

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Pós-Graduação em Engenharia Elétrica e Informática Industrial

TESE

apresentada à UTFPR
para obtenção do título de

DOUTOR EM CIÊNCIAS

por

CARLOS MARCELO PEDROSO

**DESENVOLVIMENTO DE MODELO DE TRÁFEGO E MÉTODO
PARA MELHORIA DE DESEMPENHO DE SERVIDORES WEB**

Banca Examinadora:

Presidente e orientador:

PROF^a. DR^a. KEIKO V. ONO FONSECA UTFPR

Examinadores:

PROF. DR. DALTON S. ARANTES	UNICAMP
PROF. DR. EMILIO C. G. WILLE	UTFPR
PROF. DR. LAU CHEUK LUNG	PUCPR
PROF. DR. MARIO S. T. DE FREITAS	UTFPR

Curitiba, maio de 2006.

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

CARLOS MARCELO PEDROSO

**DESENVOLVIMENTO DE MODELO DE TRÁFEGO E MÉTODO
PARA MELHORIA DE DESEMPENHO DE SERVIDORES WEB**

Tese apresentada ao programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de “Doutor em Ciências” - Área de concentração: Telemática.

Orientadora: Prof^a. Dr^a. Keiko V. Ono Fonseca

Curitiba, maio de 2006.

Agradecimentos

Dedico meus sinceros agradecimentos para:

- À Professora Doutora Keiko Verônica Ono Fonseca, por sua dedicação em reuniões semanais de orientação ao longo dos últimos 5 anos e pelo incentivo à produção desta tese;
- Aos colegas que participaram das palestras e reuniões semanais. Muitos foram os alunos de mestrado: Douglas Mendes, Airton Kuada, Augusto Foronda, Marcia Kotelok, Marcelo Tostes e Edmundo Zuchowski. Os debates foram momentos de compartilhamento de idéias criativas e contribuíram para evolução dos trabalhos de todos os envolvidos;
- À minha família que me apoiou sempre que necessário: aos meus pais e irmãos, à minha esposa e à minha filha.

*Obrigado por tudo quanto
Você me fez por nada
Djavan*

Sumário

1	Introdução	1
1.1	Objetivos	2
1.2	Metodologia utilizada	3
1.3	Organização da Tese	5
2	Modelos de Tráfego	6
2.1	Introdução	6
2.2	Definições iniciais e conceitos básicos	7
2.2.1	Processos com dependência de longa duração	8
2.2.2	Processos com dependência de curta duração	8
2.2.3	Distribuições de cauda pesada	9
2.3	A auto-similaridade no tráfego de telecomunicações	10
2.3.1	Auto-similaridade e a geometria fractal	11
2.3.2	Definições matemáticas de um processo auto-similar	13
2.3.3	Identificação de um processo auto-similar	14
2.3.4	Não estacionariedade e <i>LRD</i>	19
2.4	Modelos de desempenho	20
2.4.1	Modelos com dependência de curta duração	20
2.4.2	Modelos com dependência de longa duração	25
2.5	Conclusão do capítulo	30
3	Um novo modelo para geração de tráfego <i>Web</i>	32
3.1	Introdução	32
3.2	Definições iniciais e conceitos básicos	34
3.3	Trabalhos Relacionados	34
3.4	Um modelo composto para servidores <i>Web</i> utilizando classificação semântica de conteúdo	36

3.5	Levantamento das classes de arquivos e da matriz de transição de estados .	38
3.5.1	Formato dos arquivos de <i>log</i>	39
3.5.2	Identificação das principais classes de tráfego	40
3.5.3	Identificação da matriz de probabilidade de transição de estados das classes de arquivos	41
3.5.4	Caracterização do tamanho dos objetos transmitidos em cada classe de arquivo	42
3.5.5	Caracterização do tempo de permanência em cada classe	44
3.6	Validação e simulação	47
3.7	Conclusões do capítulo e trabalhos futuros	47
4	Um método para melhorar o desempenho de servidores Web	52
4.1	Motivação	52
4.2	Métricas	54
4.3	Trabalhos correlatos	54
4.4	O Método Proposto	57
4.4.1	Algoritmo single rate Tree Color Marker - <i>srTCM</i>	59
4.5	Configuração automática de parâmetros	59
4.6	Análise de desempenho do método proposto	60
4.7	Simulação	62
4.7.1	Sistema com pequena utilização	63
4.7.2	Sistema com grande utilização	64
4.7.3	Considerações sobre a implementação	67
4.8	Conclusões do capítulo	68
5	Conclusões	70
A	Níveis do Nilo	72
B	Distribuições de Probabilidade	73

Lista de Figuras

2.1	Conjunto de Cantor.	11
2.2	Fractal floco de neve.	11
2.3	Nível mínimo anual do rio Nilo entre os anos de 622 a 1281.	15
2.4	Função de auto-correlação calculada para a série dos níveis mínimos anuais do rio Nilo (Figura 2.3).	15
2.5	Método da variância aplicado à série dos níveis mínimos do rio Nilo.	16
2.6	Série temporal acumulada para $N(0.1, 1)$	19
2.7	Série da Figura 2.6 diferenciada uma vez.	19
2.8	Diagrama de estados para modelar o tráfego de sistemas telefônicos.	20
2.9	Modelo ON-OFF e IPP.	22
2.10	Série temporal gerada com o modelo ARIMA para o exemplo e respectiva função de auto-correlação.	25
2.11	Modelo Hierárquico <i>ON-OFF</i> de 3 níveis.	29
3.1	Diagrama hipotético de classes semânticas para transmissão de objetos durante o estado <i>ON</i>	37
3.2	Volume de tráfego para as principais classes de arquivo transmitidos identificadas nos arquivos de <i>log</i> do servidor da WC98.	40
3.3	Histograma do tamanho dos arquivos da classe <i>Zip</i> do servidor da Copa do Mundo de 1998.	40
3.4	Distribuição de probabilidade teórica e distribuição amostral para cada uma das classes de arquivos do sistema IRCache. A classe <i>GIF1</i> é comparada com a distribuição Normal e as demais com a distribuição Lognormal.	45
3.5	Distribuição acumulada para o tamanho dos arquivos das principais classes de arquivos da Copa do Mundo de 1998 comparado com a distribuição Lognormal.	46
3.6	Distribuição acumulada de probabilidade para o tempo de permanência em cada classe comparada com a distribuição de Weibull (linha pontilhada).	48

3.7	Topologia da Simulação.	49
3.8	Série temporal representando o tráfego de saída do servidor da WC98 na escala de 1 segundo.	49
3.9	Estimador do parâmetro de Hurst utilizando a transformada Wavelet para o tráfego de saída do servidor WC98.	50
4.1	Componentes do sistema.	57
4.2	Número de tarefas no sistema com desvio padrão de $\sigma = 0, 2, 8, 16, 32$ no sistema $M/G/1$ e no sistema $M/G/1 PS$	61
4.3	Topologia da simulação.	63
4.4	Tempo médio de resposta percebido pelos clientes em um sistema não congestionado.	64
4.5	Tempo médio de resposta percebido na simulação do servidor em situação de grande carga.	65
4.6	Estimativa das funções de auto-correlação e auto-correlação parcial do tempo de resposta percebido pelos clientes na situação de grande carga da série da Figura 4.5.	66
4.7	Índice de justiça entre sessões de usuário com o servidor em situação de congestionamento da Figura 4.5.	66
4.8	Comparação da variação do valor de CIR para $\alpha = 0,01$ e $\alpha = 0,001$ ao longo da simulação.	67

Lista de Tabelas

2.1	Exemplo de geração de uma série com modelo auto-regressivo.	23
2.2	Exemplo de geração de uma série com modelo de médias móveis.	23
2.3	Exemplo de geração de uma série com modelo ARIMA.	24
3.1	Coleção de dados em estudo.	39
3.2	Formato dos registros do histórico de acessos do sistema <i>IRCache</i>	39
3.3	Percentual observado em relação ao volume total trafegado em bytes para as principais classes de arquivos transmitidos no sistema <i>IRCache</i>	41
3.4	Matriz de probabilidade transição de estados para o servidor da <i>WC98</i>	42
3.5	Sumário de estatísticas e ajuste de distribuição para os arquivos pertencentes às principais classes de arquivos do sistema <i>IRCache</i>	43
3.6	Sumário de estatísticas e ajuste de distribuição para os arquivos pertencentes às principais classes de arquivos da <i>WC98</i>	43
3.7	Distribuições ajustadas para o tempo de transmissão de arquivos em cada classe para o servidor <i>WC98</i>	44
4.1	Sumário dos parâmetros utilizados na geração de carga para o servidor <i>Web</i>	63
4.2	Sumário de estatísticas do tempo de resposta percebido pelos clientes da simulação do servidor com grande carga apresentado na Figura 4.5.	65

Resumo

Esta tese trata do desenvolvimento de um modelo para geração de tráfego sintético e de um método para melhoria de desempenho de servidores Web. Como hipótese fundamental das técnicas propostas, foi considerado o perfil auto-similar do tráfego apresentado por servidores Web. Até onde temos conhecimento, ambas as propostas são inovadoras e possuem vantagens em relação às técnicas conhecidas atualmente. O modelo para geração de tráfego sintético foi desenvolvido a partir do modelo SURGE (*Scalable URL Reference Generator*) e apresenta novas capacidades de representação da dinâmica de sistemas Web. O modelo proposto pode ser utilizado para geração de tráfego consistente com os padrões de tráfego real e também para análise de desempenho do servidor. Extensões do modelo podem estabelecer novos paradigmas de planejamento de capacidade de servidores Web. O método para melhoria de desempenho objetiva a redução do tempo médio de resposta percebido pelos clientes através da redução da latência de transmissão na rede. Possibilidades de aplicações e trabalhos futuros são discutidas.

Abstract

This thesis deals with the development of a model for Web servers synthetic traffic generation and a method for improvement of Web servers performance. We assume that the traffic produced by Web servers exhibits behavior consistent with the self-similar models. As far as we know, both proposals are novel and presents advantages with respect to the presently known techniques. The model for synthetic traffic generation was based on the SURGE (*Scalable URL Reference Generator*) model and presents new capacities for representation of Web systems dynamics. The model can also be used for performance analysis. Extensions of the model can establish new paradigms for Web servers capacity planning. The method for performance improvement is based on the reduction of the average response time perceived by the clients due to the reduction of the transmission latency over the network. Possibilities of future applications are explored.

Capítulo 1

Introdução

TRADICIONALMENTE, quando uma tecnologia de telecomunicações era desenvolvida demandava-se a implantação de uma infra-estrutura de rede própria para torná-la disponível. Ao longo do século passado as tecnologias de transmissão foram sucessivamente substituídas umas pelas outras. Assim ocorreu com o telefone, a comunicação de dados e a TV a cabo, cada um acompanhado por sua própria rede de serviços. Para cada sistema foram desenvolvidas técnicas de modelagem e avaliação de desempenho que tornaram possível aos engenheiros a realização de projetos e dimensionamento para garantir uma boa qualidade do serviço aos usuários.

No entanto, em meados da década de 1990 iniciou-se uma grande mudança que resultou em uma alteração na filosofia de projeto de sistemas de telecomunicações: iniciou-se um movimento em direção à adoção de tecnologias de redes convergentes. A convergência na área de telecomunicações se refere à utilização de uma única tecnologia de rede de comunicação digital como suporte a todos os serviços, com conseqüente economia de escala. Uma única rede irá suportar diversos tipos de tráfego; dados, voz e imagem compartilhando uma mesma infra-estrutura.

Paralelamente, ao longo da década de 1990 foram realizadas importantes descobertas na área de modelagem de tráfego produzido por redes de computadores. Descobriu-se que, via de regra, o tráfego produzido possui características auto-similares. Como conseqüência, segundo [Paxson e Floyd 1995], as ferramentas tradicionalmente utilizadas para o dimensionamento de sistemas de transmissão de voz já não se mostram adequadas às redes de dados de alta velocidade modernas. Esta descoberta abriu novos caminhos para pesquisa e desenvolvimento nesta área. O fenômeno da auto-similaridade do tráfego agregado em redes de computadores descrito por [Leland et al. 1994] estabeleceu uma noção importante para a compreensão da natureza dinâmica do tráfego das redes de

dados digitais modernas e abriu novas grandes vias de pesquisa nas áreas de análise de dados, modelagem matemática, teoria de filas e técnicas de controle utilizadas em redes de alta velocidade. Como hipótese fundamental das técnicas propostas nesta tese, foi considerado o perfil auto-similar do tráfego apresentado por servidores *Web* como reportado por Crovella e Bestavros em [Crovella e Bestavros 1995]. A auto-similaridade do tráfego observado em redes de comunicação digital continua sendo um tópico aberto para pesquisa, como pode ser visto em trabalhos recentes como [Melo e Fonseca 2005], [Veitch et al. 2005a], [Veitch et al. 2005b], [Ribeiro et al. 2005], [Sarvotham et al. 2005], [Park et al. 2005], [Stoev et al. 2005], [Rolls et al. 2005] e [Kim e Shroff 2005].

A partir deste esforço foram identificadas as características de auto-similaridade do tráfego em aplicativos (ex. servidores *Web*), redes *Ethernet* e até mesmo em tráfego de redes geograficamente distribuídas (em alguns casos).

A Internet tem sido cada vez mais importante na integração das tecnologias de comunicação. A carga de tráfego da Internet é dominada pelo protocolo *HTTP* (Hyper Text Transfer Protocol [Fielding et al. 1999]), que foi identificado em medições do *backbone* da Sprint em [Cao et al. 2004a] como sendo a classe de aplicativos mais utilizado na Internet, com uma faixa de 31% a 59% do total de bytes transmitidos. Adicionalmente, o protocolo *HTTP* está sendo utilizado como interface para entregar conteúdos para aplicativos mais especializados de alto nível. Neste trabalho, a atenção foi concentrada no desenvolvimento de novas técnicas envolvendo a modelagem e a melhoria de desempenho de servidores *HTTP*. No entanto os métodos e técnicas apresentados podem ser adaptados para outros tipos de servidores de aplicativo.

1.1 Objetivos

Esta tese trata de dois temas dentro do contexto da auto similaridade do tráfego observado em sistemas servidores *HTTP*. O primeiro propõe um modelo de tráfego para servidores *HTTP* que permite a geração de carga para simulações e possui boas características para realização de estudos analíticos. A validação do modelo foi realizada através do estudo de dados de servidores reais e de simulações computacionais.

O segundo tema estudado é o desempenho de sistemas servidores *HTTP* em situações onde o tempo de serviço possui grande variabilidade. O tráfego auto-similar observado na saída de servidores *HTTP* é explicado em [Crovella et al. 1998] como sendo consequência da distribuição de cauda pesada do tamanho dos arquivos transmitidos. Este perfil de tráfego pode produzir situações transientes de congestionamento nos servidores, o que

pode levar o sistema a apresentar problemas de desempenho. Como solução, é proposto um método para melhoria do tempo médio de resposta percebido aos clientes de servidores de página na presença de tráfego com grande variabilidade. O método proposto é baseado na marcação de tráfego de saída na camada de transporte (no servidor) e no encaminhamento e descarte prioritário no núcleo da rede (na camada de rede, pelos roteadores), observando-se a marcação realizada. Mostra-se que o tempo médio de resposta percebido pelos clientes pode ser melhorado com uso do método proposto. O método proposto pode ser implementado utilizando-se os mecanismos já previstos para uma rede de serviços diferenciados.

1.2 Metodologia utilizada

Foi realizado um estudo bibliográfico sobre o estado da arte na modelagem e avaliação de desempenho do tráfego de redes digitais. Foram estudadas as características de modelos clássicos e auto-similares.

A revisão bibliográfica foi realizada continuamente durante o desenvolvimento deste trabalho, devido a atualidade do tema

Foram realizados estudos analíticos utilizando modelos estocásticos, modelos de séries temporais e a teoria de filas. Também foram realizadas simulações para fornecer uma avaliação qualitativa dos métodos propostos.

Ao longo do trabalho foi dada ênfase na produção de artigos para conferências cujo retorno dado pelos revisores foi de grande valia na determinação dos rumos da pesquisa. Os resultados produzidos refletiram nos seguintes trabalhos:

Artigos publicados em conferências com arbitragem:

- Modeling WRR packet scheduler with Petri Nets [Pedroso e Fonseca 2002], apresentado no IEEE International Conference on Communication Singapore, ICCS'2002;
- Bandwidth Fairness of a single rate Three Color Marker algorithm implementation [Mendes et al. 2002], apresentado no IEEE International Conference on Communication Singapore, ICCS'2002;
- Um Modelo para Avaliação de Desempenho de Servidores Web Utilizando Classificação de Conteúdo [Short Paper] [Pedroso et al. 2005], apresentado no 4th International Information and Telecommunication Technologies Symposium (I2TS2005), Florianópolis, 2005;

- Um Método para Melhorar o Desempenho de Servidores Web que Apresentam Perfil de Tráfego Altamente Variável, [Pedroso e Fonseca 2005d], apresentado no 4th International Information and Telecommunication Technologies Symposium (I2TS2005), Florianópolis, 2005.
- Um modelo para geração de carga de servidores *Web* utilizando classificação de conteúdo [Pedroso e Fonseca 2006], XXIV Simpósio Brasileiro de Redes de Computadores, Curitiba, 2006.

Dissertação de conclusão de curso de mestrado:

- Modelagem de tráfego de servidores Web por classificação de conteúdo, Dissertação de mestrado apresentada ao curso de Pós Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, por Márcia Kotelok em 2006;

Monografias de conclusão de curso de graduação:

- Simulador de Redes de Petri Estocásticas [Greca e Gielow 2004], trabalho de conclusão do curso de Bacharelado em Engenharia de Computação da Pontifícia Universidade Católica do Paraná, 2003;
- Método para otimização da resposta de servidores [Muniz e Kamogawa 2003], trabalho de conclusão do curso de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica do Paraná, 2003;
- Servidor de vídeo sob demanda com tráfego condicionado [R.Scorsin e da Silva 2003], trabalho de conclusão do curso de Bacharelado em Engenharia de Computação, 2003.

Relatórios técnicos do CPGEI, que incluem artigos em submissão e artigos aceitos em conferências e não publicados por falta de recursos financeiros:

- Improving the average response time of Web servers as perceived by the clients [Pedroso e Fonseca 2005a];
- Web server workload generation by clustering the server files [Pedroso e Fonseca 2004f];

- A simple method to improve Web servers average time response and availability [Pedroso e Fonseca 2005c];
- A queue management discipline to protect aggregate traffic systems against bad behaving data streams [Pedroso e Fonseca 2004d];
- Traffic modeling of an e-learning server with ARMA model: why this server traffic is not self similar? [Pedroso e Fonseca 2004e];
- A method to improve average client perceived response time [Pedroso e Fonseca 2005b];
- Improving the performance of Web servers [Pedroso e Fonseca 2004a];
- Modelos de tráfego em redes de comunicações digitais [Pedroso e Fonseca 2004c];
- Improving the performance of Web servers [Pedroso e Fonseca 2004b];

1.3 Organização da Tese

Além desta seção introdutória, esta tese está organizada da seguinte maneira. O Capítulo 2 apresenta os conceitos de auto-similaridade de uma variável aleatória e definições básicas que serão utilizadas no conjunto do trabalho, bem como os principais modelos de tráfego utilizados atualmente para representar o tráfego em redes de alta velocidade. O Capítulo 3 apresenta a proposta de um novo modelo de tráfego baseado na separação de classes de arquivo transmitidas por um servidor de dados. No Capítulo 4 é proposto um método para melhorar o tempo médio de resposta de um servidor de páginas na presença de tráfego altamente variável. São apresentados resultados de simulações computacionais para demonstrar o funcionamento do método.

Capítulo 2

Modelos de Tráfego

2.1 Introdução

UM modelo de tráfego é uma representação que idealmente deve capturar precisamente todas as propriedades estatísticas relevantes do tráfego original, normalmente envolvendo componentes estocásticos. As componentes relevantes dependem da aplicação planejada para o modelo, sendo que em certos casos podem-se suprimir características não relevantes para o sistema em estudo. No entanto, muitas vezes tal modelo pode tornar-se extremamente complexo [Zukerman et al. 2003].

Identificar modelos simples e precisos para o tráfego de redes de dados digitais não é tarefa fácil, como pode ser visto na vasta literatura disponível sobre o assunto. Um consenso geral reside no fato que o tráfego real observado em redes de dados digitais não segue o modelo de Poisson em nenhum nível de agregação [Paxson e Floyd 1995][Willinger e Park 2000][Crovella e Bestavros 1995].

As técnicas de modelagem de desempenho incluem as técnicas analíticas, as simulações computacionais e a experimentação prática. Modelos de desempenho requerem modelos de tráfego precisos que possam capturar as características do tráfego real. Se os modelos de tráfego não representarem precisamente o tráfego real, o desempenho da rede pode ser superestimado ou subestimado [Adas 1997].

A análise de artigos publicados no XXI Simpósio Brasileiro de Telecomunicações de 2004 mostra que aproximadamente 50% dos artigos utilizam técnicas de simulação computacional como meio de obter soluções numéricas para os problemas tratados. Proporção similar pode ser observada nos trabalhos apresentados no *4th International Information and Telecommunication Technologies Symposium* de 2005. No entanto, a grande maioria dos trabalhos não utiliza modelos apropriados para geração de tráfego. A preferência

dos pesquisadores é pela utilização de fontes de tráfego constante ou mesmo seguindo o processo de Poisson. Em uma parcela destes trabalhos os resultados simulados poderiam ser diferentes caso fossem utilizados modelos auto-similares para geração de tráfego. Como reportado em [Paxson e Floyd 1995], o processo de Poisson não é apropriado para caracterizar o tráfego de redes de alta velocidade.

Este capítulo apresenta os principais conceitos envolvendo as técnicas modernas para modelagem e geração de tráfego sintético de redes digitais.

2.2 Definições iniciais e conceitos básicos

Uma série temporal é a amostragem seqüencial de uma variável durante um intervalo de tempo, via de regra longo, para possibilitar a identificação de padrões. Pode ser obtida analiticamente, ou também por simulação numérica, ou ainda, por medição experimental.

A observação de uma série temporal discreta realizada em instantes de tempo $\tau_1, \tau_2, \dots, \tau_t, \dots, \tau_n$ pode ser denotada por $X(\tau_1), X(\tau_2), \dots, X(\tau_t), \dots, X(\tau_N)$. Neste capítulo serão consideradas apenas séries temporais discretas, onde as observações são realizadas em um intervalo fixo h . Quando N valores sucessivos da série forem analisados, será escrito $X_1, X_2, \dots, X_t, \dots, X_N$ para denotar observações realizadas a intervalos de tempo equidistantes $\tau_0 + h, \tau_0 + 2h, \dots, \tau_0 + th, \dots, \tau_0 + Nh$. A esperança será denotada $E(X) = \mu$; a variância será denotada por $V[X] = E[(X - \mu)^2] = \sigma^2$; a auto-covariância com defasagem k será denotada por $E[(X_t - \mu)(X_{t-k} - \mu)] = \gamma_k$ e a auto-correlação com defasagem k será denotada por $\rho_k = \gamma_k / \gamma_0$.

