

Novas evoluções integradas à ferramenta Tangram-II v3.1 *

Antonio Rocha, Guilherme Jaime, Fabrício Murai, Bruno Alves
Daniel R. Figueiredo, Rosa M. M. Leão, Edmundo A. de Souza e Silva

¹COPPE/Prog. de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro

{arocha, gdjaime, fabricio, balves, daniel, rosam, edmundo}@land.ufrj.br

Resumo. Neste artigo apresentamos uma breve descrição do Tangram-II, um ambiente de análise, modelagem e medições em redes de computadores, ressaltando as funcionalidades recentemente integradas à versão 3.1 da ferramenta. São também descritos alguns dos exemplos que serão usados na “Sessão de Exibição de Ferramentas do SBRC” para demonstrar todo o potencial e flexibilidade do Tangram-II.

Abstract. In this paper we briefly describe Tangram-II, an integrated environment for computer network analysis, modeling and measurement, pointing out the new features implemented for version 3.1 of the tool. It is also described a set of examples which will be used during the “SBRC Tools Exhibition Session” to show all the power and flexibility of the tool.

1. Introdução

As tarefas de especificar modelos, resolvê-los analiticamente ou através de simulação, e executar experimentos de medições em redes são passos essenciais no processo de desenvolvimento de alguns sistemas de comunicação. Nas últimas décadas, diversas ferramentas de modelagem e avaliação de desempenho foram apresentadas na literatura (mais detalhes em [de Souza e Silva et al. 2006]). Além dos simuladores, um enorme esforço vem também sendo dedicado para o desenvolvimento de técnicas e ferramentas de medições em redes (ver [Paxson 1998, Rocha et al. 2004] e suas referências). Os resultados de medições de experimentos reais coletados na Internet podem ser úteis aos analistas para extrair parâmetros que serão utilizados no desenvolvimento de modelos mais precisos.

Um ambiente integrado, que permita ao analista especificar modelos precisos, estimar as métricas de interesse desejadas, e obter as medidas de desempenho da rede através de experimentos, é certamente de grande utilidade. Exatamente com esse propósito, a ferramenta Tangram-II [Site Tangram-II] vem sendo desenvolvida pelo LAND/UFRJ¹ há mais de uma década. Nesse período, diferentes versões da ferramenta já foram lançadas, sempre contendo o estado da arte dos algoritmos e métodos de soluções analíticas e simulação, além das técnicas de medições em redes. A seguir iremos apresentar uma descrição geral do ambiente Tangram-II e, na Seção 3, detalharemos as novas evoluções integradas à versão 3.1 da ferramenta.

*Este trabalho é parcialmente financiado pela CAPES, CNPq e Faperj.

¹Laboratory for modeling, analysis and development of networks and computer systems / Universidade Federal do Rio de Janeiro (<http://www.land.ufrj.br>)

2. Breve descrição da ferramenta Tangram-II

A ferramenta Tangram-II foi desenvolvida com o propósito de dar o suporte necessário à pesquisa e a educação em avaliação de desempenho de sistemas de computação e comunicação. O ambiente é formado por módulos de diferentes finalidades, porém, a forte integração entre esses módulos destaca-se como mais um dos diferenciais da ferramenta em relação às existentes na literatura. A visão geral dos módulos, juntamente com a interface inicial da ferramenta, é ilustrada na Figura 1. Uma breve descrição dos principais módulos será apresentada a seguir.

