma recompensa alcança um determinado limite. Foram definidos 2 símbolos, $/\backslash$ e $\backslash/$ que, ao serem usados na condição do evento, definem se o disparo deve ocorrer no momento em que o *CR* da recompensa alcança um dado valor por baixo ($/\backslash$) ou por cima ($\backslash/$).

Os testes são realizados através de sentenças especiais do tipo “*get_cr(recompensa) símbolo limite*” onde *get_cr(recompensa)* representa o valor acumulado de *recompensa*, *símbolo* é um dos caracteres $\backslash/$ ou $/\backslash$ e *limite* representa o valor a ser alcançado. O cálculo

do tempo para o próximo disparo do evento é realizado através da seguinte expressão:

$$\Delta t = \begin{cases} \frac{CR_i - L_i}{|IR_i|} & , \text{ se } CR_i > L_i \text{ e } IR_i < 0 \text{ e símbolo} = \setminus / , \\ \frac{L_i - CR_i}{IR_i} & , \text{ se } CR_i < L_i \text{ e } IR_i > 0 \text{ e símbolo} = / \setminus , \\ \infty & , \text{ demais casos} \end{cases} \quad (9)$$

onde CR_i e IR_i representam, respectivamente, o valor acumulado e o valor instantâneo da recompensa especificada na sentença i , assim como L_i representa o limite desta sentença e *símbolo* é o caracter especial que define o sentido por onde o valor limite será alcançado.

A condição do evento *REWARD_REACHED* suporta a presença de vários testes baseados nos valores acumulados CR das recompensas, através das sentenças especiais “*get_cr(recompensa) símbolo limite*”, além de expressões aritméticas e testes sobre variáveis de estado e constantes, como ocorre nos demais eventos. A condição do evento é avaliada em duas etapas: inicialmente todas as sentenças especiais são consideradas e a condição como um todo é analisada. Se a condição inteira for falsa, nenhum evento é escalonado. Caso seja verdadeira, o sistema analisa uma a uma as sentenças especiais e extrai o valor mínimo de Δt dentre todas elas, escalonando a ocorrência do evento para um tempo futuro, distante Δt do tempo presente de simulação.

Por exemplo seja a condição: $(set_cr(fluido1) \setminus / 0) \parallel (set_cr(fluido2) \setminus / 0)$. Neste caso o valor dos tempos Δt_1 e Δt_2 para que cada fluido atinja o valor 0 é obtido. O valor $\min_i \{ \Delta t_i \}$ é utilizado no disparo do próximo evento.

Soma de Recompensas Acumuladas e Instantâneas

Para monitorar e controlar diversas recompensas de forma integrada foi criado um mecanismo que atua sobre a soma dos valores acumulados e instantâneos das recompensas de taxa. Este mecanismo denominado *rate_reward_sum*, permite a especificação da soma de várias recompensas de taxa. Além da monitoração destes valores, a *rate_reward_sum* controla as recompensas associadas, com relação aos limites dos valores acumulados. Assim como os limites individuais sobre o CR de cada recompensa são respeitados, o limite especificado para a soma das recompensas também é controlado. A complexidade do mecanismo encontra-se justamente neste ponto. Para que a soma das recompensas seja condizente com a situação atual de cada recompensa individual, a partir do momento em que esta soma atingir um limite, faz-se necessário um controle refinado sobre a taxa de acúmulo de cada uma das recompensas associadas.

A Figura 3 ilustra através de exemplos simples, o funcionamento do mecanismo. Caso 1: Considera-se que os fluidos 1 e 2 possuem IR positivos de forma que a fila de fluido tem seu volume aumentado com o tempo. Se nenhuma taxa mudar, *buffer* atingirá B em algum ponto no tempo, o que caracteriza a ocorrência de um evento t_B (*buffer* cheio). A partir deste ponto, onde o limite da soma foi alcançado, as recompensas devem sofrer uma alteração em seu comportamento, de forma que a soma não ultrapasse o limite pré-estabelecido. Neste exemplo, os fluidos 1 e 2 param de crescer, indicando que há perda de fluido para ambos os fluxos. Perda esta que pode ser calculada através da diferença do ângulo de inclinação do CR , entre seu crescimento natural e depois da

atuação do mecanismo.

Caso 2: Entretanto, situações complexas podem aparecer, como no exemplo em que os fluidos A e B têm IR positivos e C possui IR negativo, de forma que a soma seja positiva. Nesta situação, de forma a manter a soma *buffer* constante, os fluxos A e B devem sofrer mudanças de taxa de crescimento, já que C deve continuar com a mesma taxa de decréscimo. Nenhum fluido tem o CR congelado neste exemplo, já que o espaço liberado por C é ocupado pelos demais proporcionalmente a suas taxas de crescimento originais. Nota-se que existe perda para os fluidos A e B .