Uma medição de tráfego em uma rede digitail resulta em uma série temporal discreta $Y_t, t \in \mathbb{Z}$, onde Y_t pode ser interpretado como o volume de tráfego total (medido em pacotes, bytes ou bits) observado a partir do instante inicial 0 até o instante t , sendo uma função cumulativa. Para minimizar a confusão do estudo de processos cumulativos, considere o processo não cumulativo X_t que pode ser utilizado na definição recursiva $X_t = Y_t - Y_{t-1}$. No estudo de tais séries temporais, será considerado o processo não cumulativo X_t .

Definição 2.1 Estacionariedade estrita [Willinger e Park 2000]. X_t é estritamente estacionário se $[X_{t_1}, X_{t_2}, \dots, X_{t_n}]$ e $[X_{t_1+k}, X_{t_2+k}, \dots, X_{t_n+k}]$ possuem a mesma distribuição conjunta para todo $n \in \mathbb{N}$.

Na estacionariedade estrita, o processo deslocado por k , chamado X_{t+k} , e o primeiro chamado X_t , devem ser equivalentes.

A estacionariedade estrita é muito restritiva. Existe muito interesse em uma forma menos rígida, representada pela estacionariedade de segunda ordem (ou fraca, ou covariância).

Definição 2.2 Estacionariedade fraca ou de segunda ordem [Willinger e Park 2000]. A função de auto-covariância $\gamma(r, s) = E[(X_r - \mu)(X_s - \mu)]$ deve satisfazer à relação de invariância $\gamma(r, s) = \gamma(r + k, s + k) \quad \forall r, s, k \in \mathbb{Z}$.

A estacionariedade fraca exige que a covariância de duas amostras não sobrepostas deslocadas no tempo deve ser sempre igual; os dois primeiros momentos devem existir e serem finitos. Para um processo deste tipo, o valor esperado converge e a variância pode não convergir.

2.2.1 Processos com dependência de longa duração

Processo com dependência de longa duração ou Long Range Dependence (LRD) é o fenômeno que ocorre quando observações de um evento são significativamente correlacionadas com observações muito distantes no tempo. Este fenômeno é de particular interesse na modelagem de tráfego, uma vez que as pesquisas comprovaram que o tráfego em redes modernas de comunicação freqüentemente apresenta *LRD* [Leland et al. 1994].

Em um processo *LRD* a função de auto-correlação ρ_k decai para zero muito lentamente em função do deslocamento k . Segundo [Willinger e Park 2000] processos auto-similares apresentam *LRD*.

Definição 2.3 Processo com dependência de longa duração. *Seja $X_t, t = 1, 2, \dots$, um processo estacionário de segunda ordem. Se existe uma constante c tal que $\gamma_k \sim c/k^\alpha, 0 < \alpha < 1$, então o processo possui LRD.*

Hurst foi um famoso hidrólogo no Egito e no rio Nilo que mostrou em 1951 que o nível do rio Nilo é uma seqüência *LRD*. Por razões históricas, o parâmetro de Hurst $H = 1 - \alpha/2$ é utilizado para representar a auto-similaridade (ver item 2.3).

Um processo com dependência de longa duração apresenta auto-similaridade assintótica de segunda ordem [Willinger e Park 2000].

2.2.2 Processos com dependência de curta duração

Processo com memória de curta duração ou Short Range Dependence (SRD) indica o fenômeno onde a observação corrente não é correlacionada com observações muito

antigas. Para um processo *SRD*, a função de auto-correlação decai rapidamente para zero [Willinger e Park 2000].

O valor $H = \frac{1}{2}$ é o limite entre a *LRD* e a *SRD*. Processos com $\frac{1}{2} < H < 1$ são *LRD*, enquanto processos com $0 < H < \frac{1}{2}$ são *SRD*. O efeito da *LRD* será maior quando H estiver próximo a 1 [Willinger e Park 2000].

2.2.3 Distribuições de cauda pesada

Uma das explicações sobre as causas da auto-similaridade do tráfego de telecomunicações é a natureza de elementos tais como o tamanho de arquivos ou o tempo que o usuário processa informações, que normalmente podem ser descritos por distribuições de cauda pesada [Crovella e Bestavros 1995][Leland et al. 1994].

Definição 2.4 Distribuição de cauda pesada. *Uma variável aleatória X possui distribuição de cauda pesada se [Willinger e Park 2000]*

$$\Pr\{X > x\} \sim x^{-\alpha}, \quad x \rightarrow \infty, \quad 0 < \alpha < 2 \quad (2.1)$$

onde α é chamado *parâmetro de forma* (shape parameter).

A auto-similaridade do tráfego pode ser obtida através de uma soma de fontes de tráfego *ON-OFF* (ver item 2.4.1) onde os períodos de tempo *ON* e *OFF* têm uma distribuição de cauda pesada, como a distribuição de Pareto [Leland et al. 1994].

A distribuição de Pareto é uma distribuição de cauda pesada cuja função de distribuição de probabilidade [Willinger e Park 2000] é dada por

$$\Pr\{X \leq x\} = 1 - \left(\frac{\beta}{x}\right)^\alpha, \quad \beta \leq x \quad (2.2)$$

onde $0 < \alpha < 2$ é o parâmetro de forma e β é chamado de parâmetro de localização. A esperança é dada por $E[X] = \alpha\beta/(\alpha - 1)$. A principal característica de uma variável aleatória como esta é a variabilidade extrema exibida.

A literatura sugere que a auto-similaridade do tráfego observado em redes de telecomunicações está associada ao tamanho dos objetos transmitidos, que normalmente segue uma distribuição de cauda pesada [Willinger e Park 2000]. Uma das maneiras utilizadas para verificar se uma distribuição possui cauda pesada é através do gráfico da distribuição complementar em escala logarítmica (*log-log complementary distribution, LLCD*). Este gráfico representa o valor da distribuição de probabilidade acumulada complementar $\bar{F}(x) = 1 - F(x)$ com $F(x) = \Pr\{X \leq x\}$, em escala logarítmica, de modo a procurar

a invariância $d \log(\overline{F}(x))/d \log(x) = -\alpha$, $x > \theta$. Na prática, escolhe-se um valor de θ a partir do qual o gráfico passa a ser linear.

Em alguns casos têm sido utilizadas na modelagem do tempo de serviço de servidores distribuições de decaimento sub-exponencial, como as distribuições Lognormal ou Weibull, e que possuem variância finita [Willinger e Park 2000].

2.3 A auto-similaridade no tráfego de telecomunicações

A caracterização clássica de tráfego seguia, via de regra, um modelo Markoviano (ou processo de Poisson). Nesta teoria, o tráfego de entrada não possui memória (por exemplo, a chegada de um novo pacote é independente da chegada outros pacotes) e o intervalo entre chegadas segue uma distribuição exponencial negativa. Este modelo pode ser facilmente contestado através da observação do tráfego gerado por um aplicativo. Pode-se colocar o exemplo de um servidor de páginas na Internet: em uma conexão *HTTP*, onde em geral uma solicitação de um cliente é respondida com a transmissão de arquivos contendo hipertextos, sons, imagens e muitos outros conteúdos. Todas estas informações estarão sendo colocadas em mais de um pacote para transmissão. Desta forma, a chegada de um pacote não representa mais um evento independente; na realidade a chegada de um pacote aumenta a chance de chegada do próximo pacote. Este raciocínio mostra empiricamente que a hipótese da independência não pode ser considerada premissa válida para a modelagem de tráfego.

A primeira abordagem para resolver este problema foi a introdução do conceito de “trens de pacotes”, publicado em 1986 [Jain e Routhie 1986]. Este modelo assume que um grupo de pacotes percorre a rede como um trem, ao contrário do modelo Markoviano. O intervalo entre os pacotes dentro do trem seria muito pequeno, mas o intervalo entre os “trens de pacotes” continuava sendo um modelo Markoviano.

A inacurácia dos modelos existentes até meados da década de 1990 para descrever o tráfego agregado observado nas redes de computadores levou os pesquisadores à utilização de um modelo baseado em fractais, chamado auto-similar [Willinger e Park 2000]. Um objeto auto-similar preserva suas características independentemente da escala em que é observado. Este objeto pode ser uma série temporal, uma imagem, um sistema dinâmico ou qualquer relação matemática de interesse.

O modelo auto-similar foi estudado pelo matemático George Cantor no século XIX,

em um estudo que o levou à teoria sobre fractais aprimorada por Benoit B. Mandelbrot [Willinger e Park 2000].

A montagem do Conjunto de Cantor apresentado na Figura 2.1 [Kaplan e Glass 1998], onde a seqüência de iterações é representada usando a dimensão vertical, produz um objeto auto-similar. Para a montagem do conjunto, um dado segmento de reta é dividido em 3 partes iguais, sendo a parte central do segmento descartada, e assim recursivamente. O modelo geométrico referido é atingido no limite de infinitas iterações do processo descrito.



Figura 2.1: Conjunto de Cantor.

Um objeto auto-similar em 2 dimensões é mostrado na Figura 2.2. Este objeto é conhecido como floco de neve Koch [Kaplan e Glass 1998], formado pela divisão do segmento de reta em três partes iguais, removendo-se o segmento central e incluindo-se dois segmentos do mesmo tamanho do segmento que foi removido conectados em suas extremidades aos segmentos restantes e entre si. A regra recursiva é executada sobre todos os segmentos de reta restantes. Imagine que a montagem da figura prossiga indefinidamente. Qualquer ampliação em escala de uma porção do objeto resultante possui semelhanças entre si.



Figura 2.2: Fractal floco de neve.

2.3.1 Auto-similaridade e a geometria fractal

O termo auto-similar descreve a geometria de objetos em que uma pequena parte quando expandida se parece com o todo [Kaplan e Glass 1998]. Muitos objetos encontrados na natureza apresentam características auto-similares (ou fractais aproximados). Pode-se citar como exemplo a estrutura geométrica de nuvens, montanhas e árvores.

Um dos conceitos fundamentais para a compreensão da auto-similaridade é o conceito de dimensão, que será abordado a seguir. Na geometria Euclidiana, um ponto possui dimensão zero, uma linha possui dimensão 1, um plano possui dimensão 2 e assim por diante. A dimensão fractal de um objeto é definida pela Equação 2.3: um objeto auto-

similar é caracterizado pelo número N de cópias de si próprio criado com uma alteração de escala, enquanto ε indica o tamanho relativo da cópia em relação ao objeto original.

Suponha, por exemplo, o conjunto de Cantor apresentado na Figura 2.1. Cada divisão da figura original produz uma mudança de escala onde são criadas duas figuras iguais à original ($N = 2$) e o tamanho da figura será 3 vezes menor ($\varepsilon = 3$). O processo se repete de modo que a dimensão $d = 0,631$ permanece invariável. Para o fractal floco de neve apresentado na Figura 2.2, a quantidade de novos segmentos criados a partir da figura original será $N = 4$, com seu tamanho reduzido em $\varepsilon = 3$. A dimensão fractal deste objeto será $d = 1,262$. É possível determinar a dimensão fractal de um objeto observado fazendo uso da Equação 2.3 [Kaplan e Glass 1998],

$$d = \frac{\log N}{\log \varepsilon} \quad (2.3)$$

O procedimento, chamado *contagem de caixas*, é o seguinte:

1. Coloque um *grid* sobre o objeto com quadrados de dimensão ε_0 . Conte quantos quadrados contêm pontos da figura e chame o resultado de $N(\varepsilon_0)$;
2. Repita o passo 1 utilizando quadrados com dimensão $\varepsilon_1 = \varepsilon_0/2, \varepsilon_2 = \varepsilon_1/2, \dots$ e anote o resultado em $N(\varepsilon_0), N(\varepsilon_1), \dots$;
3. A dimensão d é o número tal que $\lim_{\varepsilon \rightarrow 0} a \cdot \varepsilon^{-d}$ (derivado da Equação 2.3), onde a é uma constante. Na prática, d pode se estimado como

$$d = \frac{\log N(\varepsilon_{i+1}) / \log N(\varepsilon_i)}{\log \varepsilon_{i+1} / \log \varepsilon_i} \quad (2.4)$$

A grande dificuldade do procedimento apresentado é selecionar o valor apropriado para i . Objetos reais não possuem um nível de granularidade infinitamente grande ou infinitamente pequeno. Além disso, a auto-similaridade pode manifestar-se apenas em determinadas escalas.

A equação 2.3 também traduz a importante propriedade dos fractais de invariância de escala. Uma função na forma $y(x) = x^a$ mostra um gráfico reto na escala $\log y \times \log x$, com uma determinada inclinação d . Se os valores de x forem multiplicados por um fator de escala $c(x' = c \cdot x)$, a nova reta formada pelos novos pares $\{x', y'\}$ continuará com a mesma inclinação d . Esta propriedade não se verifica, por exemplo, para funções exponenciais, na forma $y(x) = a^x$. Desta forma, a dimensão fractal d de um conjunto auto-similar é sempre a mesma, independente da escala em que é medido.

O parâmetro de Hurst está relacionado com a dimensão fractal através da relação $d = 2 - H$.

Exemplo 2.1 *O fractal do floco de neve mostrado na Figura 2.2 possui dimensão fractal dada por $d = 1,262$ e parâmetro de Hurst dado por $H = 0,738$. Na prática, a figura que representa o tráfego de uma rede de computadores será um objeto com dimensão fractal entre 1 e 2. Isso leva a uma possível variação para o parâmetro de Hurst na faixa entre 0 e 1.*

2.3.2 Definições matemáticas de um processo auto-similar

Suponha uma variável aleatória com $E[X] = 0$. Pela estacionariedade, $\gamma(r, s) = \gamma(r - s, 0)$ e a auto-covariância será dada por γ_k . Para formular a invariância de escala, primeiro define-se o processo agregado $X^{(m)}$ de X em um nível de agregação m , como mostrado pela equação

$$X_k^{(m)} = \frac{1}{m} \sum_{t=m(k-1)+1}^{mk} X_t \quad (2.5)$$

Isso significa que X_t é particionado em blocos não sobrepostos de tamanho m , seus valores são anotados e k é utilizado como índice nestes blocos. O valor de X_t representa a quantidade de informação observada.

Seja $\gamma^{(m)}(k)$ a função de auto-covariância de $X^{(m)}$. Assumindo que o processo é estacionário de segunda ordem, pode-se colocar a seguinte definição de auto-similaridade [Willinger e Park 2000]:

Definição 2.5 X_t é exatamente auto-similar de segunda ordem com parâmetro Hurst H ($\frac{1}{2} < H < 1$) se

$$\gamma_k = \frac{\sigma^2}{2} [(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}], \quad \forall k \geq 1 \quad (2.6)$$

A Equação 2.6 implica em $\gamma_k = \gamma_k^{(m)}$, $\forall m \geq 1$. A auto-similaridade estacionária tem sido o modelo dominante para a modelagem de tráfego em redes de comunicação modernas [Willinger e Park 2000]. Normalmente a função de auto-correlação de um processo auto-similar apresenta um lento decaimento em função do deslocamento k .

Definição 2.6 X_t é assintoticamente auto-similar de segunda ordem se

$$\lim_{m \rightarrow \infty} \gamma_k^{(m)} = \frac{\sigma^2}{2} [(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}], \quad \forall k \geq 1 \quad (2.7)$$

Um processo será *auto-similar do ponto de vista estocástico* se o processo mantém características estocásticas sobre uma certa faixa de escalas, de modo a satisfazer a relação dada pela Equação 2.8. A interpretação desta equação é a seguinte: o processo estocástico X_t deslocado em escala por um fator constante m possui aproximadamente as mesmas características de X_t em termos de sua distribuição de probabilidade. A amplitude do processo X_{mt} resultante será o de X_t multiplicado por m^H , com H representando o parâmetro de Hurst.

$$X_{mt} \cong_d m^H X_t, \quad m > 0 \quad (2.8)$$

Pode ser encontrado em [Willinger e Park 2000] o desenvolvimento matemático rigoroso para o conceito em questão, bem como várias implicações. Em [Willinger e Park 2000] são encontrados métodos para a determinação da auto-similaridade e do parâmetro de Hurst em função de medições do processo X_t .

2.3.3 Identificação de um processo auto-similar

Vários métodos têm sido utilizados para identificar a auto-similaridade de uma série temporal. Esta seção irá descrever vários métodos conhecidos.

A Figura 2.3 apresenta uma série temporal que representa o nível mínimo do rio Nilo entre os anos de 622 a 1281, em Roda, próximo ao Cairo (dados de [Toussoun 1925], citado por [Koutsoyiannis 2004]). Esta série motivou um dos primeiros estudos acerca do fenômeno da *LRD*, que foi estudado por Hurst no ano de 1951, em um estudo na área da hidrologia. Os dados consistem em um total de 663 observações. A série completa é mostrada no Apêndice.

Um dos fatores que indicam a presença da auto-similaridade é o lento decaimento da função de auto-covariância. A Figura 2.4 apresenta a auto-correlação para a série temporal do nível mínimo do rio Nilo, indicando uma possível presença da auto-similaridade. No entanto, a auto-similaridade pode ser facilmente confundida com a não estacionariedade de um processo estocástico, uma vez que a não estacionariedade também apresenta uma função de auto-correlação com decaimento lento. A série com os níveis mínimos do Nilo apresenta uma leve tendência ao incremento, o que pode indicar uma não estacionariedade.

Estatística R/S A auto-similaridade manifesta-se através de uma linha reta no gráfico *log-log* na estatística R/S . Para um dado conjunto de números $\{X_1, \dots, X_n\}$ com

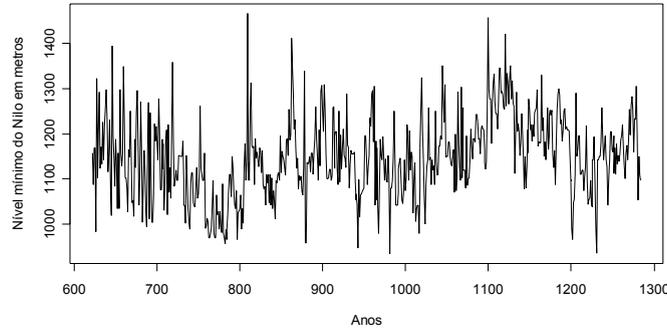


Figura 2.3: Nível mínimo anual do rio Nilo entre os anos de 622 a 1281.

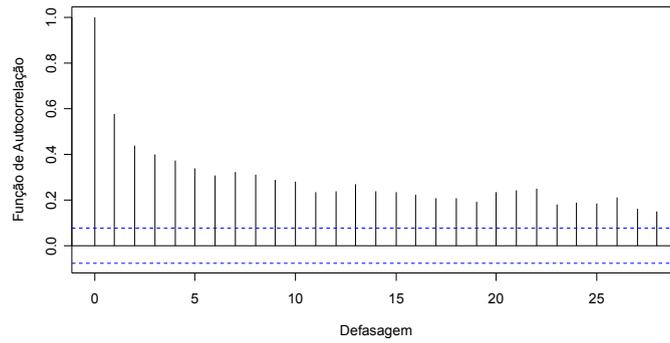


Figura 2.4: Função de auto-correlação calculada para a série dos níveis mínimos anuais do rio Nilo (Figura 2.3).

média amostral $\hat{\mu}$ e variância amostral $\hat{S}^2(n)$, o parâmetro de Hurst H é dado pela razão $\hat{R}(n)/\hat{S}(n)$ (ou estatística R/S), onde \hat{R} é obtido por

$$\hat{R}(n) = \max \left\{ \sum_{i=1}^k (X_i - \hat{\mu}), 1 \leq k \leq n \right\} - \min \left\{ \sum_{i=1}^k (X_i - \hat{\mu}), 1 \leq k \leq n \right\} \quad (2.9)$$

O valor esperado de $R(n)/S(n)$ assintoticamente satisfaz a relação $E[R(n)/S(n)] \rightarrow a n^H$ conforme $n \rightarrow \infty$ com $\frac{1}{2} < H < 1$.

Método da Variância Este é um método simples e apresenta resultados aproximados. Consiste em verificar a auto-similaridade assintótica através do gráfico do estimador $\sigma^{2(m)}$ da variância do processo agregado, dado pela Equação 2.10. Para valores suficientemente grandes de m , $\sigma^{2(m)} \sim c m^{-2(1-H)}$ para uma constante c . Isto pode ser verificado em uma escala logarítmica.

$$\sigma^{2(m)} = \frac{m}{N} \sum_{t=1}^{N/m} (X_t^{(m)} - \mu)^2 \quad (2.10)$$

Neste método, a variância $\sigma^{2(m)} = \text{Var}\{X_t^{(m)}\}$ em um nível de agregação m deve ser colocada em um gráfico em escala logarítmica ($\sigma^{2(m)} \times m$). A inclinação assintótica da reta formada será $\alpha = -2(1 - H)$.

Exemplo 2.2 Foi calculado o estimador da variância para o processo agregado da série temporal com os níveis mínimos do rio Nilo da Figura 2.3. O resultado, na forma de um gráfico em escala logarítmica de $\sigma^{2(m)} \times m$ é apresentado na Figura 2.5. Nesta figura também é mostrada a equação da reta calculada pelo método de regressão linear e sua equação. Tomando-se o valor de $\alpha = -0,3615$ pode-se calcular $-0,3615 = -2(1 - H)$ e encontra-se o valor de $H = 0,81925$. Esse valor está consistente com a geometria observada na Figura 2.3, tendo em vista que $d = 2 - H = 1,18075$. Um valor de dimensão fractal compreendido entre 1 e 2 se refere a um objeto geométrico com uma estrutura organizacional que o distingue de um segmento de arco suave ($d = 1$), porém não consistindo numa figura plana ($d = 2$) por ter área nula. Qualitativamente, os gráficos referentes ao tráfego em uma rede de computadores também apresentam essa propriedade.

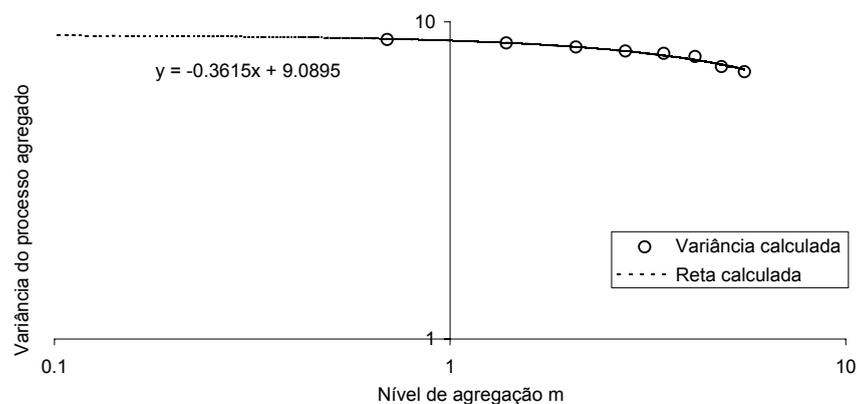


Figura 2.5: Método da variância aplicado à série dos níveis mínimos do rio Nilo.