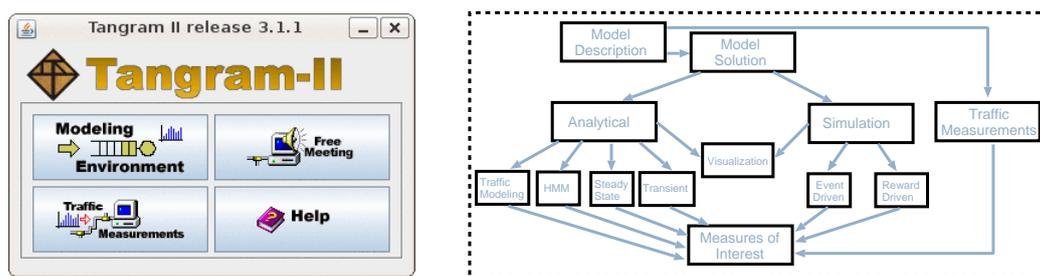


Figura 1. Módulos do ambiente Tangram-II versão 3.1.

2.1. Especificação de modelos

O paradigma definido no Tangram-II para o processo de especificação de modelos é baseado em objetos. Cada objeto contém uma série de atributos (declaração, inicialização, eventos, mensagens e recompensas) e pode estar associado a um conjunto de variáveis de estado. Os valores dessas variáveis em um determinado instante de tempo definem o estado do objeto.

Os “eventos” e “mensagens” definem o comportamento associado a um objeto. Eventos são disparados segundo uma distribuição definida na especificação. Objetos interagem através de trocas de mensagens. Ações podem ser especificadas no modelo para cada evento disparado ou mensagem recebida.

A linguagem de descrição dos modelos é similar ao C. A especificação dos modelos é feita usando o TGIF *Tangram Graphical Interface Facility* [Site TGIF]. A Figura 2(direita) ilustra o exemplo de um modelo simples especificado no Tangram-II para representar um sistema de filas $M/M/1/K$. Após a especificação, o usuário pode optar (seguindo as restrições dos métodos de solução) por resolver o modelo analiticamente ou via simulação. A Figura 2(esquerda) mostra a interface gráfica do ambiente de modelagem do Tangram-II, que guia o usuário pelas etapas a serem seguidas desde a criação do modelo até a obtenção das medidas de interesse.

Juntamente com a ferramenta é disponibilizada uma rica biblioteca de objetos e exemplos de modelos que podem ser utilizados como ponto de partida para os usuários.

2.2. Soluções analíticas

Se o modelo especificado for Markoviano, ou pertencer a uma classe específica de modelos não Markovianos, poderá ser resolvido analiticamente através de um dos métodos de solução implementados na ferramenta. Para soluções em estado estacionário, métodos

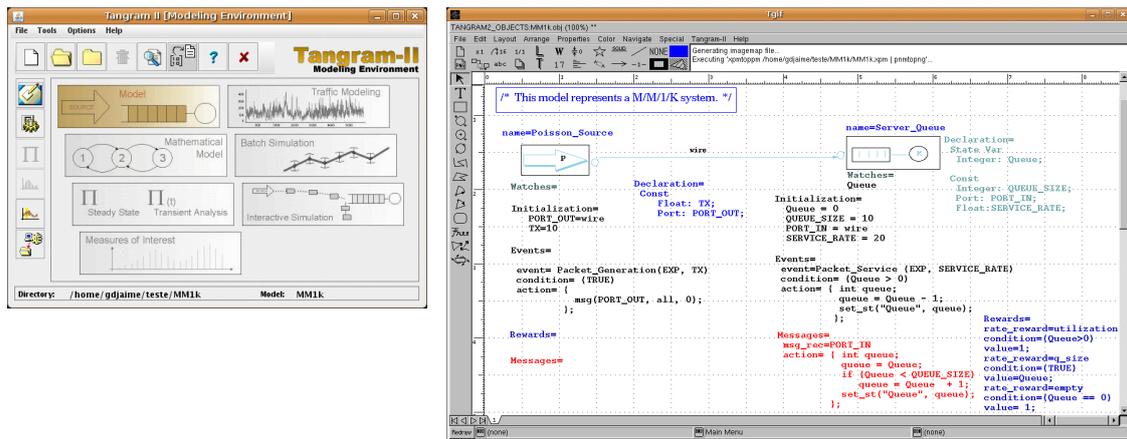


Figura 2. Interfaces gráficas do Tangram-II: (esquerda) Ambiente de modelagem; (direita) Interface TGIF utilizado para especificação do modelo

iterativos (Power, Gauss-Siedel, Jacobi e SOR) e exatos (GTH, GTH Block) estão implementados, além do método de solução não Markoviana. Uma série de algoritmos para solução em estado transiente também estão disponíveis no Tangram-II (ver [de Souza e Silva et al. 2006]).