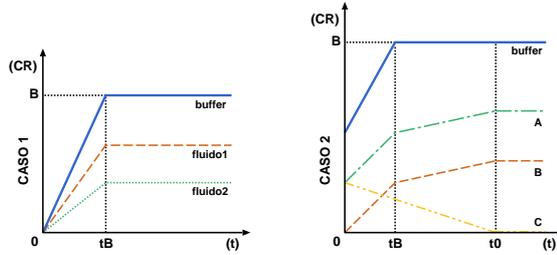


Figura 3: Exemplos do Controle dos CRs Baseados na Soma

Para que o mecanismo seja robusto e abrangente o suficiente de modo a cobrir todos os casos possíveis, foi elaborado um algoritmo que analisa as mudanças de taxas de cada fluido individual e da soma geral. Resumidamente, a soma deve ser atualizada a cada mudança de taxa individual, uma vez que as equações dos fluidos são caracterizadas pelas seguintes equações de primeiro grau:

$$LI_i \leq CR_i \leq LS_i \quad (10)$$

$$CR_i(t) = IR_i(t) \cdot t + CR_i(t_{inicial}) \quad (11)$$

$$CR_{soma}(t) = \sum_{j=1}^i CR_j \quad (12)$$

$$LI_{soma} \leq CR_{soma} \leq LS_{soma} \quad (13)$$

onde, LI_i e LS_i representam o limite inferior e superior do valor acumulado da recompensa i , e CR_i o próprio valor acumulado. A equação 10 indica que o valor acumulado CR de uma recompensa i deve estar sempre dentro dos limites individuais desta. Os símbolos $CR_i(t_{inicial})$ e $IR_i(t)$ representam o valor acumulado no tempo inicial do intervalo e a recompensa instantânea atual da recompensa i no tempo t . A equação 11 é válida para um intervalo de tempo onde não haja mudança de taxa IR_i , neste exemplo $(t_{inicial}, t]$, e descreve o comportamento do valor acumulado da recompensa i . Em 12, CR_{soma} é o valor acumulado resultante da soma das demais recompensas e a equação 13 indica que o valor acumulado da soma deve estar dentro do intervalo (LI_{soma}, LS_{soma}) delimitado por seus limites inferior e superior, respectivamente.

4.2. Problemas de precisão

Os modelos de fluido podem ser pensados como modelos de estado contínuo, e em consequência erros de precisão numérica podem ter efeitos significativos na evolução da simulação. Os computadores digitais só conseguem armazenar números de forma discreta e portanto números reais são aproximados para o número discreto mais próximo do real. Este fato traz problemas para a comparação direta de números reais, operação

necessária para o andamento de um simulador de fluido. Por exemplo, comparação entre valores de reservatórios para disparar um evento. Consideremos o cálculo do tempo de disparo de um evento *REWARD_REACHED* realizado de acordo com a equação 9, sendo a recompensa acumulada em um tempo t de simulação positiva e a taxa de recompensa negativa. Suponha que o valor Δt calculado para que CR alcance 0 seja tal que, por erros de precisão na operação de subtração em 9, a atualização da recompensa não alcance o valor correto 0, isto é, $\Delta t \times IR_i + CR > 0$. Sendo assim, um novo evento *REWARD_REACHED* pode ser escalonado uma vez que CR ainda está acima de zero. A simulação pode entrar em um laço infinito onde sucessivos valores cada vez menores de Δt são obtidos. Portanto técnicas para lidar com esse problema devem ser incluídas em um simulador de fluido.