Transformada *Wavelet* Wavelets são funções matemáticas que seccionam os dados em diferentes componentes de frequência e então estudam cada componente com uma resolução ajustada à sua escala. Elas possuem vantagens sobre os métodos tradicionais de Fourier para a análise de situações onde o sinal contém descontinuidades e picos. Wavelets foram desenvolvidas independentemente nos campos da matemática, física quântica, engenharia elétrica e geologia sísmica [Graps 1995].

A idéia fundamental é realizar a análise de acordo com a escala. Wavelets são funções com determinadas propriedades que as tornam adequadas a servirem de base para decomposição de outras funções, assim como senos e cossenos servem de base para decomposições

de Fourier. Quando o sinal é examinado em uma janela grande e pequena, certas características de interesse podem ser identificadas.

O procedimento de análise adota uma função padrão (ou “*mother wavelet*”). A análise temporal é realizada com uma versão em alta frequência da função padrão e a análise em frequência é realizada com uma versão em baixa frequência da função padrão. A função original é representada em termos de uma expansão utilizando coeficientes em combinação linear da onda padrão. Logo, a escolha da onda padrão foi tema de estudos constantes. A primeira família de funções utilizada, proposta por Alfred Haar em 1909, consistia de um pulso positivo seguido por um pulso negativo. Uma família de funções com propriedades atraentes foi descoberta por Ingrid Daubechies em 1987 e estas funções tem sido chamadas Daubechies Wavelets desde então. As funções Daubechies possuem conexões com a teoria dos fractais.

O procedimento básico de análise de uma série temporal consiste em um algoritmo recursivo que constrói a função, desloca-a e troca sua escala. Logo, a função original é decomposta em um conjunto de escalas e representada como parâmetros da função padrão, o que torna a análise menos sensível ao ruído porque mede as flutuações médias do sinal em diferentes escalas.

A transformada contínua Wavelet é definida pela Equação 2.11. $x(t)$ representa a função contínua no tempo sobre a qual se deseja aplicar a transformada. $\psi(t)$ é a função de transformação e é chamada Wavelet mãe (ou padrão). As variáveis s e τ são inteiros que re-escalam e deslocam a função mãe para gerar Wavelets. O índice de escala s indica a largura da wavelet e o índice de localização τ dá sua posição. As funções padrão são normalmente re-escaladas ou dilatadas por potências de 2 e transladadas por números inteiros. O que faz a base da wavelet especialmente interessante é a auto-similaridade causada pelos re-escalamentos e dilatações.

$$CWT_x^\psi(\tau, s) = \Psi_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^*\left(\frac{t - \tau}{s}\right) dt \quad (2.11)$$

A Wavelet é colocada no início do sinal no ponto correspondente a $t = 0$. A função Wavelet na escala $s = 1$ é multiplicada pelo sinal e então integrada sobre todo o tempo. O resultado da integração é então multiplicado pelo número constante $1/\sqrt{|s|}$. Esta multiplicação é realizada para normalização de energia (para que o sinal transformado tenha a mesma energia em todas as escalas). O resultado final é o valor da transformação, ou seja, o valor da transformada contínua Wavelet no tempo zero e escala $s = 1$. Este valor será correspondente ao ponto $\tau = 0$, $s = 1$ no plano tempo-escala.

Para realizar o cálculo sobre uma amostra discreta, deve ser utilizada a transformada discreta Wavelet. Neste caso, uma função $f(t)$ pode ser representada por um somatório de coeficientes multiplicados pela função padrão, de modo semelhante à expansão de termos de uma série de Fourier, mas definido por dois parâmetros, como indicado na Equação 2.12.

$$f(t) = \sum_{\tau} \sum_s a_{s,\tau} \psi_{s,\tau}(t) \quad (2.12)$$

Os coeficientes são dados por $a_{s,\tau} = \int f(t) \psi_{s,\tau}(t) dt$, sendo a função padrão dada por $\psi_{s,\tau}(t) = 2^{s/2} \psi(2^s t - \tau)$. A transformada discreta em uma dimensão pode ser calculada com o algoritmo Mallat [Mallat 1989].

O resultado da transformada consiste em um conjunto de valores de escala seguido pelas faixas de valores da função Wavelet (ou coeficientes Wavelet), em frequência crescente. Os tamanhos das faixas destes coeficientes são ordenados em potências crescentes de 2. Se existem N elementos no conjunto de dados (onde N é potência de 2), as faixas de coeficientes seguindo o valor de escala terão tamanhos $2^0, 2^1, 2^2 \dots N/2$.

Em [Abry e Veitch 1998] é estabelecido um método para determinação do parâmetro de Hurst em uma série baseado na transformada Wavelet. O método é relativamente imune a ruído e a não estacionariedade dos dados. O parâmetro de Hurst de um conjunto de dados é calculado a partir da densidade espectral da transformada Wavelet. O gráfico de densidade espectral é gerado do espectro de potência da transformada. A equação para o cálculo da potência normalizada para uma oitava s é dada pela equação $P_s = \frac{1}{2^s} \sum_{i=0}^{2^s-1} c_s^2$

A potência é calculada da soma dos quadrados dos coeficientes (resultado da transformada) para a oitava s . Uma oitava contém 2^s coeficientes. A soma dos quadrados é normalizada realizando a divisão por 2^s , resultando na potência normalizada. Na análise espectral não é sempre necessária a normalização da potência. No entanto, para o cálculo do expoente de Hurst a potência deve ser normalizada.

O expoente de Hurst é calculado a partir da densidade espectral, calculando-se uma reta utilizando uma regressão linear sobre o conjunto de pontos (x_s, y_s) , onde x_s é a oitava e y_s é o \log_2 da potência normalizada. A inclinação desta reta é *proporcional* ao estimador do expoente de Hurst. O parâmetro de Hurst será dado por $H = \left| \frac{\text{slope}-1}{2} \right|$.

Exemplo 2.3 *O cálculo do parâmetro de Hurst para a série com o nível mínimo do rio Nilo apresenta como resultado $H=0.8435$ (com 95% de confiança no intervalo entre $[0.770, 0.917]$).*

2.3.4 Não estacionariedade e *LRD*

Em certos casos, um processo estocástico não estacionário pode ser confundido com a *LRD*. Uma série temporal obtida a partir da distribuição normal acumulada com média 0,1 e desvio padrão 1 é mostrada na Figura 2.6, bem como sua função de auto-correlação. Apesar do lento decaimento da função de auto-correlação, a série não apresenta *LRD*: ela apenas não é estacionária. A não estacionariedade é evidente quando observa-se a tendência geral da série. Aplicando-se uma diferenciação à série da Figura 2.6, obtém-se a série estacionária mostrada na Figura 2.7; esta sim uma série estacionária que pode ser tratada com um modelo *SRD*. Observe o rápido decaimento da função de auto-correlação da série diferenciada na Figura 2.7, indicando o processo formador do tipo *SRD*.

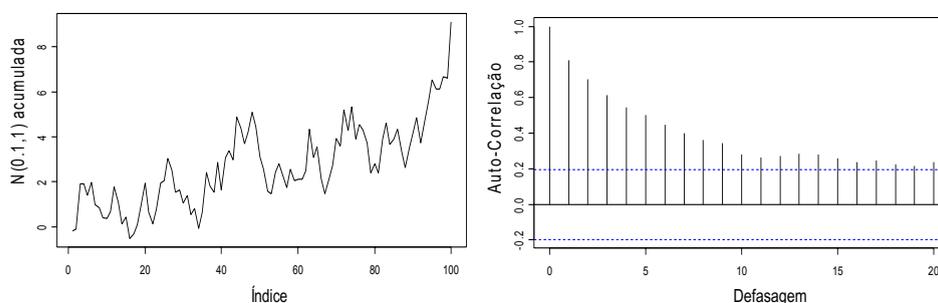


Figura 2.6: Série temporal acumulada para $N(0.1, 1)$.

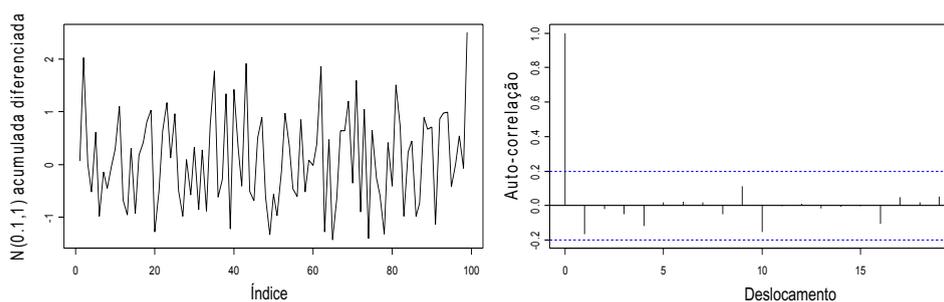


Figura 2.7: Série da Figura 2.6 diferenciada uma vez.

Como demonstrado, o fenômeno *LRD* pode ser confundido com a não estacionariedade de uma série. Desta maneira, a primeira providência da análise de uma série deve ser a remoção de tendências (por diferenciação) e a procura por sazonalidades; a não observação desta regra pode levar a conclusões incorretas.

2.4 Modelos de desempenho

Nos últimos anos, muito esforço tem sido realizado para desenvolver modelos que descrevem apropriadamente o tráfego de redes de computadores [Crovella e Bestavros 1995], [Leland et al. 1994], [Willinger e Park 2000], [Muscariello et al. 2004], [Nossenson e Attiya 2004] e [Gong et al. 2005]. Um dos resultados apresentados por estes trabalhos mostra que o modelo auto-similar descreve apropriadamente o tráfego agregado observado, sendo necessário o desenvolvimento de novos modelos para representação do tráfego atual. Nos últimos 10 anos (e possivelmente ainda no futuro), geradores sintéticos de tráfego são um componente essencial em virtualmente toda a simulação da Internet.

A seção 2.4.1 apresenta os modelos clássicos, utilizados nos casos onde não é observada a auto-similaridade do tráfego. Na seção 2.4.2 são apresentados modelos desenvolvidos especificamente para sistemas onde o tráfego apresenta características auto-similares.

2.4.1 Modelos com dependência de curta duração

Cadeias de Markov

Este modelo supõe que as atividades da fonte de tráfego podem ser modeladas de acordo com um número finito de estados [Adas 1997].

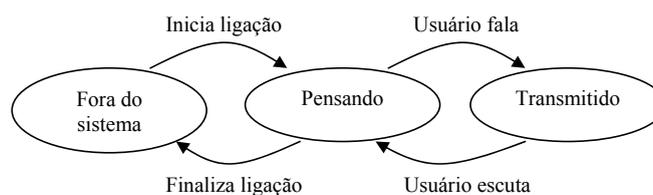


Figura 2.8: Diagrama de estados para modelar o tráfego de sistemas telefônicos.

A Figura 2.8 mostra um modelo muito utilizado em telefonia. Ele representa os estados de um usuário do sistema de telefonia: ou ele está fora do sistema, ou está transmitindo pacotes (falando) ou está pensando (escutando).

Segundo [Taylor e Karlin 1998], um processo de Markov X_t é um processo estocástico tal que, dado o valor de X_t , os valores de X_s para $s > t$ não são influenciados por valores de X_u para $u < t$. Ou seja, a probabilidade de qualquer comportamento futuro do processo, quando o estado corrente é conhecido exatamente, não é alterado por qualquer conhecimento adicional sobre o estado passado do processo.

Uma cadeia de Markov discreta é um processo de Markov cujo o espaço de estados é finito e cujo índice de tempo é $T = \{1, 2, \dots\}$. É conveniente rotular o espaço de estados da cadeia de Markov por números inteiros não negativos $i = \{0, 1, \dots\}$, que será denotado por $X_n = i$, onde i é a representação do estado e n a representação do tempo.

A probabilidade de X_{n+1} se encontrar no estado j dado que X_n está no estado i é chamado de probabilidade de transição em um passo, e é denotado por $P_{ij}^{n,n+1}$. Isto é, $P_{ij}^{n,n+1} = \Pr \{X_{n+1} = j | X_n = i\}$ [Taylor e Karlin 1998]. Quando as probabilidades de transição são independentes do tempo (probabilidades estacionárias no tempo), a cadeia de Markov possui uma matriz de transição de probabilidades estacionária, dada pela matriz quadrada

$$\begin{bmatrix} P_{00} & P_{01} & \dots & P_{0j} \\ P_{10} & P_{11} & \dots & P_{1j} \\ \vdots & \vdots & \ddots & \vdots \\ P_{i0} & P_{i1} & \dots & P_{ij} \end{bmatrix}$$

A partir da matriz de probabilidade de transição, é possível deduzir os parâmetros de desempenho do sistema. No entanto, a utilização deste modelo implica que o futuro depende apenas do estado corrente e não dos estados anteriores nem do tempo já gasto no estado atual. Isto restringe a variável aleatória que descreve o tempo gasto no estado para uma distribuição geométrica para o caso discreto e para uma distribuição exponencial para o caso contínuo [Taylor e Karlin 1998].

Considerando a cadeia de Markov de tempo contínuo [Taylor e Karlin 1998], onde o ritmo de transições de um estado i para outro estado j seja dado por $q_{ij} = \lambda_i$. Caso existam mais transições transformando o estado i para o estado j , então a taxa de transições será dada por $q_{ij} = q_{i1} + q_{i2} + \dots$. Caso não existam transições levando o estado i ao estado j , $i \neq j$, então $q_{ij} = 0$. Se $i = j$, q_{ii} será determinado para satisfazer a equação $\sum_j q_{ij} = 0$. A matriz $Q = q_{ij}$ representa as taxas de transições do espaço de estados e torna possível o cálculo das probabilidades de estado estacionário $\Pi = [\pi_1 \pi_2 \dots \pi_s]$, que pode ser realizado resolvendo o sistema linear dado pela Equação 2.13 [Murata 1989].

$$\Pi Q = 0, \quad \sum_{i=1}^s \pi_i = 1 \quad (2.13)$$

Modelos *ON-OFF* e *IPP*

A Figura 2.9 mostra um dos modelos mais populares para geração de tráfego de voz, o modelo *ON-OFF* [Adas 1997]. Neste modelo, os pacotes são gerados somente durante

o período onde o estado do sistema é *ON*. Durante este período, os pacotes são gerados com intervalos fixos entre chegadas; a sucessão entre estados *ON* e *OFF* é modelada pela distribuição exponencial.

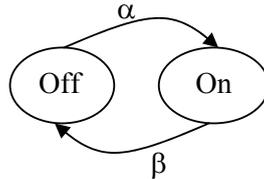


Figura 2.9: Modelo ON-OFF e IPP.

O modelo *IPP* (*Interrupted Poisson Process*) modifica o modelo *ON-OFF* de tal maneira que, quando o processo está no estado *ON*, a chegada dos pacotes será realizada de acordo com o processo de Poisson com taxa λ [Adas 1997].

Modelos baseados em regressão

O modelo *ARIMA* (*Auto Regressive Integrated Moving Average*) e o modelo *SARIMA* (*Seasonal ARIMA*) são estudados por Jenkins e Box em [Box et al. 1994]. Estes modelos se baseiam na dependência temporal do valor X_t em função dos elementos X_{t-k} , $k \in \mathbb{N}$ e foram construídos a partir da união do modelo $AR(p)$ e $MA(q)$. Métodos para estimar os parâmetros de tais modelos são descritos em [Box et al. 1994] e softwares de análise são disponíveis gratuitamente em [R Development Core Team 2005].

Seja \tilde{X}_t a diferença entre X_t e μ , $\tilde{X}_t = X_t - \mu$. O modelo $AR(p)$, ou auto-regressivo, é dado por $\tilde{X}_t = \phi_1 \tilde{X}_{t-1} + \dots + \phi_p \tilde{X}_{t-p} + a_t$, onde a_t é um ruído branco e ϕ_1, \dots, ϕ_p são parâmetros do modelo. Os parâmetros do modelo são fáceis de estimar e as séries temporais podem ser geradas de maneira simples. A função de auto-correlação decai exponencialmente, o que faz com que o modelo possa ser representado aproximadamente por um modelo de Markov Modulado [Adas 1997]. Este modelo não consegue caracterizar fluxos que possuam uma distribuição de cauda pesada.

Exemplo 2.4 *Seja um modelo $AR(2)$ com parâmetros $\phi_1 = 0,101270$ e $\phi_2 = -0,181100$. A equação do sistema será dada por $\tilde{X}_t = 0,101270\tilde{X}_{t-1} - 0,181100\tilde{X}_{t-2} + a_t$. Uma simulação numérica da série pode ser realizada através da geração de valores aleatórios para a_t seguindo a distribuição normal $N(0,1)$, como mostrado na Tabela 2.1. Para geração de a_t pode ser utilizado um dos métodos descritos em [Banks et al. 2001].*

Tabela 2.1: Exemplo de geração de uma série com modelo auto-regressivo.

t	\tilde{X}_t	a_t
1	-0,12212	-0,12212
2	-0,22040	-0,20804
3	-0,09140	-0,09120
4	1,30913	1,27847
5	0,97817	0,82904
\vdots	\vdots	\vdots

Tabela 2.2: Exemplo de geração de uma série com modelo de médias móveis.

t	\tilde{X}_t	a_t
1	1,14086	1,14086
2	-1,056751454	-0,93072
3	0,580769421	0,53614
4	0,20014546	0,21190
5	-1,389329616	-1,33857
\vdots	\vdots	\vdots

Para identificação do modelo AR em uma série existente o analista deve examinar a estrutura da função de auto-correlação da série e estimar o valor de p [Box et al. 1994]. Após esta tarefa, podem ser estimados os valores dos parâmetros $\phi_1, \phi_2, \dots, \phi_p$.

O modelo $MA(q)$, ou médias móveis, é dado por $\tilde{X}_t = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$, ou seja, o valor atual de \tilde{X}_t é formado pela soma dos choques ponderados de ruídos aleatórios passados. Os valores de $\theta_1, \dots, \theta_q$ são parâmetros do modelo.

Exemplo 2.5 *Seja um modelo $MA(2)$ com parâmetros $\theta_1 = 0,110472$ e $\theta_2 = 0,051004$. A equação do sistema será dada por $\tilde{X}_t = a_t - 0,110472a_{t-1} - 0,051004a_{t-2}$. Uma simulação numérica da série pode ser realizada através a geração de valores aleatórios para a_t seguindo a distribuição normal $N(0, 1)$, como mostrado na Tabela 2.2.*

Para estimação de parâmetros a partir de uma série, o analista deve estimar o valor do parâmetro q e depois estimar os valores de $\theta_1, \theta_2, \dots, \theta_q$.

No modelo $ARMA(p, q)$ os valores de p e q indicam respectivamente o número de parâmetros no modelo AR e MA. O modelo é dado por

$$\tilde{X}_t = \phi_1 \tilde{X}_{t-1} + \dots + \phi_p \tilde{X}_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2.14)$$

A estimação de parâmetros do modelos ARMA é mais difícil do que o modelo AR,

Tabela 2.3: Exemplo de geração de uma série com modelo ARIMA.

t	\tilde{X}_t	$\nabla^d \tilde{X}_t$	a_t
1	-2,27228	-2,27228	-2,27228
2	-2,26659	0,00569	-0,01522
3	-3,71876	-1,45217	-1,45212
4	-3,84008	-0,12132	-0,13468
5	-3,55374	0,28634	0,28523
\vdots	\vdots	\vdots	\vdots

e envolve a resolução de equações não-lineares [Box et al. 1994]. Na prática podem ser examinadas determinadas propriedades da função de auto-correlação e auto-correlação parcial da série na tentativa de determinar os valores de p e q . Neste modelo, soluções analíticas são mais difíceis de se obter. A geração da série pode ser realizada da mesma maneira mostrada nas Tabelas 2.1 e 2.2.

O modelo *ARIMA* (*Auto Regressive Integrated Moving Average*) consiste de uma extensão do modelo *ARMA*(p, q) e é dado por *ARIMA*(p, d, q) onde ∇^d é um operador de diferenças, definido como $\nabla^d X_t = (X_t - X_{t-d})$. A série original é submetida ao operador de diferenças e sobre a série diferenciada aplica-se um modelo ARMA.

No modelo ARIMA, o operador de diferenças pode ser obtido por

$$\nabla^d X_t = \sum_{i=0}^d \binom{d}{i} (-1)^i X_{t-i}, d \in \mathbb{N} \quad (2.15)$$

A diferenciação da série normalmente é aplicada na tentativa de torná-la estacionária. Um exemplo de diferenciação é mostrado na seção 2.3.4. Desta maneira, este modelo pode ser utilizado em séries não estacionárias. Uma descrição completa do modelo é encontrada em [Box et al. 1994] e softwares de análise em [R Development Core Team 2005].

Exemplo 2.6 Considere um modelo *ARIMA*(1,1,1), com parâmetros $\phi_1 = 0,101270$ e $\theta_1 = 0,110472$. O valor de $\nabla^d X_t$ será dado por $\{0,101270X_{t-1} + a_t\} + \{-0,110472a_{t-1}\}$. O valor de X_t pode ser obtido fazendo-se $X_t = \nabla^d X_t + X_{t-1}$. O resultado é mostrado na Tabela 2.3. A série e sua função de auto-correlação são mostrados na Figura 2.10. A função de auto-correlação mostra um decaimento lento, que pode ser confundido com a não estacionariedade da série ou mesmo com a presença de características auto-similares.

O modelo *SARMA* utiliza o mesmo conceito, acrescido de componentes sazonais considerando períodos de tempo de $t - T$, sendo T o período de repetição sazonal. Os

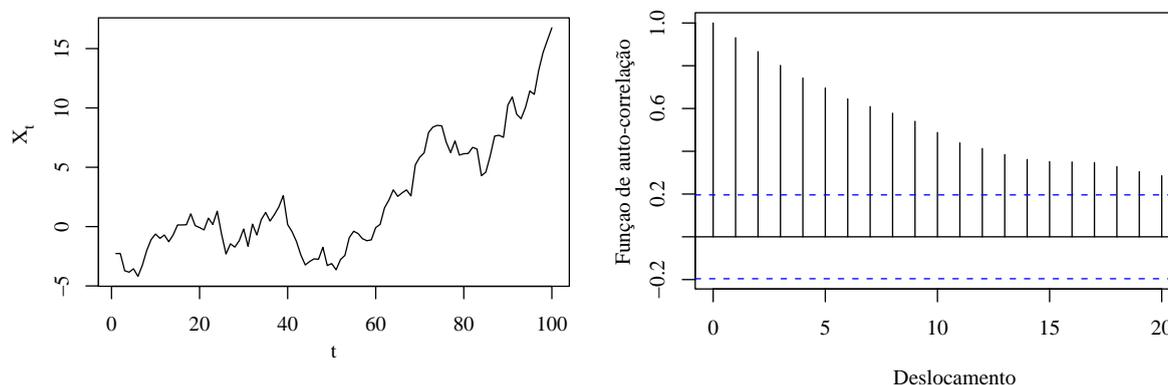


Figura 2.10: Série temporal gerada com o modelo ARIMA para o exemplo e respectiva função de auto-correlação.

parâmetros do modelo *SARMA* serão os valores de p e q (modelo ARMA não sazonal), o período sazonal T e valores de P e Q , que são respectivamente os parâmetros dos modelos AR e MA sazonais. O modelo *SARMA* descreve um processo estacionário e linear, e será dado por $SARMA(p, q)x(P, Q)T$.