2.3. Simulação

Quando o modelo especificado não satisfaz os requisitos para ser resolvido analiticamente, o analista pode utilizar o simulador para resolver o modelo. O Tangram-II oferece dois tipos diferentes de simulação, uma baseada em eventos e outra em fluidos. Para a simulação baseada em eventos, além da técnica tradicional, o Tangram-II implementa a técnica *RESTART*, apropriada para modelos com eventos raros. No simulador de fluido, ao invés de usar eventos para modelar cada característica de um sistema (ex: geração de pacotes, serviço de pacotes), o tráfego é definido como um fluxo contínuo e eventos são criados apenas para tratar eventos de mudança de taxa (ex: esvaziamento ou enchimento).

O módulo de simulação tem ainda como importante característica a alta integração com os outros módulos do conjunto de ferramentas Tangram-II. O usuário pode, por exemplo, utilizar *traces* coletados pelos ambientes de *Medição de Tráfego* para usar como parte do modelo de simulação. (Veremos um exemplo na Seção 4.)

2.4. Medidas de interesse

As medidas de interesse computadas dependem da abordagem utilizada na especificação dos modelos. As soluções analíticas permitem extrair medidas tanto das variáveis de estado quanto das recompensas. Já nos modelos de simulação, as medidas podem ser computadas apenas utilizando recompensas.

Quando são utilizadas variáveis de estado para as soluções em estado estacionário e transientes, o Tangram-II calcula a PMF, a distribuição marginal e a distribuição condicional. Quando usadas recompensas de taxa, que acumulam um valor especificado sempre que uma dada condição é satisfeita, e recompensas de impulso, que acumulam um valor definido sempre que há disparo de um dado evento, são computadas estatísticas como a média, variância, e distribuição da recompensa acumulada. Além disso, a ferramenta pos-

sibilita computar os intervalos de confiança das medidas de interesse geradas, baseado no número de rodadas e no nível de confiança especificado pelo usuário.

2.5. Medição de tráfego

O Tangram-II possui um módulo de experimentos de medições ativas em redes, que fornece ao usuário suporte a diferentes tipos de geração de tráfego e algoritmos para cálculo de diversas estatísticas. O gerador de tráfego existente possibilita que pacotes sejam enviados seguindo diferentes padrões (CBR, Markoviano, Pares de Pacotes ou a partir de um *trace*) e em diferentes sentidos entre as máquinas na rede (“em um sentido”, “nos dois sentidos” e “ida-e-volta”). Algumas estatísticas básicas podem ser estimadas diretamente dos dados coletados, tais como: *jitter*, distribuição de perda e sucesso. Recentemente, algoritmos sofisticados foram adicionados à ferramenta e possibilitam estimar outras métricas mais complexas. Detalhes sobre essas métricas são descritos na Seção 3.3.

3. Novas Funcionalidades do Tangram-II versão 3.1

3.1. Novas facilidades na linguagem de descrição de modelos

Nos últimos anos houve um forte crescimento no número de trabalhos que envolvem modelos significativamente detalhados. A exemplo disso estão os modelos de redes sem fio [Marca et al.] e mobilidade de usuários [Jaime et al. 2008]. Modelos desta natureza tipicamente requerem o uso de funções matemáticas como potência, logaritmos e funções trigonométricas. Por isso tornou-se necessária a adição de novas funcionalidades à linguagem da ferramenta Tangram-II na versão 3.1. A linguagem de descrição de modelos agora suporta todas as funções matemáticas descritas no cabeçalho *math.h* do compilador C, tais como *fabs*, *floor*, *ceil*, *sin*, *cos* e *tan*, entre outros.