Uma solução para minorar o problema acima é usar intervalos para a comparação entre dois números, isto é, um número x é considerado igual à outro y se diferir no máximo ϵ do valor de y . Este conceito pode ser estendido a duas dimensões, por exemplo para variáveis no tempo. Para avaliar a igualdade entre $x(t)$ e $y(t)$ no simulador de fluido do TANGRAM-II, leva-se em consideração a variação dentro de um intervalo de tempo δ em torno de t . Além disto, um sistema de aproximações foi desenvolvido de forma a “atrair” valores de variáveis muito próximos a *constantes* do sistema para estas mesmas constantes. Por exemplo aquelas que indicam os limites superior ou inferior de um reservatório de fluido. Quando um valor Δt é calculado, o sistema armazena o limite alcançado e qual recompensa será responsável pelo disparo do evento especial (por exemplo: a recompensa fluido1 alcançando o limite 0). No disparo do evento, a aproximação é acionada e utiliza o valor pré-definido para o limite atingido. Desta forma, tem-se uma melhora na precisão do resultado. A Figura 4 apresenta um exemplo de um modelo onde um evento é escalonado pela condição $(get_cr(z) \setminus B) \text{ — } (get_cr(w) \setminus 0)$ isto é quando o valor acumulado da recompensa z atinge o limite B ou o valor acumulado da recompensa w atinge 0. Neste caso, como os instantes onde ambas as recompensas alcançam os valores limites estão próximos, um único evento é escalonado para t ou t' .

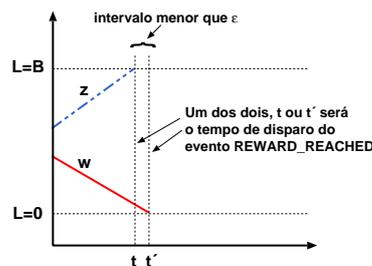


Figura 4: Aproximação na atualização de Recompensas

Além do mecanismo acima, um detector de falta de progressão de simulação monitora o tempo de simulação e inicia uma contagem do número de disparos de eventos especiais (*REWARD_REACHED*), assim como dos demais eventos, toda vez que o tempo de simulação não progride. Como é perfeitamente possível que ocorram diversos eventos em um mesmo tempo de simulação, mas pouco provável que isto ocorra por um número elevado de vezes, se uma das duas contagens ultrapassar um valor determinado, o simulador avisa o usuário sobre um possível problema de laço infinito.

Outro problema relacionado a precisão numérica pode ser exemplificado pela Figura 5. Neste exemplo um evento dispara no tempo e e altera o valor acumulado da recompensa w para um valor muito próximo ao limite B . Além disso, um evento especial (*reward_reached*) é especificado para disparar quando o valor acumulado de w atinge B . Como o valor acumulado para w é praticamente B após a execução de e , é possível que o novo valor mascare o disparo do evento especial, uma vez que este já alcançou seu limite. O simulador monitora todas as recompensas utilizadas em condições de eventos especiais e, durante o processo de atualização das recompensas, é feita uma verificação de alcance de um limite especificado por uma condição. Neste caso, e sabendo-se que o evento está escalonado num tempo muito próximo ao tempo atual, o evento em questão é acionado.

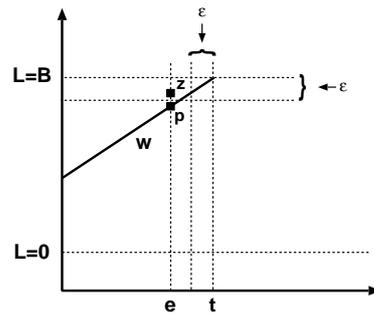


Figura 5: Possível falha no disparo de eventos especiais.

5. Exemplos

Nesta seção iremos apresentar dois exemplos. O primeiro ilustra o ganho computacional que pode ser obtido com a técnica de fluido e mostra também que a diferença entre os resultados da simulação tradicional e de fluido é bastante pequena. O segundo é um exemplo mais complexo que visa mostrar o tipo de medidas de interesse que podem ser obtidas. Outros exemplos mais complexos podem ser encontrados em [de Freitas Reinhardt, 2002].

Foram criados na ferramenta TANGRAM-II dois modelos semelhantes, um de fluido e outro tradicional de pacotes. A Figura 6 mostra o modelo. Os parâmetros esco-

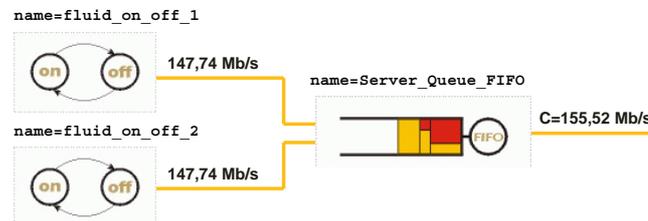


Figura 6: Modelo de um Roteador.

lhidos para a simulação de fluido foram: $\lambda = 1$, $\mu = 1$ e $\gamma = 348.452$ pacotes/segundo (on-off/off-on/taxa de transmissão) e uma fila servidor FIFO de parâmetros: $B = 18.000$ pacotes, $\phi_1 = \phi_2 = 0,5$ e $C = 366.790$ pacotes/segundo onde a capacidade de serviço

corresponde a uma utilização de 0,95. O modelo de pacotes, também têm mudanças de estado das fontes regidas por variáveis exponenciais, no entanto o tempo entre a geração de pacotes é determinístico.