Um exemplo com a caracterização de um sistema servidor de páginas utilizando-se os modelos ARIMA e SARIMA pode ser visto em [Pedroso e Fonseca 2004e].

2.4.2 Modelos com dependência de longa duração

Pode-se dividir os modelos de tráfego *LRD* em duas categorias: aqueles que procuram captar as características do tráfego sem tentar explicar porque ele possui características *LRD*, como por exemplo, os modelos *FARIMA* e *MMPP* discutidos mais adiante, e modelos que procuram capturar o comportamento através de uma aproximação do fenômeno gerador da *LRD*, como o modelo *ON-OFF* Pareto ou o modelo composto para chegada de conexões proposto em [Nuzman et al. 2002].

Fractional ARIMA (*FARIMA*)

O modelo *FARIMA* proposto por Hosking [Hosking 1981] em 1981 é uma extensão natural do processo *ARIMA* que permite valores reais para o grau de diferenciação (parâmetro d do modelo *ARIMA*). \tilde{X}_t é um processo estacionário e inversível¹ definido pela Equação 2.14 aplicado após a diferenciação utilizando-se a Equação 2.15 alterada

¹A propriedade que garante que X_t depende de seus valores passados com pesos que diminuem com a defasagem é chamada inversibilidade [Box et al. 1994].

com d representando um número real, $-1/2 < d < 1/2$. Neste modelo, X_t é um processo *LRD* se $0 < d < 1/2$ e *SRD* se $d = 0$.

Em um modelo *FARIMA* com *LRD*, o parâmetro de Hurst está relacionado com o parâmetro d através da relação $H = d + \frac{1}{2}$.

O operador de diferenças é redefinido como

$$\nabla^d X_t = \sum_{i=0}^{\infty} \binom{d}{i} (-1)^i X_{t-i}, \quad -1/2 < d < 1/2 \quad (2.16)$$

e o coeficiente binomial pode ser interpretado como

$$\binom{d}{i} (-1)^i = \frac{\Gamma(-d+i)}{\Gamma(-d)\Gamma(i+1)} \quad (2.17)$$

Para o ajuste dos parâmetros do modelo, determina-se o parâmetro de Hurst e encontra-se o valor de d .

A série pode ser gerada utilizando-se o algoritmo proposto em [Garrett e Willinger 1994], que possui grande complexidade computacional (da ordem de $o(n^2)$).

Devido a sua origem na análise de séries temporais, o modelo *FARIMA* pode sintetizar tráfego para simulações e ser utilizado na previsão de tráfego. Detalhes sobre a caracterização do tráfego e estimação de parâmetros podem ser encontrados em [Xue e Lee 1999b] e [Liu et al. 1999]. Um modelo para análise de filas com entrada auto-similar pode ser encontrada em [Xue e Lee 1999a].

Movimento Browniano Fracionário

O termo movimento Browniano² refere-se a:

1. O fenômeno físico em que minúsculas partículas imersas em um fluido movem-se randomicamente; ou
2. Os modelos matemáticos utilizados para descrever tais movimentos randômicos.

Matematicamente, o movimento Browniano pode ser caracterizado por um processo de *Wiener*, que é um processo contínuo no tempo com incrementos independentes. Para cada valor positivo de t , denote o valor do processo por W_t . O processo será caracterizado pelas duas condições seguintes:

²Em homenagem ao botânico Robert Brown

1. Para $0 < s < t$, $W_t - W_s \sim N(0, t - s)$. $N(\mu, \sigma^2)$ denota a distribuição normal com média μ e variância σ^2 ;
2. Para $0 \leq s < t \leq u < v$, ou seja, dois intervalos $[s, t]$ e $[u, v]$ não sobrepostos, $W_t - W_s$ e $W_v - W_u$ são variáveis aleatórias independentes.

Estas propriedades estabelecem que o movimento Browniano é um processo Markoviano. O movimento Browniano é relacionado ao problema “passeio aleatório” (*random walk*) e é genérico no sentido que muitos processos físicos podem ser caracterizados por este modelo.

No movimento Browniano fracionário (*fBm*), os incrementos também são normalmente distribuídos – mas não independentes. Para um número H com $0 < H < 1$ em um intervalo de tempo $[s, t]$, $0 < s < t$, o valor do processo $W_t - W_s \sim N(0, (t - s)^H)$. Seja o valor $\delta = t - s$ a escala de observação. O incremento do processo estará relacionado em todas as escalas. Este modelo leva a um processo auto-similar, com dependência de longo prazo.

O modelo *fBm* tem sido utilizado para estudar o tráfego em redes de dados digitais. Uma variação proposta é o movimento Browniano multi-fracionário (*mBm*), onde ao invés de utilizar-se um número H se utiliza uma função contínua no tempo $H(t)$, chamada função de Hölder, e tem merecido atenção dos pesquisadores (ver [Véhel e Riedi 1997] e [Melo e Fonseca 2005]). No entanto, Darryl Veitch, et al., em [Veitch et al. 2005a] questionam a qualidade das ferramentas matemáticas disponíveis atualmente para a análise das propriedades multi-fractais do tráfego.

Markov Modulated Poisson Process (MMPP)

O modelo *MMPP* é considerado o melhor processo de Markov a emular *LRD* [Muscarello et al. 2004][Andersen e Nielsen 1998] e a invariância de escala [Horvath e Telek 2002]. Neste modelo, sempre é possível encontrar uma escala sobre a qual a correlação do processo *MMPP* decai exponencialmente ao invés do longo decaimento esperado em um processo *LRD*, logo ele não pode ser considerado um processo perfeitamente auto-similar do ponto de vista da definição.

Os autores deste modelo definem um parâmetro de Hurst local que é válido em um conjunto limitado de escalas de tempo.

O modelo apresentado em [Muscarello et al. 2004] sugere a aplicação de 3 camadas envolvidas na geração de tráfego: sessões, geradas por um processo externo (usuário); fluxos, gerados dentro de sessões; e pacotes, que são gerados dentro de fluxos. A aplicação

de 3 camadas apresenta bons resultados segundo [Gong et al. 2005], mas existe a possibilidade de aumentar o número de níveis. Na seqüência será considerado o modelo de 3 camadas.

As sessões chegam ao sistema de acordo com um processo de Poisson. Os fluxos dentro de sessões também chegam de acordo com um processo de Poisson, bem como a chegada de pacotes dentro de um fluxo. O número médio de fluxos por sessão e o número médio de pacotes por fluxo seguem uma distribuição geométrica, de modo que todo o sistema pode ser modelado por uma cadeia de Markov contínua no tempo.

O modelo necessita de 5 parâmetros:

λ_s : ritmo de chegada de novas sessões;

λ_f : ritmo de chegada de fluxos por sessão ativa;

λ_p : ritmo de chegada de pacotes por sessão ativa;

N_f : número médio de fluxos por sessão;

N_p : número médio de pacotes por fluxo.

A relação entre o parâmetro de Hurst e os parâmetros foi estabelecida empiricamente, com ênfase na dependência entre o valor de H e o parâmetro N_f . Não existe nenhum mapeamento direto entre os parâmetros do modelo e o parâmetro de Hurst. No entanto, um algoritmo para ajustar o modelo a um conjunto de dados real foi proposto em [Muscariello et al. 2004].

A grande vantagem deste modelo é a sua simplicidade e a maior facilidade para estabelecer resultados analíticos em comparação com os modelos *LRD* “puros”.

Modelo Markoviano Hierárquico ON-OFF

Em [Gong et al. 2005] é proposto um modelo para caracterizar o tráfego produzido por aplicações *TCP*. O modelo proposto consiste em n camadas, como mostrado na Figura 2.11, onde a transição de estados é um processo Markoviano. Somente no estado 111 o sistema estará gerando tráfego. Este modelo também pode ser explicado através de uma analogia *sessão-fluxo-pacote*. Este modelo foi estudado anteriormente por [Nuzman et al. 2002].

A aplicação do modelo para geração de tráfego sintético levou a uma boa aproximação do tráfego observado na rede, com $n = 3$, inclusive com características auto-similares dentro de certos limites de escala.

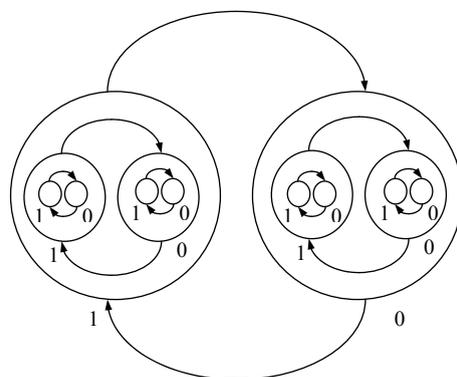


Figura 2.11: Modelo Hierárquico *ON-OFF* de 3 níveis.

O modelo *Markov Modulated Poisson Process* descrito no item 2.4.2 é um caso específico deste modelo com $n = 5$.

Modelo ON-OFF com distribuição de cauda longa

Neste modelo, cada fluxo tem um comportamento *ON-OFF*, onde o tempo que o fluxo permanece *ON* e *OFF* possui uma distribuição de cauda pesada, como a distribuição de Pareto [Willinger e Park 2000]. O tráfego agregado resultante da soma de diversos fluxos apresenta comportamento assintoticamente auto-similar e *LRD*. Este modelo tem sido freqüentemente utilizado para produção de tráfego em simulações.

Modelo SURGE (*Scalable URL Reference Generator*)

Modelos de desempenho para servidores Web podem utilizar ou não características de comportamento do usuário. Modelos baseados na teoria de filas, com o apresentado por [Cao et al. 2003] (modelo $M/G/1/K^*PS$) não consideram o comportamento do usuário nem o conteúdo do servidor. Por outro lado, no modelo *SURGE* proposto por [Barford e Crovella 1998] o conhecimento do comportamento do usuário e do conteúdo do servidor são utilizados para atingir uma melhor aproximação do tráfego real.

O modelo *SURGE* para geração de tráfego é baseado em um autômato *ON-OFF* que captura o comportamento do usuário. Quando o sistema está no estado *ON*, a sessão está ativa enviando os objetos requisitados na sessão. O intervalo de tempo entre os arquivos enviados durante a sessão é denominado de tempo *active-off*. O tamanho dos arquivos e o número de referências em uma sessão de usuário também é utilizada. As principais variáveis do modelo são:

- *Tempo OFF*: É o tempo que o usuário permanece pensando. Normalmente modelado por uma distribuição de Pareto;
- *Tamanho dos arquivos*: É o tamanho dos objetos transmitidos. Normalmente modelado por uma distribuição de Pareto;
- *Número de referências*: Número de arquivos transmitidos em uma sessão de usuário. Também modelado normalmente por uma distribuição de Pareto;
- *Tempo active-off*: É o intervalo de tempo entre os arquivos transmitidos em uma sessão de usuário. Modelada pela distribuição de Weibull;
- *Popularidade*: É o número relativo de acessos realizados a um arquivo individual. A popularidade de arquivos em servidores Web segue, via de regra, a lei de Zipf. A lei de Zipf argumenta que se os arquivos forem ordenados do mais popular para o menos popular, então o número de referências a um arquivo P tende a ser inversamente proporcional à sua posição na classificação r , ou $P = kr^{-1}$ para uma constante positiva qualquer k ;
- *Localidade temporal*: A localidade temporal assume que, uma vez tendo sido requisitado um arquivo, a probabilidade de que ele seja novamente requisitado no futuro aumenta. Para o estudo desta variável os acessos são armazenados em uma estrutura de pilha. A distância entre os acessos nesta pilha são estudados e modelados comumente com uma distribuição lognormal.

2.5 Conclusão do capítulo

Os modelos de tráfego mostrados neste capítulo são utilizados para prever o comportamento de redes de alta velocidade em diversos aspectos. Estes modelos podem ser utilizados em estudos analíticos ou mesmo através de simulações computacionais. A simulação computacional tem sido muito utilizada quando o sistema possui grande complexidade e resultados analíticos são muito difíceis de obter.

A utilização do modelo auto-similar para representar o tráfego em redes de telecomunicações é freqüente nas publicações científicas atualmente. No entanto, em vários casos o tráfego não apresenta tais características e a utilização dos modelos clássicos é mais adequada.

Para realizar a avaliação de desempenho de uma rede, o engenheiro deverá estar preparado para identificar as características de tráfego de modo a utilizar o modelo que descreve apropriadamente o sistema real.

Quando o modelo auto-similar puder ser utilizado, a compreensão dos aspectos do modelo (como a presença de rajadas de tráfego) será fundamental para realizar, entre outras, a análise de desempenho, o desenvolvimento de algoritmos de controle de congestionamento mais eficientes ou realizar dimensionamento de sistemas.

Capítulo 3

Um novo modelo para geração de tráfego *Web*

3.1 Introdução

O desenvolvimento de modelos que permitam representar o desempenho de servidores *Web* é uma importante área de pesquisa. Bons modelos permitem realizar previsões acuradas sobre métricas de desempenho [Cao et al. 2003] e, a partir destas, por exemplo, o planejamento da capacidade do servidor. Entre as características desejadas para um modelo estão a sua simplicidade, obtida ao se restringir somente aos aspectos que influenciem significativamente no comportamento que se deseja analisar, e a sua tratabilidade, geralmente associada à complexidade de se gerar resultados analisáveis a partir do modelo.

A carga de tráfego da Internet é dominada pelo protocolo *HTTP* (Hyper Text Transfer Protocol [Fielding et al. 1999]), que segundo [Cao et al. 2004a], foi identificado em medições do backbone da Sprint como sendo a classe de aplicativos mais utilizado na Internet, com uma faixa de 31% a 59% do total de bytes transmitidos. Nos últimos 10 anos, e possivelmente no futuro, geradores de tráfego para sintetizar o tráfego *Web* têm sido um componente essencial em virtualmente toda a simulação da Internet [Cao et al. 2004a]. Adicionalmente, o protocolo *HTTP* está sendo utilizado como interface para entregar conteúdos para aplicativos mais especializados de alto nível.

Em se tratando de planejamento de capacidade de servidores *Web*, os modelos denominados de *modelos de desempenho* devem ser suficientemente precisos para capturar o comportamento real do servidor, pois modelos inadequados poderiam resultar em super ou sub dimensionamento da capacidade do servidor.

Nos últimos anos, muito esforço tem sido realizado para desenvolver modelos que des-

crevem apropriadamente o tráfego de redes de computadores [Crovella e Bestavros 1995], [Leland et al. 1994] e [Willinger e Park 2000]. O resultado destes trabalhos mostram que o modelo auto-similar descreve apropriadamente o tráfego agregado observado na saída de servidores *Web*. Um resultado importante é o fato que os modelos utilizados anteriormente, como os modelos Markovianos, não têm apresentado resultados satisfatórios para a modelagem do tráfego em redes de alta velocidade.

Um modelo de tráfego para servidores *Web* pode utilizar características de comportamento do usuário ou não. Por exemplo, modelos baseados na teoria de filas, como o proposto por [Cao et al. 2003] (modelo M/G/1/K*PS) não considera o comportamento do usuário nem o conteúdo do servidor. Por outro lado, no modelo proposto por [Nuzman et al. 2002] ou por [Muscariello et al. 2004] o conhecimento do comportamento do usuário é utilizado em associação com o mecanismo de produção de fluxos para atingir uma melhor representação do tráfego real.

Neste capítulo é proposto um modelo de desempenho para servidores *Web* utilizando a classificação semântica do conteúdo transmitido pelo servidor e o comportamento do usuário, sendo derivado do SURGE (*Scalable URL Reference Generator*) [Barford e Crovella 1998]. O modelo aqui proposto baseia-se no fato observado durante o desenvolvimento desta tese de que o tamanho dos objetos de cada classe semântica não segue, via de regra, distribuições de cauda pesada e sim distribuições de decaimento exponencial ou sub-exponencial. Com isto, evitam-se as características pouco desejáveis para análise exibidas por distribuições de cauda pesada, como a não convergência do desvio padrão. Assim sendo, o uso da classificação semântica permite estabelecer um modelo mais tratável para avaliação do desempenho de servidores *Web*.

A nova abordagem de modelagem aqui proposta contribui em 3 pontos: geração de carga sintética para simulações computacionais envolvendo servidores *Web*, possibilidade de análise de desempenho do servidor através de cadeias de Markov e possibilidade de pesquisa de novas técnicas relacionadas a melhoria de desempenho de servidores (por exemplo, algoritmos de descarte de dados em cache).

A validação do modelo foi realizada a partir da análise estatística de dados de vários servidores que possuem seus arquivos de *log* disponíveis para uso em pesquisa. Foram analisados os arquivos de *log* do servidor da copa do mundo de 1998 [Arlitt e Jin 2000] e de servidores de Proxy do *IRCache*, que é um projeto do *NLANR* - National Laboratory for Applied Network Research [Wessels e Claffy 1996]. Os servidores cache do *NLANR* registram os acessos de toda a Internet e estão geograficamente distribuídos pelo Estados Unidos da América para balanceamento de carga e de forma a atender todos os continentes.

Desta forma, as milhares de requisições diárias destes servidores refletem o comportamento típico de acessos a servidores *Web* em geral, pois não se restringem a um grupo restrito de clientes ou a aplicações específicas.

3.2 Definições iniciais e conceitos básicos

Definição 3.1 *Um fluxo é definido como um grupo de pacotes com um conjunto de atributos comuns, tais como [porta origem, porta destino, endereço origem, endereço destino].*

Definição 3.2 *Uma sessão identifica um fluxo particular que é utilizado na transferência de arquivos entre o cliente e o servidor *Web*. Nesta tese, será utilizado o conceito de sessão para representar uma conexão persistente para transferência de arquivos entre o cliente e o servidor *Web*.*

Definição 3.3 *O termo objeto será utilizado como sinônimo de arquivo transferido entre o servidor *Web* e seus clientes.*

Definição 3.4 *Foi adotada a definição de justiça adaptada de [Jain 1991]. Para um conjunto representando o consumo de banda por sessões (x_1, x_2, \dots, x_n) , a seguinte função pode ser utilizada como índice de justiça para este conjunto: $f(x_1, x_2, \dots, x_n) = (\sum_{i=1}^n x_i)^2 / n \sum_{i=1}^n x_i^2$. Se todos os usuários receberem igual divisão de banda, então o índice de justiça será 1. Como este índice de justiça está entre 0 e 1, a métrica oferece um valor intuitivo para determinação da justiça.*

3.3 Trabalhos Relacionados

Os geradores de tráfego *Web* em uso atualmente são normalmente baseados nos dados de três projetos pioneiros que procuravam capturar características comportamentais de clientes e servidores: os estudos de [Mah 1997], [Crovella e Bestavros 1995] e [Barford e Crovella 1998]. Geradores de tráfego utilizando os conceitos propostos foram construídos como parte do simulador *NS-2* [Breslau et al. 2000], amplamente utilizado nesta área.

Em [Hernandez-Campos et al. 2003] é apresentado um modelo empírico do tráfego gerado pelo protocolo *HTTP*. Ao invés de confiar nos *logs* de servidores ou clientes, seu método é baseado nos pacotes coletados durante a conversação *HTTP*. Através da análise

de tráfego, Hernández-Campos et al. determinaram estatísticas e distribuições para elementos de alto nível, como tamanho de arquivos transferidos, o número de arquivos por sessão e o comportamento do usuário. Estas quantidades formam um modelo que pode ser utilizado em simulações para imitar o tráfego gerado por aplicações.

Barford et al., em [Sommers e Barford 2004] apresentam uma ferramenta chamada Harpoon, projetada para geração de pacotes no nível de fluxo *IP*. A ferramenta gera pacotes *TCP* e *UDP* que possuem as mesmas características medidas em roteadores reais em termos de byte, pacote, temporal e espacial. A ferramenta se distingue de outras pela sua capacidade de auto configuração a partir de arquivos de *log*. A análise se diferencia de outras tentativas de montagem de modelos utilizando o conceito de fluxos por combinar distribuições empíricas de características que podem ser medidas em roteadores de redes reais. O modelo propõe a geração de fluxos baseada em um nível de duas hierarquias. Sessões são formadas por uma série de conexões separadas por um tempo de duração. Conexões são formadas pela transferência de objetos com intervalo de tempo caracterizado por distribuições de probabilidade. O tamanho dos arquivos transferidos possui uma distribuição de cauda pesada e um modelo ON/OFF gera características auto-similares ao nível de pacotes.

Em [Cao et al. 2004a] os autores apresentam um modelo que expressa o tráfego *Web* como uma coleção de conexões *TCP* independentes, cada qual caracterizada por valores das seguintes variáveis: tempo entre conexões, *RTT* (*Round Trip Time*) para o cliente, *RTT* para o servidor, número de trocas requisição/resposta, intervalo de tempo entre trocas, tamanho de requisições individuais, tamanho de respostas individuais e atraso do servidor.

O modelo proposto neste capítulo é um modelo que captura características comportamentais do sistema através da classificação dos diversos tipos de objetos transmitidos pelo servidor. Este modelo amplia as possibilidades de análise e geração de tráfego, não sendo necessário o uso de distribuições de cauda pesada ou de duas ou mais distribuições de probabilidade distintas para caracterizar o tamanho dos objetos transmitidos, como necessário no modelo *SURGE*. Se comparado ao modelo *SURGE*, o modelo aqui proposto possui ainda a vantagem de ser mais claro na separação dos arquivos para a caracterização das distribuições de probabilidade, possivelmente levando a uma representação mais fiel do sistema.

3.4 Um modelo composto para servidores *Web* utilizando classificação semântica de conteúdo

O modelo proposto é baseado no modelo *SURGE* proposto por Crovella e Barford em [Barford e Crovella 1998], um dos modelos mais citados para geração de tráfego *Web*. O modelo *SURGE* foi descrito no item 2.4.2.

A novidade proposta será classificar os arquivos transmitidos pelo servidor em classes para que depois as classes sejam estudadas para determinar a distribuição de probabilidade do tamanho dos arquivos de cada classe.

Trabalhos importantes reportam que o tamanho dos arquivos transmitidos por servidores *Web* podem ser modelados por distribuições de cauda pesada, por exemplo, a distribuição de Pareto [Crovella e Bestavros 1995]. Existe grande dificuldade no tratamento matemático deste tipo de distribuição devido a sua grande variabilidade [Willinger e Park 2000].