Duas outras funções foram também adicionadas à linguagem de descrição de modelos: *get_random* e *get_simul_time*. A função *get_random* permite que o modelo especificado gere uma amostra pseudo-aleatória, entre $[0, 1)$, segundo a distribuição uniforme. A partir de uma v.a. uniforme é possível gerar v.a. para diversas outras distribuições [Trivedi 1982]. A criação da função *get_simul_time* tem como objetivo facilitar a depuração de modelos de simulação. A função retorna o tempo de simulação durante a execução.

Por fim, uma última funcionalidade implementada também para auxiliar no processo de depuração dos modelos é a definição das sementes de geração de números aleatórios. Com isso, o usuário pode optar por usar uma mesma semente e fazer com que diferentes experimentos de simulação percorram sempre o mesmo caminho amostral, o que facilita a investigação de comportamentos indesejados no modelo.

3.2. MTK

Outra importante funcionalidade incorporada ao Tangram-II é o Modeling Tool Kit (MTK). O MTK é um *framework* recém-integrado ao módulo de simulação e que permite ao usuário implementar algoritmos complexos, que são disparados com a ocorrência de um evento ou o recebimento de uma mensagem. Esses algoritmos são implementados sob a forma de plugins escritos em C++, conferindo grande liberdade na sua descrição e, conseqüentemente, na especificação dos modelos. É possível, por exemplo, que esses plugins

invocarem métodos de soluções analíticas do Tangram-II, resultando em modelos híbridos de simulação.

Os plugins do MTK são instanciados como objetos na descrição do modelo feita pelo usuário. Durante a execução de uma ação, esses objetos podem ser criados e destruídos, os valores de seus atributos podem ser alterados ou recuperados, e podem também fazer chamadas aos seus métodos. Em todos esses casos, o Tangram-II aguarda até que o plugin execute as respectivas instruções e então retoma o controle da execução. Um dos plugins que acompanham a ferramenta implementa os algoritmos que resolvem os 3 problemas básicos de Cadeias de Markov Ocultas (HMM), tanto para o caso do HMM clássico quanto para o caso do HMM hierárquico [Rabiner 1989].

3.3. Medições de Tráfego

Recentemente, sofisticadas técnicas de medição foram adicionadas ao Tangram-II. Um dos métodos disponibilizados na ferramenta é um conjunto de algoritmos que possibilitam estimar o atraso em um único sentido entre duas máquinas distintas, tratando problemas referentes à falta de sincronia entre os relógios [Rocha et al. 2004]. Problemas como *skew* e *offset* são tratados pelos algoritmos implementados. A Figura 3(a) apresenta a distribuição do atraso em um sentido obtido a partir de um experimento, comparado com algumas distribuições conhecidas. Dessa forma o analista pode assumir a distribuição mais adequada para utilizar no modelo (no caso da Figura 3(a), a distribuição exponencial) ou utilizar o próprio *trace* para alimentar o modelo. O Tangram-II oferece três tipos de

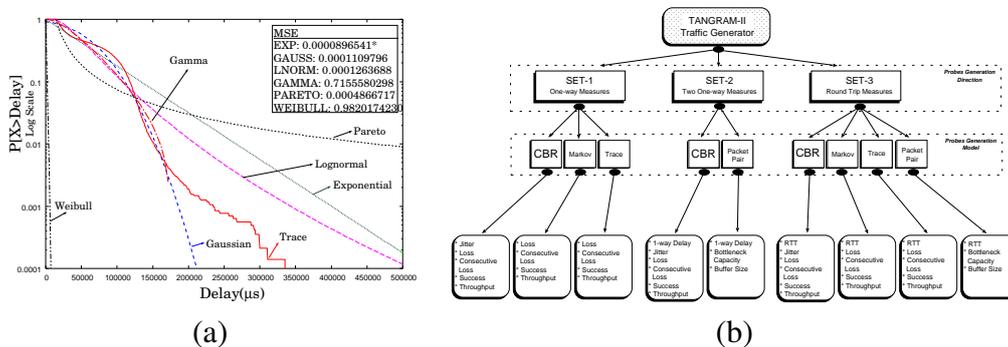


Figura 3. Módulo de medição de tráfego: (a) distribuição do atraso em um sentido computado por um experimento; (b) visão geral e métricas disponíveis