A Tabela 1 mostra os resultados obtidos de 10 execuções de 50.000 segundos cada.

Modelo	Média da fila (x1000 pacotes)	Tempo de Execução (segundos)
Pacotes	6,9903 ± 0,03051	41.618
Fluido	6,8430 ± 0,01757	728

Tabela 1: Roteador - Modelo de pacotes X fluido

Neste exemplo os resultados diferem de 2,7% e o simulador de fluido é aproximadamente duas ordens de grandeza mais rápido que o simulador tradicional, o que demonstra a ótima relação custo benefício do método.

A Figura 7 apresenta o modelo de um roteador alimentado por duas fontes policia- das por dois reguladores de tráfego. A disciplina da fila do roteador é GPS. Foi realizada

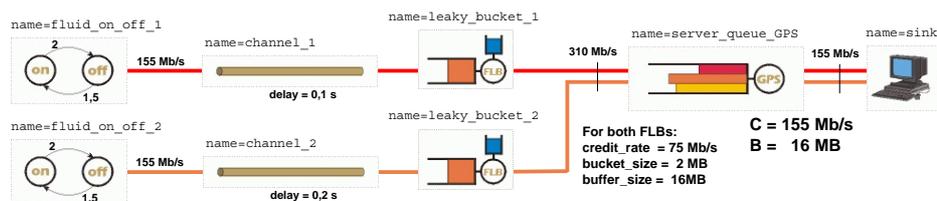


Figura 7: Roteador com Tráfego Regulado.

uma simulação de 30.000 segundos com 5 execuções e intervalo de confiança de 95%, cujos resultados são apresentados nas tabelas 2 e 3.

Medida de Interesse	Valor (intervalo)	Unidade
Média de fluido1 na Fila	15,11 ± 0,36	Mb
Média de fluido2 na Fila	16,35 ± 0,53	Mb
Média da Fila	31,45 ± 0,80	Mb
Montante de Chegada de fluido1	2.189,32 ± 10,49	Tb
Montante de Chegada de fluido2	2.232,90 ± 6,35	Tb
Montante de Chegada	4.422,22 ± 16,80	Tb
Montante de Perda do fluido1	5,54 ± 0,59	Tb
Montante de Perda do fluido2	6,77 ± 0,51	Tb
Montante de Perda	12,31 ± 1,31	Tb
Utilização	97,40 ± 0,24	%
Medida Derivada	Valor	Unidade
Porcentagem de Perda	0,2784	%
Retardo Médio do fluido1	0,2175	μ seg
Retardo Médio do fluido2	0,2255	μ seg
Retardo Médio na fila	0,2215	μ seg

Tabela 2: Medidas de Interesse do server_queue_GPS

Medida de Interesse	Valor (intervalo)	Unidade
Média da Fila	46,77 ± 0,68	Mb
Montante de Chegada	2.179,51 ± 6,36	Tb
Montante de Perda	0,00 ± 0,00	

Tabela 3: Medidas de Interesse do leaky_bucket_1

Além dos resultados apresentados acima, arquivos de traces podem ser gerados automaticamente contendo o valor instantâneo e acumulado de cada recompensa. A Figura 8.a mostra um gráfico gerado a partir de *traces* contendo valores de recompensas instantâneas do regulador de tráfego. Estas recompensas representam a taxa de chegada e saída do regulador, o estado do balde de fichas e da fila de entrada. Já a Figura 8.b mostra o valor da recompensa acumulada que representa as chegadas e saídas do regulador com os respectivos limites inferior e superior. Através deste gráfico pode-se observar a moldagem do tráfego. Os gráficos podem ser gerados a medida que o simulador executa.

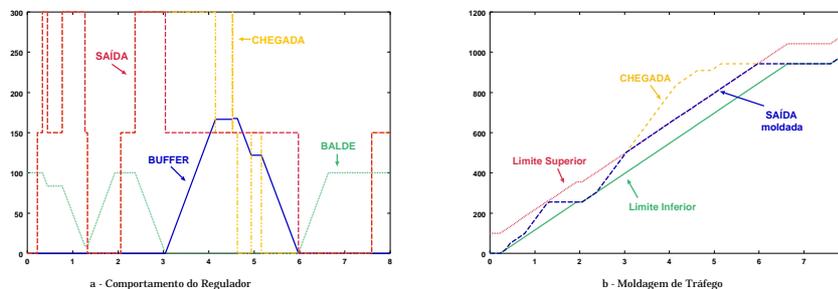


Figura 8: Regulador de Tráfego.