No modelo proposto, a atividade do usuário será modelada por um autômato finito como no *SURGE*. É razoável supor que a chegada de novas sessões de usuários constitua-se de um processo de Poisson [Roberts 2000]. O estado *ON* será considerado como o tempo gasto em uma sessão ativa. Durante o tempo de uma sessão ativa serão produzidas requisições para transferência de diversos arquivos. Os arquivos transmitidos serão classificados em classes semânticas previamente identificadas com auxílio dos arquivos de *log* do servidor.

As demais variáveis do *SURGE*, como a popularidade, a localidade temporal e o tempo *active-off*, continuam válidas.

Desta maneira, a cada sessão ativa de usuário, após a primeira requisição ser produzida, uma seqüência de arquivos será transmitida. Estas várias requisições serão produzidas pelo próprio programa cliente (*browser*) automaticamente para transferir todos os arquivos necessários para apresentar o conteúdo para o usuário. A Figura 3.1 ilustra um diagrama de estados hipotético utilizando uma classificação com três classes de arquivo:

- Arquivos em formato de hipertexto (*HTML*);
- Imagens em formato *GIF* (*IMG-GIF*);
- Notas de aula em formato *PDF* (*LN-PDF*).

A Figura 3.1 deve ser interpretada da seguinte maneira. Um acesso inicial a um arquivo pertencente à classe *HTML* será seguida por uma transmissão de um novo arquivo

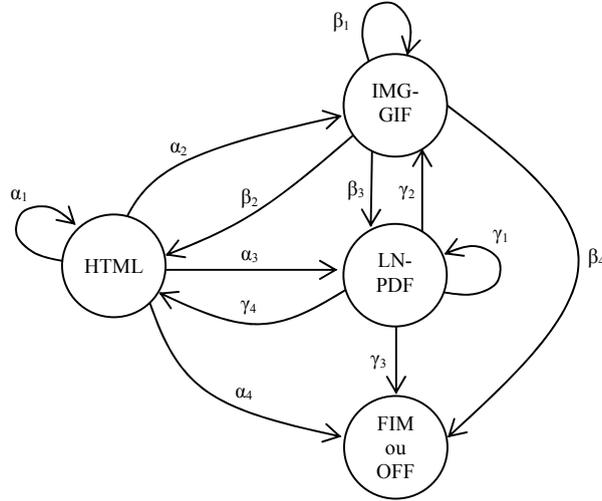


Figura 3.1: Diagrama hipotético de classes semânticas para transmissão de objetos durante o estado *ON*.

pertencente a outra classe de tráfego. No caso específico deste exemplo, α_1 , α_2 e α_3 representam as probabilidades do próximo arquivo transmitido pertencer respectivamente às classes *HTML*, *IMG-GIF* ou *LN-PDF*. α_4 é a probabilidade de não haver mais requisições de transmissão de arquivos para esta sessão, iniciando-se um período *OFF*. A relação $\sum_{j=1}^i \alpha_j = 1$ deve ser sempre verdadeira. O mesmo ocorre para as probabilidades de transição a partir de cada uma das classes de arquivos do modelo. No exemplo, $\sum_{j=1}^i \beta_j = 1$ e $\sum_{j=1}^i \gamma_j = 1$ para as classes *IMG-GIF* e *LN-PDF*.

De modo genérico, pode-se representar o diagrama de estados da Figura 3.1 através de uma matriz quadrada P com as probabilidades de transição de estados, dada por

$$P = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0(i-1)} \\ p_{10} & p_{11} & \cdots & p_{1(i-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (3.1)$$

onde i representa o número de classes de arquivos, incluindo-se o estado representando o fim da sessão. Na montagem da matriz, o estado $i - 1$ é o estado que representa o fim da sessão. Para um estado k , $\sum_{j=0}^{i-1} P_{kj} = 1$.

Este diagrama de transição de estados representa a sucessão de classes de arquivos transmitidos, e não o tamanho dos arquivos transmitidos em cada estado. O tamanho dos arquivos transmitidos em cada estado deve ser determinado através da análise dos arquivos do servidor. A determinação da distribuição de probabilidade de cada classe de

arquivos tipicamente encontrada na Internet é mostrada na Seção 3.5.

A análise do arquivo de *log* irá revelar a quantidade de classes de arquivos do servidor. A montagem da classe pode facilitar a tarefa de modelagem do tamanho dos objetos, uma vez que a classificação pode ser realizada de acordo com diversos critérios. Neste trabalho, a classificação se fez apenas através da extensão do arquivo.

Uma matriz de probabilidade de transições entre as classes de arquivos para uma dada sessão também pode ser extraída dos arquivos de *log*, de modo a completar o modelo.

3.5 Levantamento das classes de arquivos e da matriz de transição de estados

Utilizando uma heurística semelhante àquela desenvolvida originalmente por [Mah 1997] e por [Barford e Crovella 1998], foram identificadas as diversas sessões de clientes nos servidores *Web* em estudo. Para identificar as sessões foi produzido um software que identifica o primeiro acesso de um determinado cliente e busca pelas transições de requisições de classes de arquivos, apresentando como resultado a matriz de probabilidade de transição de classe de arquivos. O programa considerou um tempo limite de 120 segundos para concluir sobre a inatividade da sessão. Utilizamos o conceito de sessão para definir o período de tempo onde um cliente transfere uma página e em seguida outros arquivos referenciados por esta página, que podem ser chamados de objetos embutidos. Também será utilizado o termo "objeto" como sinônimo da resposta enviada por um servidor *Web*.

Os resultados aqui apresentados foram obtidos através da análise de coletas de dois sistemas. O primeiro sistema em estudo é o *IRCache*, que é um projeto do *NLANR - National Laboratory for Applied Network Research* [Wessels e Claffy 1996]. Os servidores cache do *NLANR* registram os acessos de toda a Internet e estão geograficamente distribuídos pelo Estados Unidos da América para balanceamento de carga e de forma a atender todos os continentes. Desta forma, as milhares de requisições diárias destes servidores refletem o comportamento típico de acessos a servidores *Web* em geral, pois não se restringem a um grupo restrito de clientes ou a aplicações específicas.

O segundo sistema em estudo foi o servidor *Web* da Copa do Mundo de 1998, já estudado na literatura [Arlitt e Jin 2000]. Apesar de ser um sistema já relativamente antigo, o fato de já existirem estudos publicados sobre este sistema possibilita que os resultados obtidos sejam confrontados com outros estudos. Foram estudados 4 dias tota-

Tabela 3.1: Coleção de dados em estudo.

Amostra	Local	Data	Número de linhas
IRCache	New York	28/Novembro/2004	366.234
IRCache	New York	29/Novembro/2004	20.530
IRCache	Palo Alto	29/Novembro/2004	140.636
WC98	Day 37	31/Maio/1998	558.6176
WC98	Day 52	15/Junho/1998	7.000.000
WC98	Day 73	6/Julho/1998	7.000.000

lizando aproximadamente 20 milhões de requisições. Os arquivos analisados no trabalho são apresentados na Tabela 3.1.

3.5.1 Formato dos arquivos de *log*

Os arquivos de registro do histórico de acessos do sistema IRCache são compostos por registros, sendo um por linha, e cada linha com 10 campos. O significado de cada linha é descrito na Tabela 3.2.

Os registros foram separados pelo campo 10 (tipo do arquivo servido) e para cada um individualmente armazenados todos os valores do campo 5 (tamanho da resposta enviada ao cliente em bytes). Os arquivos resultantes foram utilizados como amostras para a caracterização das distribuições estatísticas.

Nesta análise o campo 10 é denominado como classe de arquivo (exemplo: *HTTP*, *GIF*, etc).

Tabela 3.2: Formato dos registros do histórico de acessos do sistema IRCache.

Campo	Significado
1	Instante de tempo em que o servidor encerrou a conexão <i>TCP</i>
2	Tempo decorrido entre o início da conexão e o encerramento
3	Endereço <i>IP</i> do cliente
4	Código de resposta <i>HTTP</i>
5	Tamanho da resposta enviada ao cliente em bytes
6	Método de requisição <i>HTTP</i>
7	<i>URL</i> requisitada
8	Identificação do usuário (sempre “-” nos <i>logs</i> da <i>IRCache</i>)
9	Hierarquia e endereço do servidor
10	Tipo do arquivo servido

3.5.2 Identificação das principais classes de tráfego

A observação dos *logs* mostrou as principais classes de tráfego em cada um dos sistemas em estudo. A extração das classes de objetos transmitidos foi realizada observando-se principalmente a extensão dos arquivos transmitidos. Para identificação da contribuição da classe foi levantado o volume de tráfego efetivamente transmitido em cada classe de tráfego.

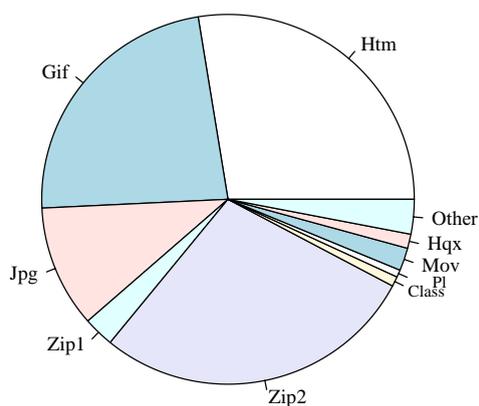


Figura 3.2: Volume de tráfego para as principais classes de arquivo transmitidos identificadas nos arquivos de *log* do servidor da WC98.

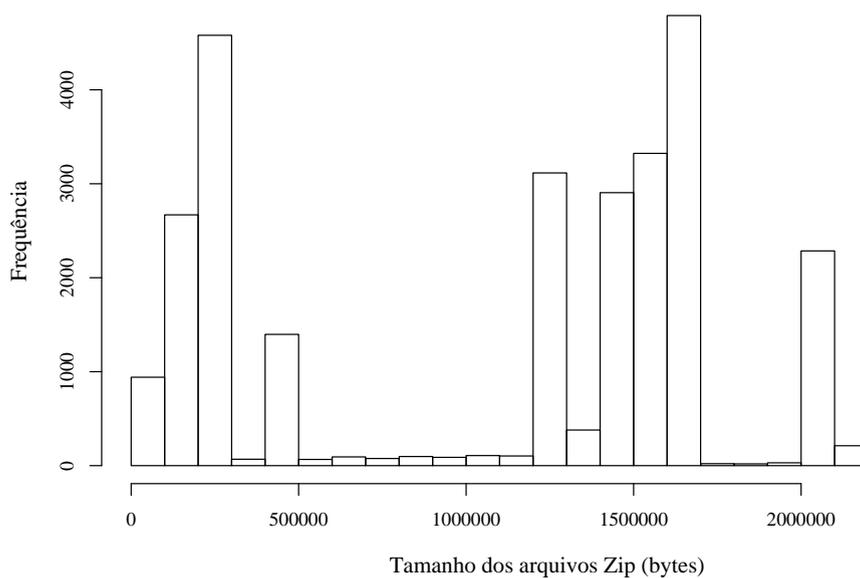


Figura 3.3: Histograma do tamanho dos arquivos da classe *Zip* do servidor da Copa do Mundo de 1998.

No servidor da Copa do Mundo de 1998 verifica-se a existência de poucas classes de arquivos que contribuíram significativamente na formação do tráfego de saída do servidor, o que permite a construção de um modelo bastante simplificado. Os tipos mais importantes foram *HTML*, *JPEG*, *GIF*, *ZIP1* e *ZIP2*, como mostrado na Figura 3.2. Os arquivos com extensão *ZIP* foram divididos em duas categorias, *ZIP1* e *ZIP2*, porque a observação do histograma da classe *ZIP* mostrou dois comportamentos distintos (ver Figura 3.3). A classe *ZIP* foi separada em duas classes de acordo com o tamanho do arquivo transmitido. Arquivos maiores que 10^6 bytes foram classificados como *ZIP2* e os outros como *ZIP1*. Isto é necessário para evitar o uso de distribuições de cauda pesada para caracterizar a classe *ZIP*, o tamanho de arquivo resultante nas classes *ZIP1* e *ZIP2* puderam ser caracterizados pela distribuição Lognormal. O volume de tráfego por tipo de arquivo está de acordo com o reportado em [Arlitt e Jin 2000].

Tabela 3.3: Percentual observado em relação ao volume total trafegado em bytes para as principais classes de arquivos transmitidos no sistema *IRCache*.

Classe de arquivo	Amostra 1	Amostra 2	Amostra 3	Média
GIF1	0,93	1,26	0,96	1,05
GIF2	4,91	6,60	5,02	5,51
HTML	22,11	15,71	20,94	19,59
JPEG	20,88	25,86	10,25	19,00
MPEG	4,22	3,7	3,11	3,68
OCTET-STREAM	13,85	28,07	20,15	20,69
OUTROS	33,1	18,8	39,57	30,49

No sistema *IRCache* foram identificadas as seguintes classes principais: *GIF*, *HTML*, *JPG*, *MPEG*, *OCTET-STREAM*. A participação de cada classe na formação do tráfego pode ser vista na Tabela 3.3. A classe *GIF* foi separada em duas classes para a realização do teste de aderência à distribuições conhecidas. O critério utilizado foi o mesmo do caso anterior com a classe *ZIP* do servidor *WC98*. Neste caso, os arquivos *GIF* com tamanho menor que 350 bytes foram classificados como *GIF1* e os arquivos com tamanho maior ou igual a 350 bytes foram classificados como *GIF2*.

3.5.3 Identificação da matriz de probabilidade de transição de estados das classes de arquivos

A Tabela 3.4 mostra a matriz de probabilidade de transição de classes obtidos a partir dos arquivos de *log* dos dias 27, 52 e 73 do servidor *Web* da copa do mundo de

Tabela 3.4: Matriz de probabilidade transição de estados para o servidor da WC98.

	HTML	JPEG	GIF	ZIP1	ZIP2	MOV	HQX	CLASS	PL	OUTR	FIM
HTML	0,3105	0,0881	0,5244	0,0003	0,0002	0,0000	0,0000	0,0045	0,0002	0,0219	0,0495
JPEG	0,1279	0,3145	0,5075	0,0014	0,0002	0,0003	0,0000	0,0053	0,0002	0,0089	0,0333
GIF	0,0657	0,0346	0,8634	0,0001	0,0002	0,0000	0,0000	0,0110	0,0002	0,0064	0,0180
ZIP1	0,1539	0,0258	0,1658	0,2601	0,0159	0,0000	0,0001	0,0008	0,0025	0,0117	0,3630
ZIP2	0,0908	0,0122	0,1142	0,0091	0,1323	0,0000	0,0025	0,0008	0,0039	0,0039	0,6298
MOV	0,1234	0,1428	0,1076	0,0014	0,0014	0,3094	0,0000	0,0000	0,0000	0,0093	0,3043
HQX	0,0815	0,0163	0,1068	0,0036	0,0724	0,0000	0,1757	0,0000	0,0054	0,0126	0,5253
CLASS	0,0661	0,0202	0,8622	0,0000	0,0000	0,0000	0,0000	0,0217	0,0000	0,0042	0,0251
PL	0,0794	0,0451	0,3973	0,0008	0,0023	0,0000	0,0000	0,0002	0,2850	0,0067	0,1828
OUTROS	0,1830	0,0509	0,6424	0,0005	0,0003	0,0000	0,0000	0,0082	0,0002	0,0380	0,0761

1998. Estes dados foram extraídos utilizando uma heurística semelhante à utilizada por Mah em [Mah 1997], Barford e Crovella em [Barford e Crovella 1998] e Hernandez et al. em [Hernandez-Campos et al. 2003], onde uma sessão do cliente é identificada a partir do primeiro acesso de um endereço *IP* ao servidor que transfere diversos arquivos refletindo o comportamento do usuário. Quando não há mais atividade do usuário dentro de um certo tempo limite, a sessão é considerada encerrada. O tempo limite utilizado neste trabalho foi de 120 segundos. Os dados foram recolhidos de modo a oferecer um estimador para a Equação 3.1

Outro fato relevante a ser observado é que a matriz de probabilidade de transição entre as classes de arquivo permanece constante ao longo dos três dias observados, o que revela que esta matriz descreve o comportamento do acesso aos arquivos do servidor em cada sessão de usuário, e pode ser utilizada para avaliação de desempenho e geração de carga.

O resultado obtido para o sistema *IRCache* não é apresentado por este sistema tratar-se de um *Proxy* que agrupa requisições a diversos servidores *Web*. Neste caso, a caracterização da matriz de probabilidade de transição de estados não estaria relacionado a nenhum servidor *Web* em particular, e sim a características mais gerais de sistemas *Web*.

3.5.4 Caracterização do tamanho dos objetos transmitidos em cada classe de arquivo

Foi realizada a caracterização do tamanho do arquivo transmitido nas principais classes identificadas na Seção 3.5.2.

A análise dos objetos transmitidos pelo sistema *IRCache* mostrou que o tamanho do arquivo transmitido em cada classe pode ser modelado na maioria dos casos pela

distribuição Lognormal. Para realizar o teste de aderência, a distribuição de probabilidade acumulada da amostra foi comparada com distribuições clássicas de probabilidade. A Figura 3.4 mostra uma comparação entre a distribuição acumulada teórica e a distribuição amostral para as principais classes. O eixo horizontal x representa o tamanho do arquivo transmitido e o eixo vertical $P(X \leq x)$ indica a probabilidade acumulada. Todas as classes de tráfego aderiram à distribuição Lognormal, à exceção da classe *GIF1*, que aderiu à distribuição Normal. Um sumário com as distribuições ajustadas e estatísticas básicas é apresentado na Tabela 3.5.

Tabela 3.5: Sumário de estatísticas e ajuste de distribuição para os arquivos pertencentes às principais classes de arquivos do sistema *IRCache*.

Classe	Modelo	Parâmetros
GIF1	Normal	$\mu = 281, \sigma = 23$
GIF2	Lognormal	$\mu = 4.255, \sigma = 24506$
HTML	Lognormal	$\mu = 8.052, \sigma = 19.971$
JPEG	Lognormal	$\mu = 4.434, \sigma = 4.548$
MPEG	Lognormal	$\mu = 388.000, \sigma = 643.102$
OCTET-STREAM	Lognormal	$\mu = 45.970, \sigma = 1.159.506$

Tabela 3.6: Sumário de estatísticas e ajuste de distribuição para os arquivos pertencentes às principais classes de arquivos da *WC98*.

Classe	Modelo	Parâmetros
GIF	Lognormal	$\mu = 1.646, \sigma = 4.100$
HTML	Lognormal	$\mu = 13.550, \sigma = 16.284$
JPEG	Lognormal	$\mu = 10.210, \sigma = 8.142$
ZIP1	Lognormal	$\mu = 254.100, \sigma = 159.773$
ZIP2	Lognormal	$\mu = 1.564.000, \sigma = 233.676$
MOV	Lognormal	$\mu = 1.441.000, \sigma = 285.251$
PL	Lognormal	$\mu = 64.720, \sigma = 212.349$
HQX	Lognormal	$\mu = 2.236.000, \sigma = 451.173$
CLASS	Lognormal	$\mu = 4.627, \sigma = 994$
OUTROS	Lognormal	$\mu = 18.210, \sigma = 68.428$

Este resultado previne o problema relatado recentemente por [Gong et al. 2005] sobre a falha dos testes de aderência e dificuldades de caracterização quando a distribuição alvo é do tipo cauda pesada, como ocorreria se os arquivos fossem agrupados em uma única classe. Mesmo para as classes menos importantes e não listadas na Tabela 3.3 o tamanho dos arquivos transmitidos pôde ser caracterizado por uma distribuição Lognormal. Isto

Tabela 3.7: Distribuições ajustadas para o tempo de transmissão de arquivos em cada classe para o servidor WC98.

Classe	Modelo	Parâmetros
GIF	Weibull	$b = 0,43, a = 0,26$
HTML	Weibull	$b = 0,37, a = 2,46$
JPEG	Weibull	$b = 0,29, a = 1,62$
ZIP1	Weibull	$b = 0,72, a = 27,58$
ZIP2	Weibull	$b = 0,73, a = 22,52$
MOV	Weibull	$b = 1,36, a = 53,97$
PL	Weibull	$b = 0,50, a = 9,12$
HQX	Weibull	$b = 0,36, a = 2,83$
CLASS	Weibull	$b = 0,28, a = 0,62$
OUTROS	Weibull	$b = 0,30, a = 1,64$

contrasta com a distribuição de cauda pesada reportada na literatura observada quando os arquivos são tratados em conjunto.

A caracterização do tamanho dos arquivos em cada classe foi repetida para o servidor da Copa do Mundo de 1998. A Figura 3.5 mostra a distribuição empírica (obtida a partir da amostra) das principais classes do servidor da Copa do mundo de 1998 comparada com a distribuição Lognormal. A linha pontilhada representa a distribuição teórica e os pontos representam a amostra. Verifica-se visualmente uma boa aderência dos dados à distribuição Lognormal. O sumário para estatísticas básicas dos arquivos analisados da WC98 são apresentados na Tabela 3.6. Esta caracterização está consistente com o estudo similar realizado por Arlitt e Jin em [Arlitt e Jin 2000].

3.5.5 Caracterização do tempo de permanência em cada classe

Outro resultado importante foi o tempo gasto em cada classe. Descobrimos que este tempo pode ser descrito por uma distribuição de Weibull. O tempo gasto na classe é resultado do tempo necessário para a transferência do arquivo somado com o tempo de processamento do servidor e do cliente. O tamanho do arquivo de cada classe e o tempo necessário para transmiti-lo são caracterizados por distribuições diferentes. No entanto, a caracterização do tempo gasto em cada classe não é parte do modelo aqui proposto; esta caracterização será valiosa no estudo analítico utilizando-se uma cadeia de Markov para o estudo de desempenho do sistema. Este estudo será tema de trabalhos futuros.