geração de tráfego distintas: *one-way*, *two-way* e *round-trip*. Para cada tipo de geração, um conjunto de medidas pode ser estimado, a depender das restrições dos algoritmos para computar essas métricas. A Figura 3(b) apresenta uma visão geral sobre os tipos de medição e as métricas possíveis de serem computadas na versão atual da ferramenta.

4. Exemplos e demonstração da ferramenta

Apresentaremos dois dos exemplos que serão utilizadas na “Sessão de Exibição de Ferramentas do SBRC”. O objetivo é demonstrar a usabilidade do Tangram-II e a integração entre os seus módulos.

4.1. Protocolo de sinalização *soft-state*

Neste primeiro exemplo modelamos um protocolo de sinalização do tipo *soft-state* a fim de obter algumas medidas de desempenho. Esse tipo de sinalização foi estudado recente-

mente e comparado com o tipo *hard-state* [Ji et al. 2007].

A sinalização *soft-state* funciona da seguinte forma: O instalador envia uma mensagem de instalação ao Mantenedor de estado para nele instalar um estado. O canal de comunicação não garante a entrega de mensagens. Em seguida, o instalador envia mensagens periódicas de atualização, indicando à outra entidade que o estado deve ser mantido. Quando o estado não for mais necessário, o instalador simplesmente pára de enviar atualizações. O Mantenedor instala o estado ao receber uma mensagem de instalação ou de atualização. Se o Mantenedor ficar durante um determinado período sem receber quaisquer dessas mensagens, o estado é desinstalado por *timeout*.

A modelagem da sinalização *soft-state* no Tangram-II pode ser feita com o uso de três objetos: o instalador, o canal e o Mantenedor. O instalador alterna entre períodos em que o estado deve e não deve ser instalado no Mantenedor. Ele envia mensagens de instalação e, periodicamente, mensagens de atualização. O canal de comunicação, ao receber uma dessas mensagens, as guarda por um certo tempo e depois as entrega para o Mantenedor probabilisticamente. Assumimos que a taxa do evento de entrega de mensagens é muito maior do que as de instalação e de atualização do estado, ou seja, é muito pouco provável que uma mensagem chegue ao canal antes que a anterior seja entregue. Por fim, o Mantenedor instala um estado após receber uma mensagem. Caso o Mantenedor fique um determinado tempo sem receber nenhuma mensagem, o estado é desinstalado. Esse objeto possui uma variável para indicar se o estado está ou não instalado. O objeto que descreve o instalador de estado na linguagem do Tangram-II é mostrado na

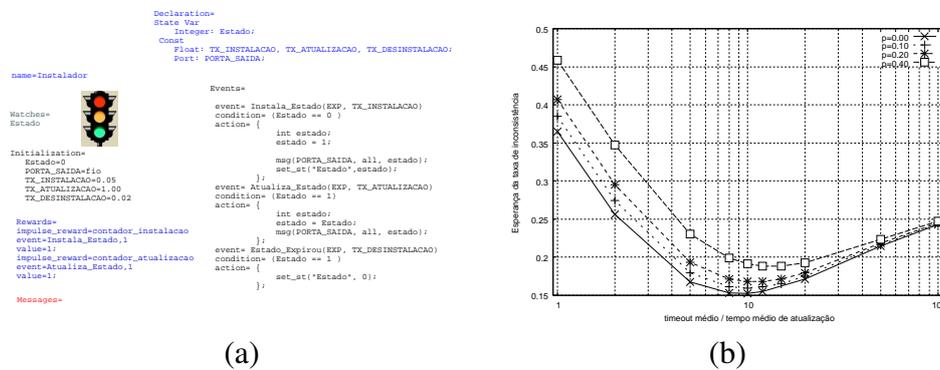


Figura 4. Modelo *soft-state*: (a) representação do Instalador de Estado; (b) esperança da taxa de inconsistência em função do timeout médio