6. Conclusão

Neste trabalho criou-se um paradigma de modelagem de fluido baseado no conceito de recompensas e as equações que regem os fluidos estão embutidas no conceito de recompensa acumulada. Dentro deste paradigma e focalizado na área de redes de computadores, um simulador de fluido foi idealizado, construído e validado. Foi elaborado um conjunto genérico de funcionalidades tomando-se como alicerce o simulador da ferramenta TANGRAM-II. Desta forma, não há diferenças a nível de usuário entre o simulador tradicional e o de fluido. A nível de usuário apenas um conjunto de facilidade de tratamento de recompensas foi adicionado. Entretanto, o paradigma elaborado permite a simulação de fluido de modelos de redes e ainda fornece ao usuário a possibilidade de construção de novos modelos de forma simples e bem mais abrangente em comparação a outras ferramentas de fluido na literatura.

O simulador herdou todas as características e poder de modelagem e apresentação de resultados do simulador do TANGRAM-II, e os novos recursos implementados são genéricos o suficiente para suportar a modelagem de quaisquer sistemas que se enquadrem à técnica de fluido. Além disso, objetos específicos foram criados para serem usados como elementos básicos na montagem de diversos tipos de modelos de redes. O simulador

também conta com recursos de apresentação de resultados onde, além de medidas sobre os pontos de interesse, há opções de geração de *traces* e apresentação gráfica de resultados durante a simulação, mostrando o comportamento dos objetos do modelo.

Referências

- Carmo, R., de Carvalho, L., de Souza e Silva, E., Diniz, M., and Muntz, R. (1998). Performance/Availability Modeling with the TANGRAM-II Modeling Environment. *Performance Evaluation*, 33:45–65.
- de Freitas Reinhardt, K. (2002). Um Ambiente de Simulação de Fluido com Aplicações a Redes Multimídia. Master's thesis, COPPE/UFRJ.
- Figueiredo, D. R. (1999). O Módulo de Simulação da Ferramenta TANGRAM-II: Suporte para Medidas com Recompensas, Recursos de Eventos Raros e Aplicações a Modelos de Redes Multimídia. Master's thesis, COPPE/UFRJ.
- G. Kesidis, A. Singh, D. C. and Kwok, W. W. (1996). Feasibility of Fluid Event-driven Simulation for ATM Networks. In *IEEE GLOBECOM*.
- Guo, Y., Gong, W., and Towsley, D. F. (2000). Time-stepped hybrid simulation (TSHS) for large scale networks. In *INFOCOM (2)*, pages 441–450.
- Kesidis, G. and Singh, A. (1995). An overview of cell-level ATM network simulation. In *High Performance Computing Systems Conf.*, pages 202–214.
- Kumaram, K. and Mitra, D. (1998). Performance and Fluid Simulations of a Novel Shared Buffer Management System. In *Proc. IEEE INFOCOM*.
- LAND (2002). *TANGRAM-II user's manual*. UFRJ.
- Leão, R., de Souza e Silva, E., and de Lucena, S. (2000). A Set of Tools for Traffic Modeling, Analysis and Experimentation. In *TOOLS'2000*, pages 40–55.
- Liu, B., Guo, Y., Kurose, J., Towsley, D., and Figueiredo, D. R. (2001). A Study of Networks Simulation Efficiency: Fluid Simulation vs. Packet-level Simulation. In *INFOCOM*.
- Liu, B., Guo, Y., Kurose, J., Towsley, D., and Gong, W. (1999). Fluid Simulation of Large Scale Networks: Issues and Tradeoffs . In *PDPTA '99*.
- Lizambri, T., Duran, F., and Wakid, S. (1999). Priority Scheduling and Buffer Management for ATM Traffic Shaping. In *Proceedings of 7th IEEE Workshop on Future Trends of Distributed Computing Systems - FTDCS '99*.
- Marie, R., Incera, J., Ross, D., and Rubino, G. (2000). FluidSim: A Tool to Simulate Fluid Models of High-Speed Network. In *Proceedings of TOOLS 2000*.
- Mutlu Arpacı and John A. Copeland (2000). Buffer Management for Shared-Memory ATM Switches. *IEEE Communications Surveys & Tutorials*.
- R. M.M. Leão and E. de Souza e Silva (2000). The TANGRAM-II Environment. In *TOOLS'2000*, pages 366–369.