A Figura 3.6 mostra a distribuição acumulada do tempo gasto nas principais classes para o servidor da Copa do Mundo de 1998 comparado com a distribuição Weibull,

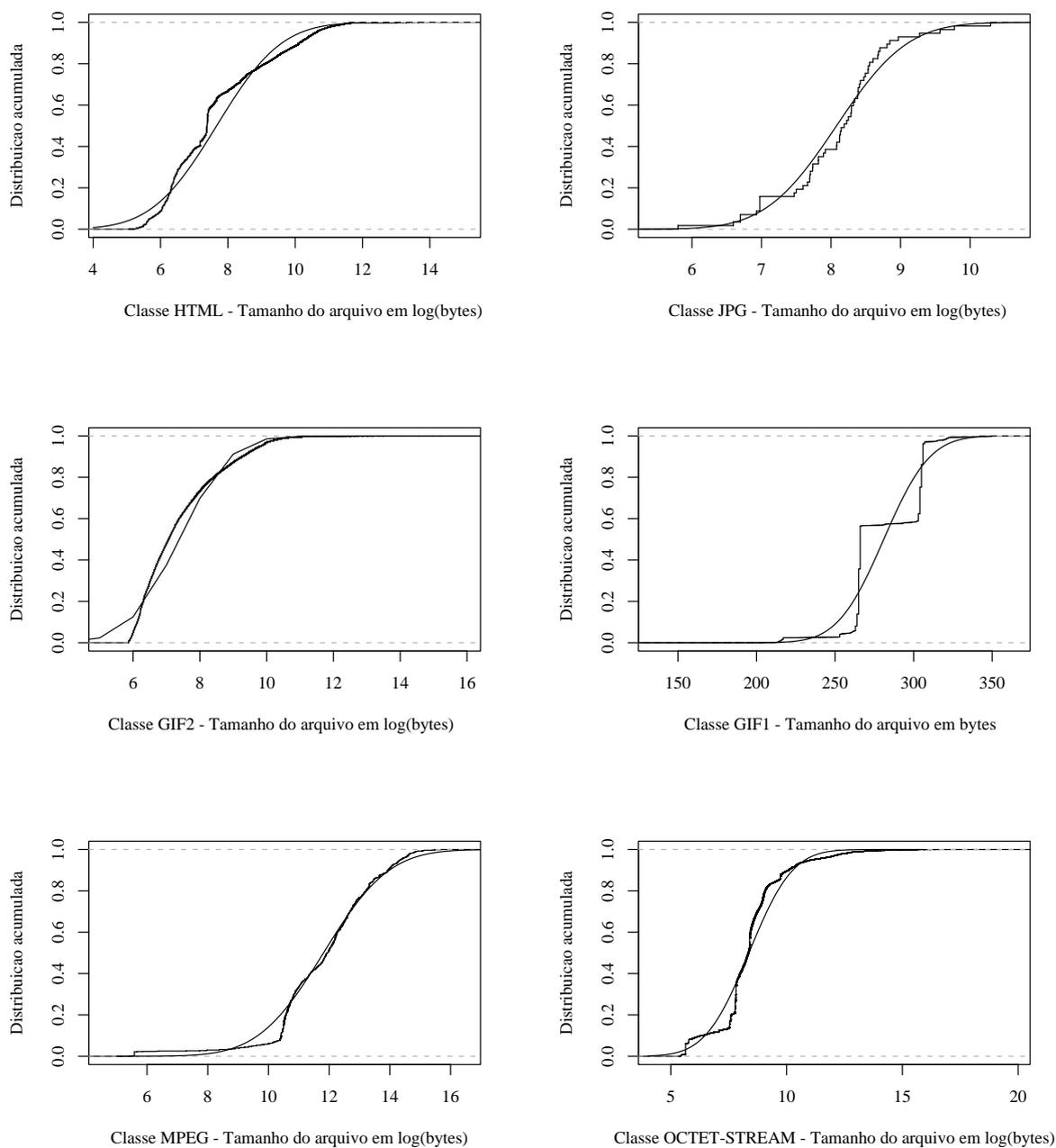


Figura 3.4: Distribuição de probabilidade teórica e distribuição amostral para cada uma das classes de arquivos do sistema IRCache. A classe *GIF1* é comparada com a distribuição Normal e as demais com a distribuição Lognormal.

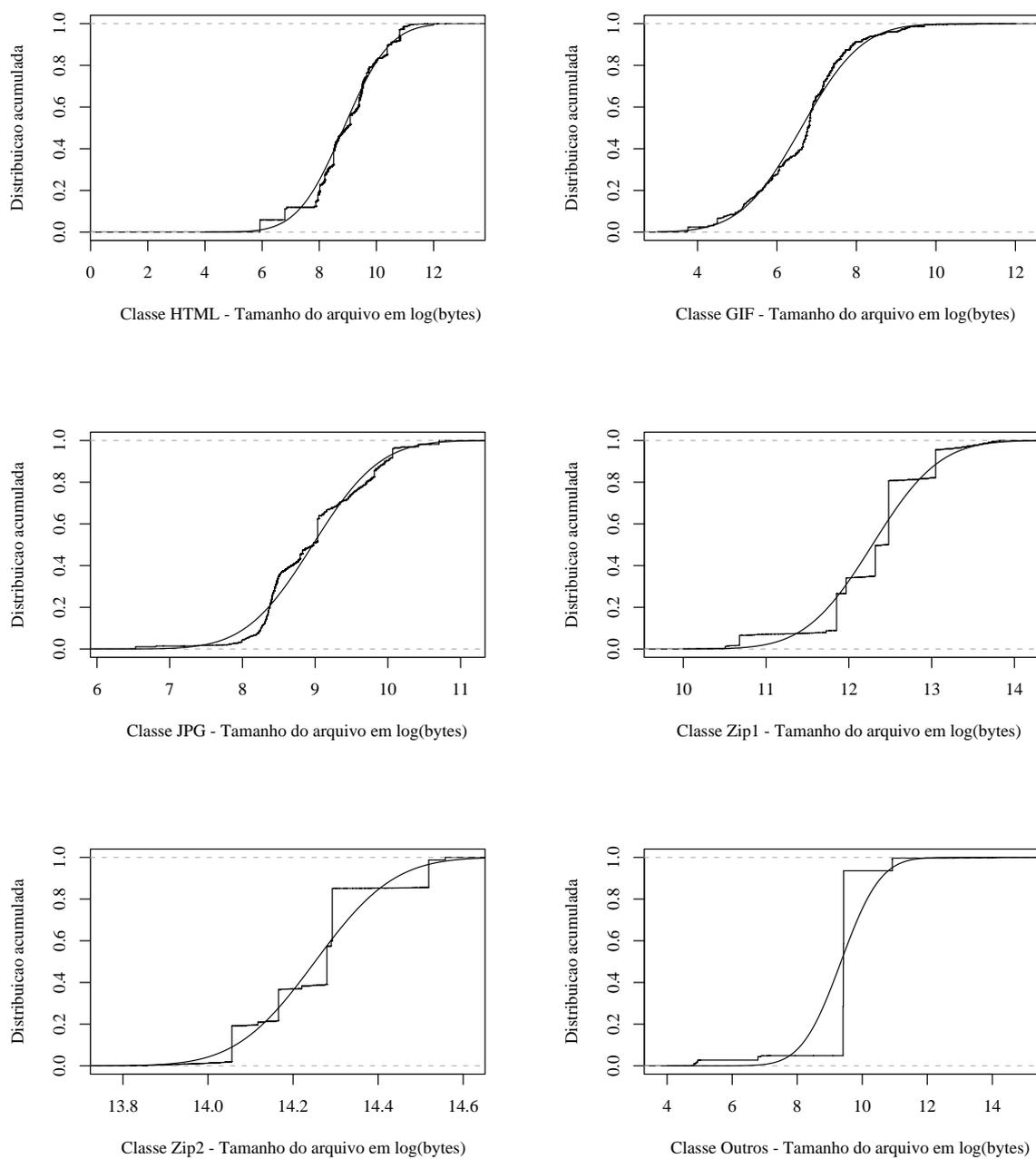


Figura 3.5: Distribuição acumulada para o tamanho dos arquivos das principais classes de arquivos da Copa do Mundo de 1998 comparado com a distribuição Lognormal.

ilustrando a boa aderência observada.

A Tabela 3.7 mostra os parâmetros da distribuição de Weibull utilizada para caracterizar o tempo de transmissão dos arquivos de acordo com sua classe.

3.6 Validação e simulação

Para validar o modelo foi realizada uma simulação utilizando o software simulador *NS-2* [Breslau et al. 2000]. A topologia de simulação é apresentada na Figura 3.7. Os parâmetros utilizados na simulação são os mesmos levantados no servidor da copa do mundo de 1998, descritos nas Tabelas 3.6 e 3.4. Foi desenvolvido um programa para gerar a seqüência de requisições à páginas e este tráfego sintético foi submetido ao simulador *NS-2* como se fosse um arquivo *trace* real de um servidor *Web*. O intervalo de tempo *active on* foi gerado de acordo com a distribuição de Weibull, como descrito por Barford e Crovella em [Barford e Crovella 1998]. O processo de chegada de sessões de usuário segue o processo de Poisson, com uma chegada média de 450 requisições por hora. Esta taxa de chegada foi configurada de modo a gerar uma carga considerável ao servidor simulado sem no entanto produzir uma condição de congestionamento permanente.

A Figura 3.8 mostra o tráfego agregado observado na saída do servidor na escala de 1 segundo. Foi realizado o teste para estimar o parâmetro de Hurst do tráfego agregado utilizando-se o método da transformada Wavelet [Willinger e Park 2000] utilizando-se as ferramentas disponibilizadas por [Roughan et al. 1998].

A Figura 3.9 mostra a densidade espectral de potência em cada oitava. O estimador para o parâmetro de Hurst foi de $H \approx 0.9$. Este resultado está consistente com o tráfego produzido por servidores *Web*, e mostra a capacidade do modelo na geração de tráfego compatível com o observado em sistemas reais, como reportado por [Crovella e Bestavros 1995].

Foi também utilizado o método da estatística R/S para estimar o parâmetro de Hurst nesta mesma série. O resultado indica $H \approx 0.75$, que indica auto-similaridade na série. No entanto, esta técnica geralmente apresenta resultados piores do que o método da transformada de Wavelet.

3.7 Conclusões do capítulo e trabalhos futuros

O modelo *SURGE* é um dos melhores modelos disponíveis para geração de tráfego *Web*, mas exige que o tamanho dos arquivos seja caracterizado de acordo com uma única

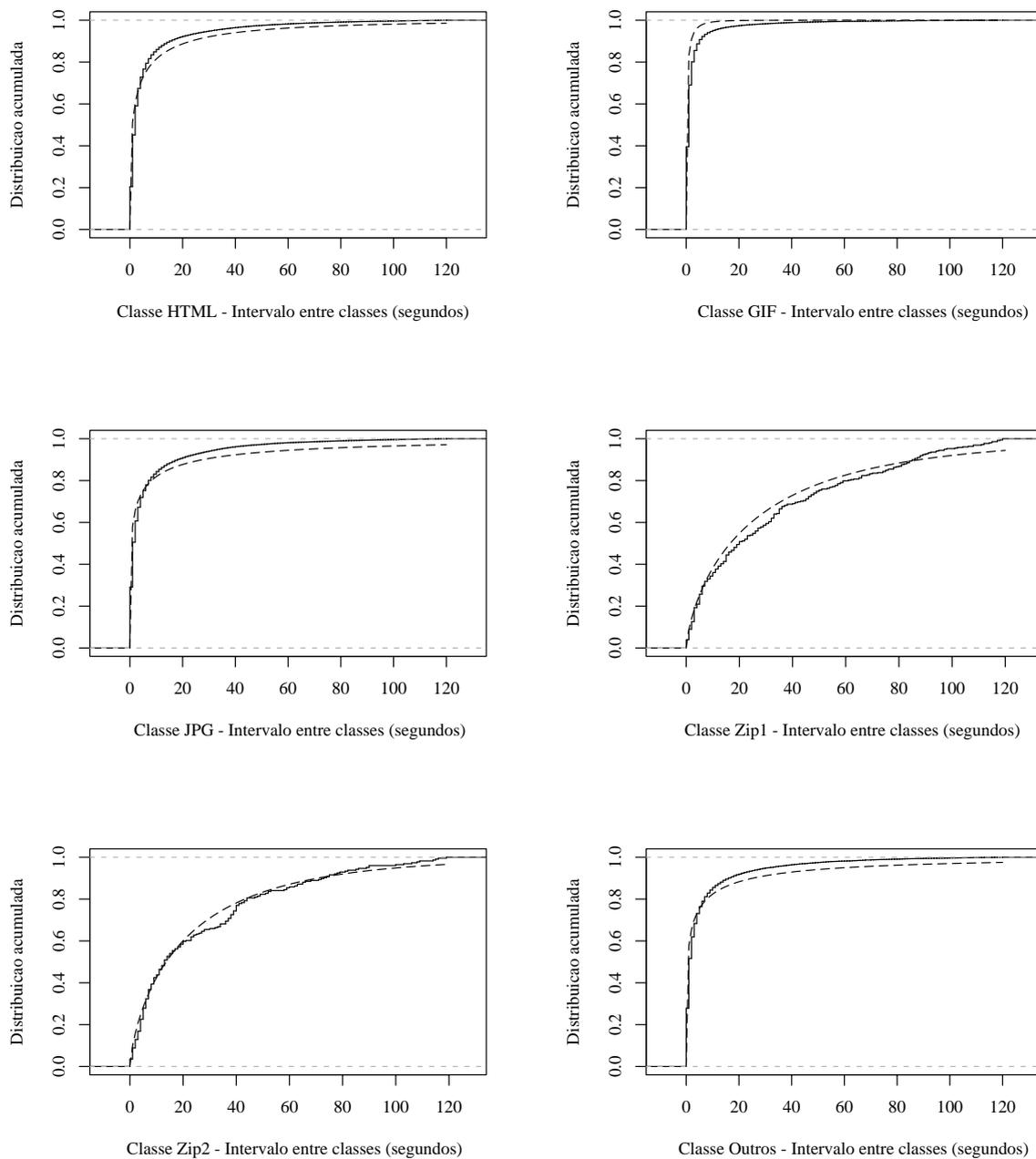


Figura 3.6: Distribuição acumulada de probabilidade para o tempo de permanência em cada classe comparada com a distribuição de Weibull (linha pontilhada).

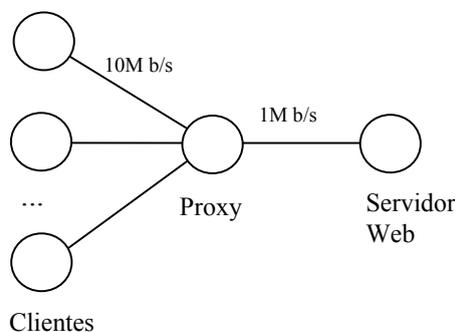


Figura 3.7: Topologia da Simulação.

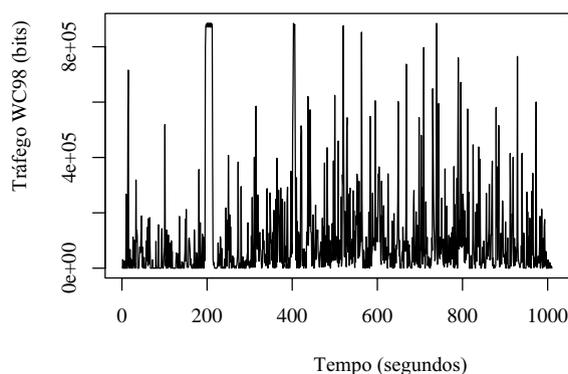


Figura 3.8: Série temporal representando o tráfego de saída do servidor da WC98 na escala de 1 segundo.

distribuição de probabilidade. Normalmente, a distribuição utilizada é uma distribuição de cauda pesada, o que pode levar a problemas na análise estatística [Gong et al. 2005]. Além disso, [Barford e Crovella 1998] sugere o uso da distribuição Lognormal para descrever o corpo do tamanho dos arquivos transferidos e a distribuição de Pareto para descrever a cauda. Isto resulta em um problema quando implementando o modelo em simuladores de redes, como no *NS-2*. O usuário é forçado a utilizar somente uma distribuição, o que pode levar a uma geração de tráfego incorreta. O modelo proposto resolve estes problemas através da classificação dos arquivos transmitidos em classes. O modelo proposto neste capítulo captura as características do tráfego realizando a separação dos arquivos transmitidos em diversas classes utilizando apenas a extensão do arquivo. No entanto, é possível realizar a montagem de classes baseadas em critérios mais refinados. Foi mostrado que, para cada classe, a distribuição Lognormal pode ser utilizada para caracterizar o tamanho do arquivo e a distribuição de Weibull pode ser utilizada para caracterizar o tempo de permanência em cada classe.

Como contribuições desta nova abordagem de modelagem podem ser citadas a am-

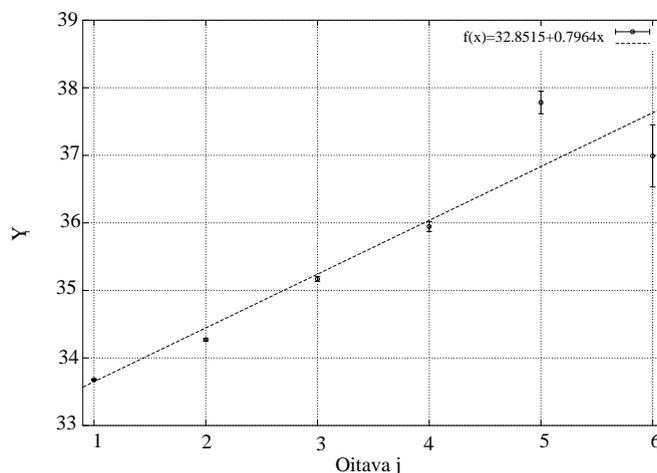


Figura 3.9: Estimador do parâmetro de Hurst utilizando a transformada Wavelet para o tráfego de saída do servidor WC98.

pliação de possibilidades em termos de novos modelos analíticos, uma geração de tráfego sintético mais precisa do que o SURGE através do refinamento em classes de arquivos e a novas possibilidades de pesquisa de técnicas de melhoria de desempenho de sistemas *Web*, por exemplo, o desenvolvimento de algoritmos de gerência de áreas de cache.

A caracterização do tráfego de um servidor *Web* através do novo modelo requer as seguintes informações:

1. Classes de arquivos transmitidos pelo servidor;
2. Média e desvio padrão do tamanho de cada classe de arquivo.
3. Probabilidades de transição de estados para transmissão de classes de arquivos em uma sessão;
4. Intervalo entre chegada de sessões.

Os dados necessários podem ser extraídos dos arquivos de *log* do servidor. Neste trabalho foram desenvolvidos programas para a extração e classificação dos dados dos arquivos de *log* de servidores *Web*. Também foram desenvolvidos geradores de tráfego e realizada a simulação com o software *NS-2*.

Um possível trabalho futuro é a utilização do modelo para realizar análise de desempenho utilizando cadeias de semi-Markovianas. Como o tempo de permanência em cada classe foi identificado como possuindo distribuição de Weibull, pode ser realizada uma análise aproximada (a distribuição de Weibull é caso geral da distribuição Exponencial)

utilizando a teoria já desenvolvida para falhas em sistemas. Esta análise pode levar ao desenvolvimento de novos métodos de dimensionamento e técnicas para a melhoria de desempenho de servidores.

Capítulo 4

Um método para melhorar o desempenho de servidores Web

NESTE capítulo é proposto um método para melhorar o desempenho de servidores Web. O objetivo será diminuir o tempo médio de resposta percebido pelos usuários. Serão descritas as premissas de tráfego sobre a qual a idéia se baseia bem como será detalhado o método proposto e apresentadas análises de desempenho qualitativas mostrando os benefícios obtidos.

4.1 Motivação

Como evidenciado por [Crovella e Bestavros 1995] para o tráfego Web, o tráfego agregado de saída do sistema apresenta alta variabilidade que pode levar a congestionamentos devido a rajadas. Também implica que as perdas na rajada refletem-se sobre o protocolo *TCP* como reportado por [Veres et al. 2003]: as características fractais de um fluxo *TCP* são refletidas em todo o agregado de fluxos.

É bem conhecido o impacto da perda de pacotes no *TCP Reno*, que é a implementação mais utilizada do protocolo *TCP* [Azeem et al. 2000]. Quando uma conexão *TCP* percebe uma perda de pacotes com número de seqüência próximo, estas perdas são entendidas como congestionamento da rede pelo mecanismo de controle de congestionamento do *TCP* causando decremento de sua janela de transmissão [Fall e Floyd 1996]. Esta reação do *TCP* impacta na latência da rede (com ou sem roteadores que implementam políticas de gerência de filas ativas, como *RED* [Floyd e Jacobson 1993]) uma vez que múltiplas perdas são experimentadas pelo mesmo fluxo *TCP*. Mesmo depois da multiplexação de pacotes na filas dos roteadores, os pacotes de múltiplos fluxos *TCP* geralmente tendem

a exibir um baixo grau de intercalação. Como resultado, quando eles encontram um gargalo, pacotes sucessivos de um mesmo fluxo têm alta probabilidade de serem tratados da mesma maneira, por exemplo, serem descartados [Azeem et al. 2000]. Logo, em uma situação de congestionamento, os pacotes de uma mesma sessão de usuário tendem a ser descartados, levando o *TCP* a reagir de maneira a levar o sistema a um compartilhamento injusto da banda.

Para fluxos Web, a utilização do *RED* resulta em tempos de transferência semelhantes caso fosse utilizado *FIFO*. Isto pode ser explicado pelo comportamento do mecanismo de controle de congestionamento do protocolo *TCP*. Fluxos curtos completam suas transferências durante a fase de *Slow Start*. No entanto, mesmo se a rede não estiver congestionada, o tempo de duração da transferência é dominado pelo *Round Trip Time (RTT)* da conexão, que geralmente é muito maior do que o tempo necessário para transferir os pacotes de fluxos curtos [Rai et al. 2005].

Particularmente será proposto um método que melhora a justiça na distribuição de banda entre as diversas sessões que realizam acessos simultâneos ao servidor Web, favorecendo aos fluxos curtos sem prejudicar excessivamente os fluxos longos.

Considerando as conseqüências do controle de congestionamento do *TCP* na latência da rede, a idéia geral da proposta é melhorar o desempenho do servidor realizando um acompanhamento do uso de banda por sessão de usuário. Este acompanhamento será realizado através da marcação do tráfego produzido por cada sessão de usuário do servidor. Em caso de congestionamento, esta informação será utilizada pelos roteadores do núcleo da rede para realizar o descarte de pacotes. Marcando os pacotes de maneira justa entre as sessões de usuário, o mecanismo de controle de congestionamento do *TCP* irá reagir, levando a um compartilhamento de banda mais justo, diminuindo a latência para fluxos curtos e melhorando o tempo médio de resposta percebido pelos clientes.

A aplicação deste esquema não requer alterações na implementação do *TCP* nos clientes. Nos roteadores da rede, escalonadores prioritários podem descartar pacotes baseando-se na informação da marca colocada pelo servidor de aplicação.

O método proposto não necessita que os roteadores da rede tenham que gerenciar estados de conexão, deixando a complexidade para as bordas. O método pode ser implementado utilizando-se os mecanismos já disponíveis atualmente para o conjunto de serviços diferenciados, especificados por [Blake et al. 1998].

Como hipótese principal, será considerado que o tráfego agregado produzida pela aplicação possua grande variabilidade. Devido a alterações dinâmicas nas características do tráfego, será apresentado um mecanismo automático de configuração de parâmetros

para atingir-se um bom desempenho. O método de configuração baseia-se no cálculo da média móvel ponderada exponencial do consumo de banda das sessões de usuário.