Figura 4(a). Ele possui uma única variável, chamada “Estado”, inicialmente com valor 0, indicando que o estado ainda não deve ser instalado. O evento “Instala_Estado” faz com que essa variável assuma o valor 1 e envia uma mensagem através da porta de saída. O valor 1 indica que o estado ‘1’ deve ser instalado no Mantenedor. Poderiam haver outros estados, mas vamos assumir por simplicidade que só existe um estado. É possível definir condições sob as quais um evento está ativo. O evento “Estado_Expirou”, por exemplo, só está ativo quando há um estado instalado. O objeto contém também a definição de recompensas, que permitem ao usuário obter a fração do tempo em que o estado devia ser instalado e o número de mensagens de atualização enviadas em uma rodada da simulação.

Iremos assumir que, para todos os eventos, o tempo entre ocorrências de um evento segue uma distribuição exponencial (indicado na Figura 4(a) por “EXP”). Assim,

é possível resolver este modelo analiticamente, gerando uma Cadeia de Markov e encontrando as probabilidades dos estados estacionários através do Tangram-II. A partir dessa solução, computamos a taxa de inconsistência esperada avaliando a probabilidade de estar em um dos estados em que as variáveis do Instalador e do Mantenedor são diferentes.

Calculamos os valores da taxa de inconsistência esperada em função do tempo médio de timeout (em múltiplos do tempo médio entre mensagens de atualização). Os resultados obtidos são mostrados na Figura 4(b), onde cada curva representa uma taxa de perda diferente do canal. Quando o timeout é muito próximo do tempo de atualização, a taxa de inconsistência esperada é alta. Conforme o timeout aumenta, a taxa de inconsistência esperada diminui, pois a remoção de estado desnecessária do Mantenedor ocorre com menos frequência. No entanto, a partir de um determinado ponto, a taxa volta a subir, pois depois que o estado expira, o Mantenedor leva muito tempo para removê-lo quando o timeout é grande. A esperança da taxa de inconsistência é maior quanto maior for a taxa de perda do canal, mas a diferença entre as curvas vai caindo com o aumento do timeout.

4.2. Simulação de algoritmos para previsão de perda

O segundo exemplo (ilustrado na Figura 5(a)) é um modelo de simulação baseado em eventos que faz uso do MTK e é alimentado por “traces” coletados de experimentos reais executados na Internet. O objetivo deste modelo é testar os algoritmos de previsão de taxa de perda de pacotes implementados em uma ferramenta VoIP. O objeto “Sender” gera

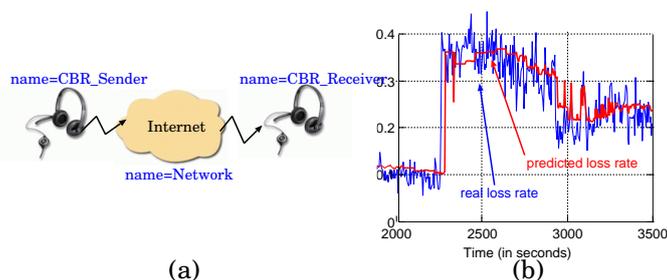


Figura 5. Exemplo de um modelo de simulação utilizando MTK

pacotes a uma taxa constante emulando uma aplicação de voz sobre IP com codificação *PCM*. Os pacotes gerados pelo objeto “Sender” são passados para o objeto “*Network*”. “*Network*” é um objeto MTK que utiliza um arquivo de *trace* coletado previamente pelo módulo de medição de tráfego do Tangram-II. O *trace* consiste de uma sequência de 0’s e 1’s indicando, respectivamente, se um pacote chegou com sucesso ou foi perdido na rede durante o experimento. Cada pacote recebido pelo “*Network*” será encaminhado para o objeto “Receiver” se a próxima linha do *trace* for 1.