Para avaliar qualitativamente os efeitos da utilização do método, foi implementada uma simulação computacional utilizando o simulador NS-2 [Breslau et al. 2000] com um servidor Web com conexões persistentes utilizando o gerador de tráfego *SURGE* e os resultados foram analisados estatisticamente. Para implementação é proposto o uso de uma rede de serviços diferenciados [Blake et al. 1998] porque ela já prevê o uso de escalonadores prioritários no núcleo da rede. No entanto, não é objetivo deste trabalho realizar diferenciação de tráfego, apenas convenientemente utilizar os mecanismos já implementados.

4.2 Métricas

Em [Menascé e Almeida 1998] são definidas as principais medidas de desempenho de um servidor Web: conexões por segundo, taxa de transferência, tempo de resposta e erros por segundo. O tempo de resposta inclui a latência no servidor, o tempo gasto comunicando na rede e o tempo de processamento na máquina cliente. Logo, o desempenho percebido pelo usuário depende da capacidade do servidor, da carga da rede, da banda passante, bem como da capacidade do computador do cliente.

Neste trabalho será proposto um método que objetiva melhorar o tempo de resposta percebido pelos clientes atuando sobre o desempenho da comunicação na rede. Não serão alvo de estudos a latência no servidor ou o tempo de processamento na máquina cliente.

Define-se *utilização* do servidor (também referenciado por *ocupação* ou *carga*, ρ) como a fração do tempo que o sistema está ocupado atendendo requisições. O tempo de resposta percebido pelos clientes será examinado em diversas situações de utilização do servidor Web. No entanto, objetiva-se incrementar o tempo médio de resposta principalmente em situações de grande carga.

4.3 Trabalhos correlatos

Os artigos [de Rezende 1999] e [Ibanez e Nichols 1999] reportam, via estudos de simulação, que a quantidade de micro-fluxos de um agregado de tráfego, o *RTT* (*Round Trip Time*) e o tamanho médio do pacote são fatores críticos para distribuição justa de banda entre agregações que competem entre si dentro de uma mesma classe de tráfego *AF* (*Assure Forward*) do conjunto de serviços diferenciados.

Em [Nandy et al. 2001] é proposto um mecanismo de controle entre os roteadores de borda da arquitetura de serviços diferenciados para regular o tráfego agregado produzido por um cliente de uma maneira justa dentro do núcleo da rede. O objetivo é melhorar a justiça entre agregados de tráfego dentro de uma mesma classe AF.

O mecanismo é chamado de *AFC* (*Aggregate Flow Control*), e trabalha da seguinte maneira: (a) um número de conexões *TCP* são associadas com cada agregação de fluxos entre dois roteadores de borda; (b) estas conexões *TCP* inserem pacotes de controle para detectar congestionamento ao longo da rota percorrida pelo tráfego. Os pacotes de controle percorrem o mesmo caminho que os pacotes de dados; (c) o tráfego agregado é regulado no nó de ingresso baseado no descarte dos pacotes de controle.

O objetivo dos pacotes de controle *TCP* é estabelecer o tamanho de um segmento virtual (*VMSS*, *virtual maximum segment size*). O valor do *VMSS* representa a quantidade de bytes que o fluxo agregado pode transmitir, e é o valor máximo de um balde de fichas. O pacote do usuário somente pode ser encaminhado quando existirem fichas no balde. Quando um pacote é transmitido, as fichas do balde são decrementadas pela quantidade de bytes do pacote. Para cada *VMSS* bytes transmitidos pelo agregado de tráfego, um pacote de controle é gerado e transmitido e o contador de fichas do balde será incrementado por *VMSS*. A perda de pacotes de controle na rede fará com que o *TCP* reaja reduzindo sua janela de congestionamento, ajustando o valor do *VMSS*, diminuindo a taxa de transmissão efetiva.

Em [Alonso 2004] é proposta uma modificação no esquema do *AFC*, onde ao invés de utilizar o algoritmo *Slow Start* na conexão *TCP* de controle entre os roteadores de borda são utilizados a implementação do *TCP Vegas* [Brakmo e Peterson 1995] e o *TCP AIMD* [Chiu e Jain 1989]. O *TCP Vegas* apresentou maiores vantagens por ser um mecanismo pró ativo de controle de fluxo: ele é capaz de se antecipar ao congestionamento. Todo o estudo em [Alonso 2004] foi realizado utilizando uma simulação com o *NS-2*, tomando-se o cuidado de calcular o intervalo de confiança para 95% em relação ao valor médio dos parâmetros em estudo.

Por outro lado, existem métodos projetados para melhorar o desempenho do sistema utilizando informações da sessão. O método proposto por [Harchol-Balter et al. 2003] utiliza o *SRPT* (*Shortest-Remaining-Processing-Time*) implementado com um escalonador prioritário e 16 filas.

Políticas de escalonamento baseadas em duas filas prioritárias têm sido propostas com o objetivo de favorecer fluxos curtos. Configurar uma maior prioridade a fluxos curtos tende a melhorar o desempenho percebido pelos clientes sem penalizar muito o desempenho de

fluxos longos. As políticas de escalonamento propostas por [Chen e Heidemann 2003] e [Avrachenkov et al. 2004] utilizam uma variável th para diferenciar entre fluxos curtos e longos. Os primeiros th pacotes de um fluxo são enfileirados na primeira fila e os demais são enfileirados na segunda. A segunda fila é servida somente se a primeira estiver vazia e a disciplina de gerência é FIFO para ambas.

Em [Rai et al. 2005] é proposta uma implementação do protocolo *TCP* integrada a um escalonador prioritário com duas filas de modo a transmitir fluxos curtos com maior prioridade, o que resulta em redução do tempo médio de resposta. [Rai et al. 2005] propõe o uso de uma política chamada *least attained service (LAS)* para favorecer os fluxos curtos sem o conhecimento prévio do seu tamanho, onde o próximo pacote a ser servido é aquele que pertence ao fluxo que recebeu o menor esforço de serviço.

No entanto, os métodos propostos por [Harchol-Balter et al. 2003], [Chen e Heidemann 2003], [Avrachenkov et al. 2004] e [Rai et al. 2005] podem apresentar problemas em situações de grande carga do servidor, quando um fluxo longo pode não ser servido (*starvation*). Outro problema comum a todos os esquemas baseados na identificação de fluxos longos e curtos é o ajuste de parâmetros para o valor do limite th para identificar corretamente os fluxos curtos. Além disso, o *LAS* necessita que seja realizada a manutenção de estados por conexão nos roteadores da rede, o que resulta em um problema de escalabilidade da solução. No *SRPT* [Harchol-Balter et al. 2003], fluxos que transmitem apenas arquivos pequenos podem monopolizar o uso do enlace. Na proposta de [Rai et al. 2005], fluxos curtos podem monopolizar o uso do enlace.

No método aqui proposto o cálculo do consumo de banda de cada sessão é realizada em tempo de execução, de modo a permitir uma distribuição mais justa dos recursos e impedindo que fluxos com certas características possam monopolizar o uso do enlace. A abordagem aqui proposta evita o monopólio do uso de recursos uma vez que o algoritmo considera não somente o tamanho do fluxo ou do arquivo transmitido, mas o consumo de banda do cliente.

Finalmente, esquemas de diferenciação de tráfego utilizando marcação nas bordas da rede em conjunto com mecanismos de controle de congestionamento são tratados em [Chait et al. 2005]. O mecanismo aqui proposto difere deste no contexto dos objetivos. É tratada a questão do desempenho de servidores Web ao invés de diferenciação de serviços.

4.4 O Método Proposto

O método proposto neste trabalho difere dos métodos conhecidos, propondo o uso de informações disponíveis sobre a sessão (obtidas na implementação do *TCP*) para estimar o uso de banda de um determinado cliente. O tráfego de cada sessão será marcado de acordo com a utilização de banda (vermelho, amarelo e verde) e um escalonador de pacotes localizado no núcleo da rede irá encaminhar preferencialmente pacotes da fila verde. Espera-se uma melhora significativa no tempo médio de resposta e na disponibilidade do sistema sem que os fluxos longos sejam excessivamente prejudicados, o que representa uma nova alternativa em relação às propostas anteriores.

Também é proposto e validado um algoritmo de configuração automática de parâmetros para o método proposto. Desta maneira, o sistema pode adaptar-se às condições de tráfego momentâneas da rede. Serão comparados os tempos de resposta com as disciplinas de gerência de filas mais utilizadas, como *DropTail* e *Red* (*Random Early Detection*) [Floyd e Jacobson 1993].

A disciplina de gerência de filas proposta é composta de 3 partes: um marcador de tráfego, três filas (verde, amarela e vermelha) e um escalonador prioritário. O algoritmo marcador de tráfego utilizado na implementação foi o *srTCM* [Heinanen e Guerin 1999] (*Single Rate Three Color Maker*). A Figura 4.1 mostra a integração entre os componentes do algoritmo proposto.

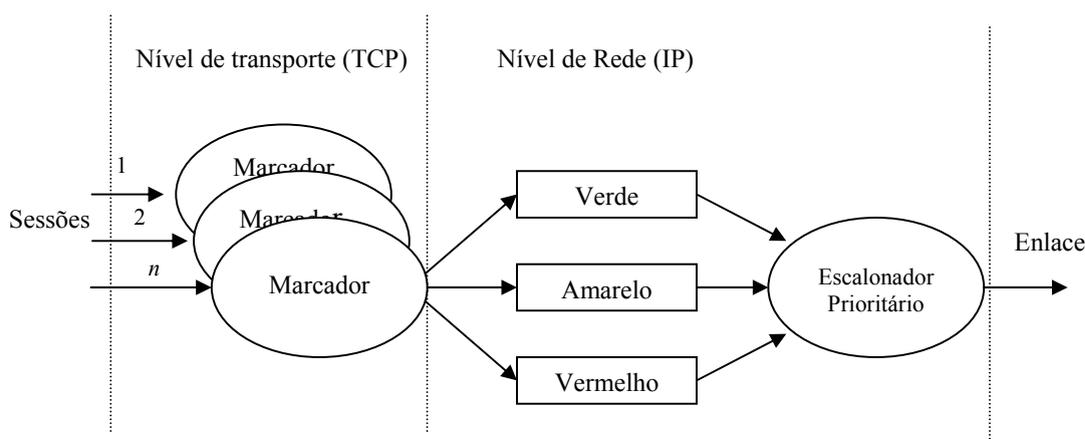


Figura 4.1: Componentes do sistema.

Os componentes da solução podem ser descritos da seguinte maneira: o marcador é responsável pela marcação dos pacotes e pela inserção do mesmo na fila correspondente à marcação (vermelha, amarela ou verde). Cada conexão realizada ao servidor irá criar

uma nova instância do marcador, que irá acompanhar o consumo de banda da conexão. Enquanto o consumo de banda estiver de acordo com o uso típico da aplicação o tráfego gerado será marcado como verde. Caso o consumo de banda exceda o uso típico, mas ainda esteja dentro de um limite para acomodar rajadas, o tráfego será marcado como amarelo. Quando o uso de banda estiver acima do consumo típico de uma sessão, o tráfego será marcado como vermelho.

Os parâmetros de configuração do marcador são o *CIR (Committed Information Rate)* representando a taxa de transmissão típica da aplicação, o *CBS (Committed Burst Size)* que indica tamanho máximo da rajada e o *EBS (Exceed Burst Size)* que informa o tamanho excedente da rajada; parâmetros “importados” do algoritmo *srTCM*.

A segunda parte da solução é composta pelas filas. Existem três filas, identificadas pelas cores verde, amarela e vermelha, respectivamente de prioridade alta, média e baixa. A disciplina de retirada de pacotes dentro de cada fila é *FIFO (First In First Out)*.

A última parte da solução é composta por um escalonador prioritário. O escalonador prioritário opera da seguinte maneira: enquanto houverem pacotes na fila verde, os pacotes desta fila serão transmitidos. Se a fila verde estiver vazia, serão retirados os pacotes da fila amarela. Se as filas verde e amarela estiverem vazias, serão retirados pacotes da fila vermelha. O algoritmo é descrito no Algoritmo 4.1.

Algoritmo 4.1 Algoritmo do escalonador prioritário.

```

Enquanto packets to be transmitted Faça
  Se greenQueue  $\neq$  empty então
    send green packets
  Senão
    Se yellowQueue  $\neq$  empty então
      send yellow packets
    Senão
      send red packets
  Fim Se
Fim Se
Fim Enquanto

```

Em resumo, o servidor seleciona um pacote para transmissão, verifica em qual fila ele deve ser inserido e o escalonador encarrega-se enviá-lo para o cliente. Dependendo do consumo de banda exigido pelo cliente, seu tráfego será classificado nas três prioridades existentes. Desta forma, quando um determinado cliente consumir uma banda maior que o uso típico da aplicação, seu tráfego será considerado menos prioritário para o sistema.

4.4.1 Algoritmo single rate Tree Color Marker - *srTCM*

O algoritmo *Single Rate Three Color Marker (srTCM)* é descrito detalhadamente na *RFC 2697* [Heinanen e Guerin 1999].

O marcador é baseado no algoritmo do balde de fichas. O algoritmo possui três parâmetros de configuração: *CIR*, *CBS* e *EBS*. O *CIR* é a taxa de incremento de fichas dos contadores para dois baldes de fichas, representados pelas variáveis *tc* e *te*. O valor inicial para o contador *tc* será de *CBS* e para *te* será de *EBS*. Cada ficha significa a permissão para transmitir um bit. Cada pacote do fluxo será tratado da seguinte maneira: se a quantidade de bits do pacote for menor que o valor de *tc*, o pacote é marcado como verde e *tc* será decrementado do número de bits do pacote. Caso contrário, se a quantidade de bits for menor que *te*, o pacote será marcado como amarelo e o valor de *te* será decrementado da quantidade de bits do pacote. Se a quantidade de bits do pacote for maior que o valor dos contadores *tc* e *te*, o pacote será marcado como vermelho (Algoritmo 4.2).

Em um processo simultâneo, os contadores *tc* e *te* estarão sendo incrementados a uma taxa *CIR* até um máximo de *CBS* e *EBS*, respectivamente.

Algoritmo 4.2 Algoritmo srTCM.

```

Se  $(tc - packet\_size) \geq 0$  então
    color = green
    tc = tc - packet_size
Senão
    Se  $(te - packet\_size) \geq 0$  então
        color = yellow
        te = te - packet_size
    Senão
        color = red
    Fim Se
Fim Se

```

4.5 Configuração automática de parâmetros

A configuração automática de parâmetros é necessária porque a mudança de cenários de tráfego pode tornar o algoritmo proposto pouco eficiente ou mesmo prejudicar o desempenho do sistema.

Cada nova sessão irá criar uma nova instância de um marcador de tráfego configurado com o valor de variáveis globais *CIR*, *CBS* e *EBS*.

A variável global CIR é recalculada ao final de cada sessão. Para prevenir a variação abrupta de parâmetros, foi escolhido o método da média móvel exponencial ponderada (*exponential weighted moving average EWMA*). Para cada conexão, i , o valor do consumo de banda, $r_i = S_b/S_t$, é calculado. S_b representa a quantidade de informação transmitida e S_t o tempo de duração da sessão. Um erro e é calculado em relação ao valor corrente de CIR ($e = r_i - CIR$), $e \in \mathbb{R}$. O novo valor de CIR é computado utilizando $CIR = CIR + \alpha \cdot e$, $0 \leq \alpha \leq 1$. α é o parâmetro que determina o peso do consumo de banda da última sessão no cálculo da média móvel.

4.6 Análise de desempenho do método proposto

O controle de tráfego realizado pelo TCP é do tipo reativo (“*close-loop*”), que pode ajustar sua taxa de transmissão de acordo com o nível de tráfego. Este é o princípio do TCP na Internet. Isso torna o estudo analítico através da teoria de filas uma aproximação do sistema real porque na teoria de filas o processo de chegada não se adapta à taxa de atendimento. Por esta razão, será realizada nesta seção uma análise aproximada utilizando a teoria de filas e posteriormente o sistema proposto será simulado para determinar qualitativamente o desempenho do sistema.

Em [Nossenson e Attiya 2004] é mostrado que quando deseja-se estudar propriedades do tempo de serviço de servidores Web é possível restringir a atenção a um sistema onde clientes enviam requisições por arquivos e os servidores os transmitem. No entanto, o tempo de serviço é influenciado pelo tamanho dos arquivos transmitidos. O tamanho dos arquivos transmitidos possui distribuição de cauda pesada e existem evidências de que o tempo de transmissão também possui características de cauda pesada, o que cria um comportamento auto-similar no tráfego agregado.

Segundo [Roberts 2000], um ponto razoável para o início da análise do tráfego auto-similar é considerar um enlace isolado e assumir a chegada de novas sessões de acordo com o processo de Poisson. Ainda segundo [Roberts 2000], se o controle de fluxo reativo do TCP atingir uma justiça exata na divisão da banda, o sistema constitui uma fila $M/G/1$ PS (*Processor Sharing*).

No entanto, o TCP não divide a banda disponível de maneira justa entre as sessões ativas. Em cada sessão o algoritmo de controle de fluxo do TCP irá ajustar o tamanho da janela corrente, o que pode produzir situações injustas de divisão de banda entre diversas sessões simultâneas, principalmente na divisão de banda entre os fluxos curtos e os fluxos longos.

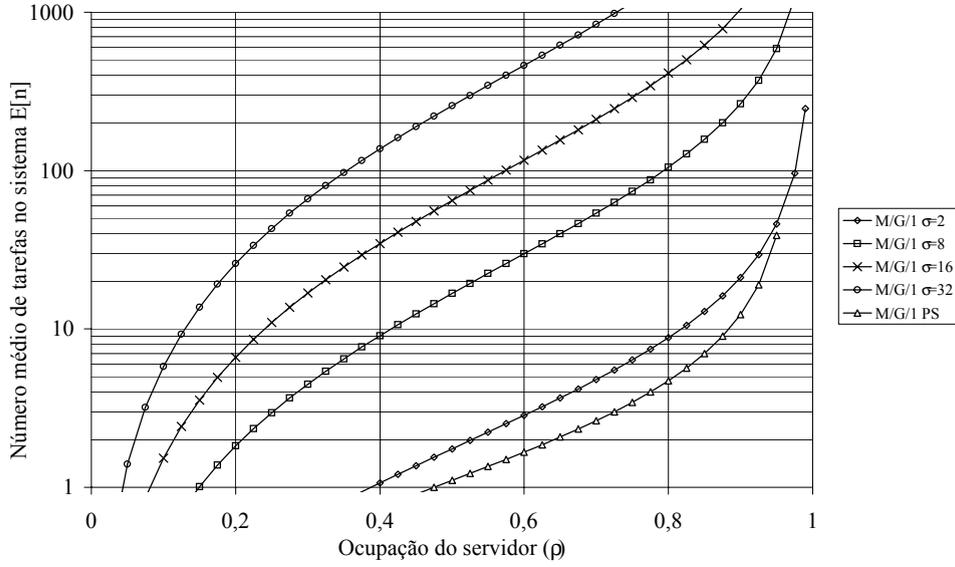


Figura 4.2: Número de tarefas no sistema com desvio padrão de $\sigma = 0, 2, 8, 16, 32$ no sistema $M/G/1$ e no sistema $M/G/1 PS$.

Para efeito de estudo, será utilizado o sistema $M/G/1$, prevendo a chegada de sessões de acordo com o processo de Poisson e será analisado o efeito do aumento da variabilidade do tempo de serviço provocado pela divisão injusta da banda entre as diversas sessões de clientes.

A equação que fornece o número médio de tarefas no sistema $M/G/1$, conhecida por equação de Pollaczek-Khinchin, é dada por

$$E[n] = \frac{\rho^2}{2 \cdot (1 - \rho)} \cdot \left[1 + \frac{\sigma^2}{S_t^2} \right] + \rho, \quad \rho < 1 \quad (4.1)$$

onde ρ é a ocupação do sistema, dado pela razão entre a taxa de chegada λ e a taxa de atendimento μ , $\rho = \lambda/\mu$. O desvio padrão do tempo de serviço é representado por σ e S_t indica o tempo médio de serviço. O tempo total de atendimento pode ser calculado utilizando-se as leis operacionais de Little [Jain 1991].

Segundo [Jain 1991], a expressão que calcula o número de elementos no sistema $M/G/1 PS$ é dada por $E[n] = \rho/(1 - \rho)$. O tempo médio de resposta pode ser obtido com as leis de Little.

O método proposto atua de maneira a tornar a divisão de banda mais justa entre as diversas sessões simultâneas (considerando fluxos longos e curtos). Deste modo, o tempo médio de resposta deve aproximar-se do sistema $M/G/1 PS$ (como afirmado por [Roberts 2000]). A Figura 4.2 compara a resposta de um sistema $M/G/1 PS$ com o

sistema $M/G/1$. Em um sistema $M/G/1$ o tempo médio de resposta aumenta proporcionalmente ao quadrado do desvio padrão do tempo de serviço.

Em [Azeem et al. 2000] mostra-se que mesmo depois da multiplexação devido a filas nos roteadores, os pacotes de múltiplos fluxos *TCP* geralmente tendem a exibir um baixo grau de intercalação. Como resultado, quando eles encontram um gargalo, pacotes sucessivos de um mesmo fluxo tem alta probabilidade de serem tratados da mesma maneira, por exemplo, serem descartados. Isso torna problemática a hipótese de [Roberts 2000] para tratar o sistema como uma fila $M/G/1$ PS. Na verdade, o desempenho do servidor estará situada entre aquele previsto por um sistema $M/G/1$ PS e o previsto por um sistema $M/G/1$. O fato do sistema constituir-se um sistema de *loop* fechado torna ainda mais difícil a utilização de modelos analíticos para análise de desempenho. Desta forma, justifica-se a utilização de ferramentas de simulação.

Na próxima seção serão apresentados os resultados da simulação do sistema, fornecendo uma demonstração qualitativa dos benefícios da utilização do método proposto.

4.7 Simulação

O método proposto foi implementado no simulador *NS-2* [Breslau et al. 2000] e foi testado utilizando um servidor *HTTP 1.1* [Fielding et al. 1999]. A implementação *full-duplex* do *TCP Reno* foi modificada para incluir a marcação de tráfego para cada sessão. A marca foi inserida no campo Prioridade do protocolo *IP* versão 6. Para a geração de tráfego foi utilizado o modelo *SURGE*. Os roteadores do núcleo da rede implementam um escalonador prioritário para as três marcas. Os parâmetros escolhidos para simulação foram os mesmos apresentados em [Barford e Crovella 1998] e [Crovella et al. 1998]. Estes dados têm sido utilizados no estudo da auto-similaridade do tráfego e foram extraídos da rede de computadores do departamento de computação da Universidade de Boston e são disponíveis na Internet para utilização em pesquisa [Cunha et al. 1995]. A Tabela 4.1 mostra um sumário dos parâmetros utilizados na simulação¹.

Para efeitos de simplicidade, foi considerado apenas um servidor de aplicação ao invés de vários. O tráfego de saída do servidor foi isolado de outras fontes de tráfego.