O “Receiver” recebe os pacotes do objeto “*Network*” e utiliza um *HMM* como base de um algoritmo de previsão de perda. O algoritmo requer o treinamento do *HMM* hierárquico, baseado em perdas passadas, para inferir o estado atual, e estimar a taxa de perda para uma janela no futuro (ver detalhes em [Filho et al. 2006]). Para realizar estas tarefas, carregamos no “Receiver” um plugin do MTK que implementa um *HMM*. A Figura 5(b) mostra a taxa de perda real obtida nos experimentos e a taxa de perda estimada pelo algoritmo durante a simulação.

5. Considerações finais

O conjunto de ferramentas Tangram-II foi desenvolvido com o objetivo de disponibilizar um ambiente que abranja uma ampla gama de questões envolvidas no processo de modelagem e, mesmo assim, seja fácil de usar quando comparado às ferramentas de mesma categoria. Neste artigo foi apresentada uma breve descrição do estado atual da ferramenta Tangram-II, destacando as funcionalidades integradas recentemente à versão 3.1. Mais informações sobre *download*, requisitos de sistema, tutoriais e outros exemplos de uso, bem como sobre a licença GPL sob a qual o Tangram-II é distribuído estão disponíveis em [Site Tangram-II].

A ferramenta Tangram-II tem sido usada por diversos grupos de pesquisa do Brasil e do exterior (ex: AT&T, SIEMENS). Na área de educação, ela já vem há algum tempo sendo utilizada nos cursos de graduação e pós-graduação da UFRJ, e recentemente pela University of Massachussets-Amherst, em um dos cursos de pós-graduação.

Referências

- de Souza e Silva, E., Leão, R. M. M., Muntz, R. R., da Silva, A. P. C., de A. Rocha, A. A., Duarte, F. P., Filho, F. J. S., and Jaime, G. D. G. (2006). Modeling, Analysis, Measurement and Experimentation with the Tangram-II Integrated Environment. In *ValueTools*.
- Filho, F. J. S., Watanabe, E. H., and de Souza e Silva, E. A. (2006). Adaptive forward error correction for interactive streaming over the Internet. In *Proceedings of the IEEE Globecom*.
- Jaime, G. D., Leão, R. M. M., de Souza e Silva, E., and de Marca, J. R. B. (2008). The Effect of Mobility on the User-Level Fairness of a 3G Wireless Technology (EV-DO). In *IEEE Wireless Communications & Networking Conference*.
- Ji, P., Ge, Z., Kurose, J., and Towsley, D. (2007). A comparison of hard-state and soft-state signaling protocols. *IEEE/ACM TON*, 15:281–294.
- Marca, J. R. B., Duarte, F. P., Jaime, G. D. G., de Souza e Silva, E., and R. M. M. Leão, P. A. B. Modelagem e Avaliação da Tecnologia CDMA EV-DO para aplicação no SBTvD. In *24o SBRC, 2006*.
- Paxson, V. (1998). On calibrating measurements of packet transit times. In *ACM Sigmetrics 1998*.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–285.
- Rocha, A. A. A., Leão, R. M. M., and de Souza e Silva, E. (2004). Metodologia para Estimar o Atraso em um Sentido e Experimentos na Internet. In *XXII SBRC*.
- Site Tangram-II. <http://www.land.ufrj.br/tools/tangram2/tangram2.html>.
- Site TGIF. <http://bourbon.usc.edu/tgif>.
- Trivedi, K. S. (1982). *Probability and statistics with reliability, queuing, and computer science applications*. Prentice-hall Englewood Cliffs, NJ.