Foram realizados testes em duas situações de carga:

- Sistema com grande utilização: situação onde a carga do servidor foi mantida supe-

¹Uma lista completa de parâmetros e distribuições de probabilidade podem ser encontrados em [Barford e Crovella 1998].

Tabela 4.1: Sumário dos parâmetros utilizados na geração de carga para o servidor Web.

Componente	Modelo	Parâmetros
Inactive OFF time	Pareto	$k = 1, \alpha = 1.5$
Tamanho dos arquivos	Pareto	$k = 133K, \alpha = 1.1$
Embedded references	Pareto	$k = 1, \alpha = 2.43$
Request Size	Pareto	$k = 1000, \alpha = 1.0$

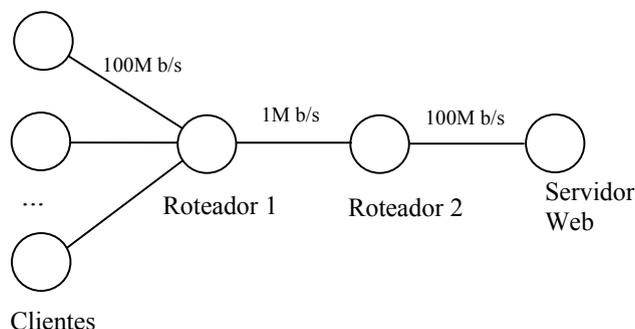


Figura 4.3: Topologia da simulação.

rior a 90% ($\rho > 0.9$);

- Sistema com pequena utilização: a carga do servidor foi alterada de modo a produzir cargas de 10% a 90% ($0.1 \leq \rho < 0.9$).

A topologia de rede utilizada na simulação é mostrada na Figura 4.3. Os roteadores foram configurados com as filas prioritárias. As requisições são produzidas simultaneamente por 100 clientes para um servidor.

O tempo necessário para transferir todos os objetos em uma sessão foi registrado. A taxa de transmissão entre os roteadores foi modificada para produzir vários níveis de congestionamento para a mesma condição de geração de tráfego. O tempo médio de resposta com intervalo de confiança de 95% foi calculado. O intervalo de confiança é mostrado nas barras verticais nos gráficos com os resultados.

4.7.1 Sistema com pequena utilização

A Figura 4.4 mostra o tempo médio de resposta percebido pelos clientes em várias situações de carga. Neste caso, a taxa de transmissão entre roteadores foi gradualmente incrementado de 1Mbps a 10Mbps. Neste gráfico a linha pontilhada é calculada com o método dos mínimos quadrados (ponderada com o erro) ajustando os pontos à curva

$f(x) = \delta + \alpha e^{\beta x}$. Por comparação, o desempenho caso fossem utilizadas as políticas *DropTail* e *Red* são apresentadas.

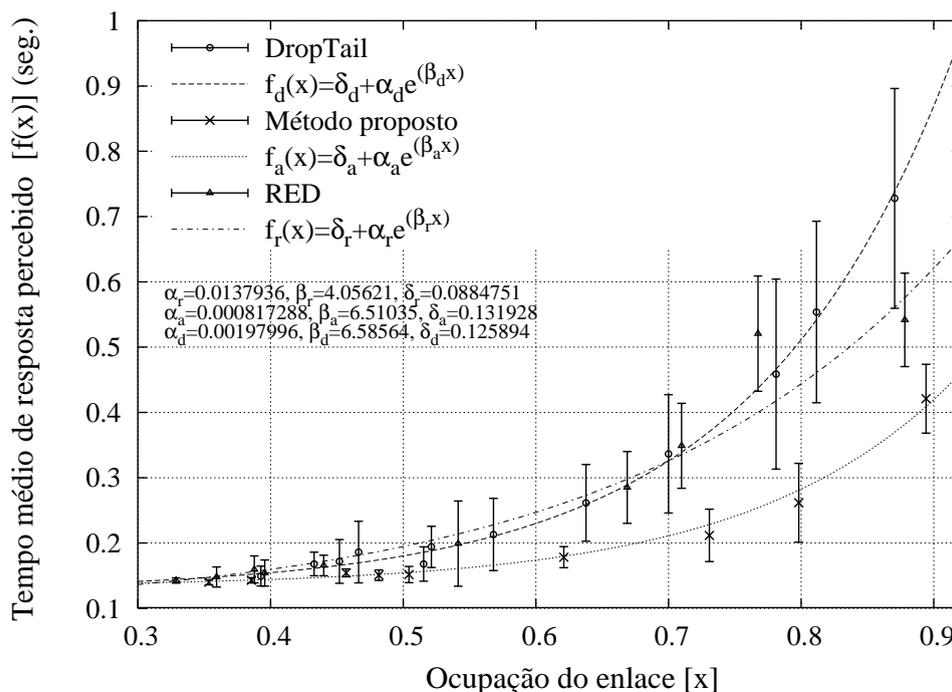


Figura 4.4: Tempo médio de resposta percebido pelos clientes em um sistema não congestionado.

O uso do escalonador prioritário com a marcação de tráfego resulta em uma melhoria significativa no tempo de resposta percebido pelos clientes a medida que a utilização do sistema aumenta. Percebe-se na Figura 4.4 que com 50% de utilização já existe uma melhora no tempo médio de resposta. Para níveis mais baixos de utilização não existe benefício significativos.

Também observa-se no gráfico da Figura 4.4 que os intervalos de confiança são menores para o método proposto. Isto ocorre porque a desvio padrão do tempo de resposta é menor se comparado ao sistema onde foram utilizados *Red* ou *DropTail*.

4.7.2 Sistema com grande utilização

A Figura 4.5 mostra o tempo de resposta para 100.000 sessões em um sistema com uma grande carga, comparando o tempo de resposta com aquele obtido por um sistema configurado com *Red* nos roteadores. A Tabela 4.2 mostra algumas estatísticas básicas

para o tempo de resposta percebido pelos usuários, comparando o desempenho de um sistema com *DropTail* e com *Red*. É importante perceber que o sistema é estacionário (a série representada pelo tempo de resposta percebido pelos usuários é um processo estacionário), como mostrado pelo rápido decaimento da função de auto-correlação e auto-correlação parcial do tempo de resposta apresentado na Figura 4.6. Os primeiros pontos da série foram descartados para análise para que a simulação atingisse a estacionariedade [Banks et al. 2001] de acordo com a inspeção visual da série. Percebe-se que o tempo de médio de resposta percebido pelos usuários foi melhorado significativamente quando utilizou-se o método proposto.

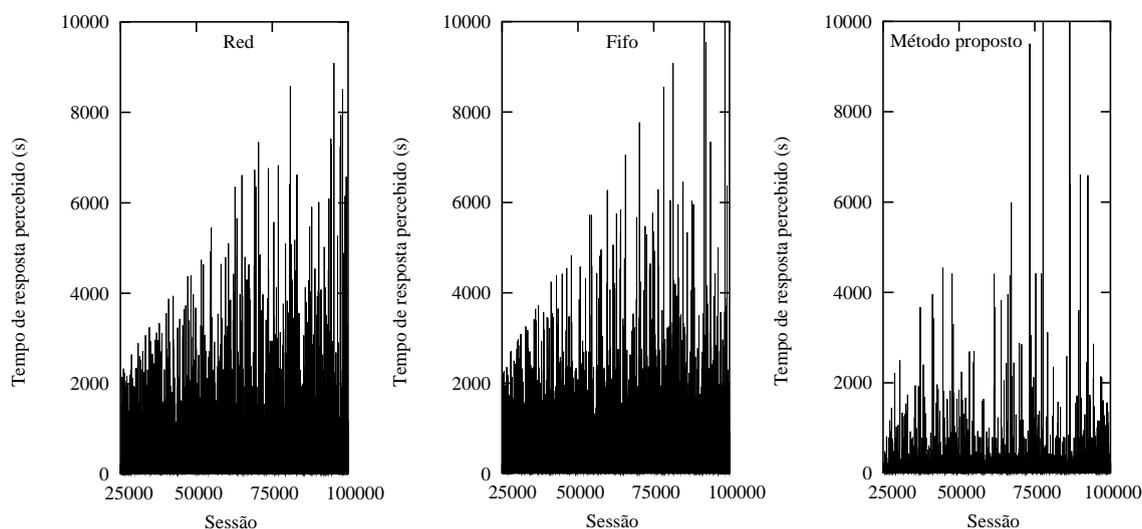


Figura 4.5: Tempo médio de resposta percebido na simulação do servidor em situação de grande carga.

Tabela 4.2: Sumário de estatísticas do tempo de resposta percebido pelos clientes da simulação do servidor com grande carga apresentado na Figura 4.5.

	Min.	1 ^o Q.	Mediana	Média ²	3 ^o Q.	Max.	Desv.P.	Índ.Just.
Proposto	0.046	1.21	6.05	24.70 ± 1.90	18.99	11010	128.84	0.506
Red	0.114	6.77	24.73	108.10 ± 12.98	83.33	9087	308.95	0.271
DropTail	0.591	6.78	24.77	114.70 ± 14.23	91.10	10480	317.18	0.360

A Figura 4.7 mostra o índice de justiça (calculado pelo índice definido no item 3.2) para a divisão de banda entre as sessões de usuário durante a simulação com grande carga.

²Média \pm semi-intervalo h para 95%confiança

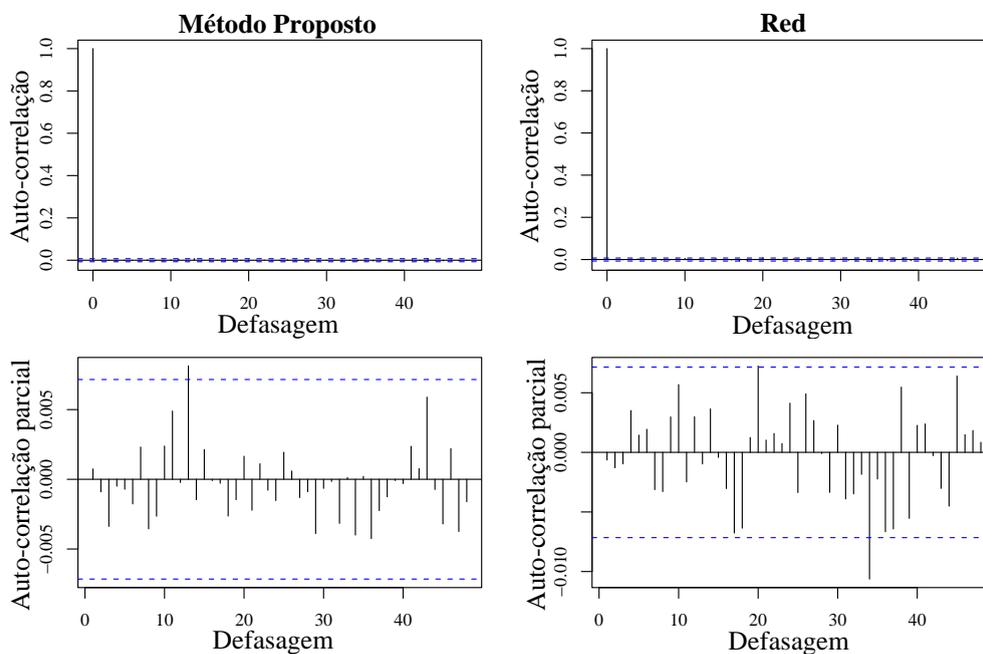


Figura 4.6: Estimativa das funções de auto-correlação e auto-correlação parcial do tempo de resposta percebido pelos clientes na situação de grande carga da série da Figura 4.5.

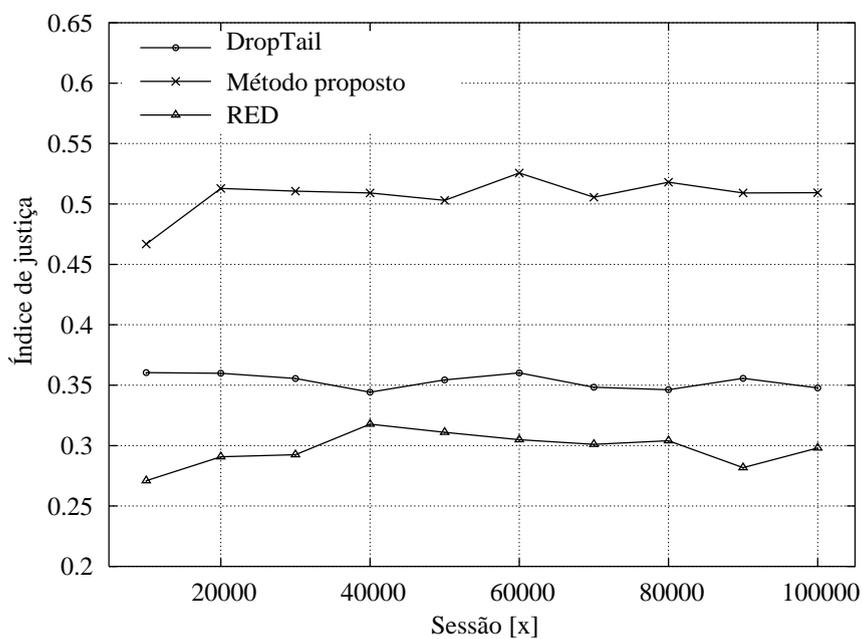


Figura 4.7: Índice de justiça entre sessões de usuário com o servidor em situação de congestionamento da Figura 4.5.

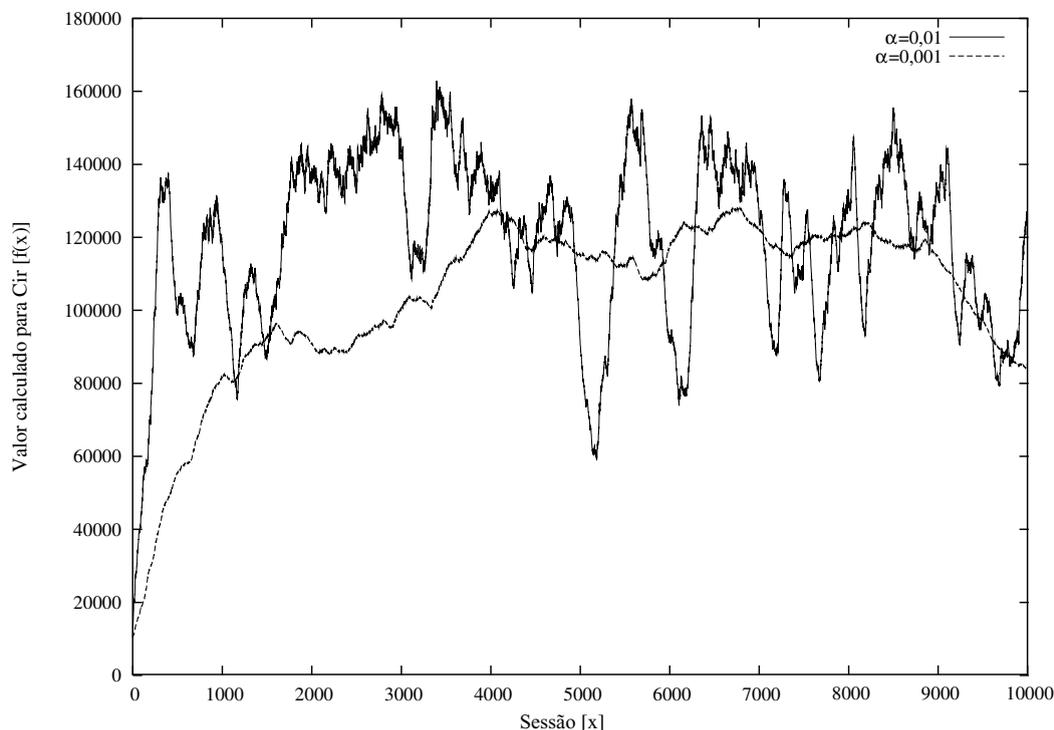


Figura 4.8: Comparação da variação do valor de *CIR* para $\alpha = 0,01$ e $\alpha = 0,001$ ao longo da simulação.

O índice foi calculado a cada 10.000 sessões. O resultado mostra uma melhora significativa da justiça na distribuição de banda entre as sessões, explicando a melhora no tempo médio de resposta.

4.7.3 Considerações sobre a implementação

O método necessita que o protocolo de aplicação implemente conexões persistentes, como o *HTTP1.1* [Fielding et al. 1999]. Conexões persistentes permitem o uso de uma única conexão *TCP* para transferência de múltiplos objetos referentes ao mesmo cliente.

Estudos atualizados reportam que aproximadamente 40% de todos os dados transmitidos por servidores *Web* são realizados utilizando-se conexões persistentes [Cao et al. 2004b].

A Figura 4.8 mostra a evolução do cálculo do valor de *CIR* ao longo da simulação para $\alpha = 0,01$ e $\alpha = 0,001$. Note que para o valor menor de α a convergência de *CIR* é mais lenta.

Nesta simulação o uso de $\alpha = 0,001$ para o algoritmo *EWMA* utilizado no ajuste

automático de parâmetros apresentou bons resultados. Com $\alpha = 0,01$ a convergência do *CIR* é perturbada por sessões consumidoras de banda. Com valores menores de α , a convergência de *CIR* é muito lenta para detectar mudanças na carga do servidor. Foram realizados testes com $\alpha = 0,01$ e $\alpha = 0,001$.

Embora o método especifique o servidor de aplicação como o marcador de pacotes, não são impostos limites ao uso do marcador em roteadores de borda da rede de serviços diferenciados. Como detalhado na *RFC 2697* [Heinane e Guerin 1999], roteadores de borda suportando a arquitetura de serviços diferenciados devem implementar a marcação de tráfego para garantir a justiça no consumo de recursos entre os diversos fluxos que participam de um agregado de tráfego. Logo, os recursos necessários para implementação do método proposto já são disponíveis nos roteadores utilizados atualmente.

O método não é limitado a redes de serviços diferenciados. A tecnologia *Diffserv* se apresenta como um método conveniente para a aplicação do método proposto, já que especifica a separação do tráfego em classes, o escalonamento prioritário e níveis de precedência de descarte. O objetivo do método é atingir melhor desempenho em condições de congestionamento e não realizar a diferenciação de serviços.

4.8 Conclusões do capítulo

A simulação realizada mostra que o uso do método proposto pode melhorar o tempo médio de resposta de sistemas quando o tempo de serviço apresenta grande variabilidade, como ocorre em servidores *Web* onde o tráfego de saída segue o padrão de tráfego auto-similar.

A disponibilidade do sistema também é melhorada porque o servidor responde um número maior de conexões simultâneas em situações de grande carga. Uma situação comum em servidores *Web* é a ocorrência de congestionamentos transientes. Nestas situações, o uso do método proposto irá apresentar grande vantagem, pois um número maior de conexões será atendida e os tempos de resposta serão mais baixos se comparado com as disciplinas de escalonamento utilizadas atualmente.

Além disso, o uso do método proposto evita necessidade da gerência de estados *por fluxo* nos roteadores da rede, que é um dos problemas das alternativas conhecidas na literatura (como em [Rai et al. 2005]).

Desde que o núcleo da Internet não esteja congestionado, não se faz necessário o uso de filas prioritárias nos roteadores de núcleo. Logo, utilização de filas prioritárias nas bordas deve ser suficiente para atingir os benefícios em termos da redução do tempo médio de

resposta percebido pelos clientes.

Neste capítulo foi apresentado também um algoritmo automático de configuração de parâmetros que apresentou bons resultados na simulação. O sistema foi capaz de adaptar-se às condições do tráfego, tornando mais eficaz o uso do método proposto.

É possível implementar o método proposto utilizando-se uma rede de serviços diferenciados e servidores que produzam uma marcação de tráfego confiável. A marcação de tráfego pode ser feita também em um equipamento separado do servidor, inclusive em um roteador de borda. As redes instaladas em universidades e nas grandes empresas já apresentam condições para implementação deste método.

Capítulo 5

Conclusões

ESTA tese apresentou um modelo para geração de tráfego e um método para melhoria de desempenho de servidores *Web*. Ambas as técnicas foram desenvolvidas supondo padrões de tráfego auto-similares apresentados por tais sistemas.

O modelo para geração de tráfego captura características comportamentais do usuário e do servidor de modo a estender os modelos existentes atualmente, gerando assim um perfil de tráfego mais próximo do tráfego real do servidor. Além disso, o modelo apresentado possui boas perspectivas para exploração em futuros projetos de pesquisa em duas direções principais. A primeira é a utilização do modelo proposto em uma análise aproximada de desempenho utilizando modelos semi-Markovianos. Outra é o desenvolvimento de técnicas de melhoria de desempenho (por exemplo, técnicas de gerência de *cache*) considerando a capacidade de representação do modelo proposto.

A utilização do método para melhoria de desempenho de servidores *Web* resultou em uma melhora considerável do tempo médio de resposta percebido pelos clientes. No entanto, a implementação de tal proposta necessita que os roteadores do núcleo da rede sejam configurados com escalonadores prioritários respeitando a marcação do tráfego realizado pelo servidor. As condições necessárias estão todas presentes na especificação do conjunto de serviços diferenciados, o que torna possível a utilização do método sem que seja necessário realizar alterações na implementação dos roteadores disponíveis atualmente. Deve-se enfatizar que o método melhora o desempenho do sistema nos casos onde o tráfego do servidor apresenta um perfil variável, que é característica do modelo auto-similar e seu uso não irá prejudicar o desempenho de sistemas com tráfego não auto-similar.

São trabalhos futuros:

- Comparação do tráfego gerado pelo modelo proposto com o tráfego gerado pelos

modelos SURGE, MMPP e com tráfego real;

- Análise do sistema utilizando modelos semi-markovianos, onde podem ser utilizados estudos sobre falhas como base. Falhas em sistemas são modelados pela distribuição de Weibull, como o tempo de permanência em cada classe do modelo proposto;
- Novo sistema para gerência de cache, reservando espaços proporcionais às classes;
- Avaliação da possibilidade de uso do método de melhoria de desempenho em outros sistemas servidores, por exemplo, servidores de correio eletrônico ou vídeo;
- Utilização do modelo de sistemas Web para modelagem de sistemas ponto-a-ponto (*peer-to-peer*).

Apêndice A

Níveis do Nilo

1157, 1088, 1169, 1169, 984, 1322, 1178, 1103, 1211, 1292, 1124, 1171, 1133, 1227, 1142, 1216, 1259, 1299, 1232, 1117, 1155, 1232, 1083, 1020, 1394, 1196, 1148, 1083, 1189, 1133, 1034, 1157, 1034, 1097, 1299, 1157, 1130, 1155, 1349, 1232, 1103, 1103, 1083, 1027, 1166, 1148, 1250, 1155, 1047, 1054, 1018, 1189, 1126, 1250, 1297, 1178, 1043, 1103, 1250, 1272, 1169, 1004, 1083, 1164, 1124, 1027, 995, 1169, 1270, 1011, 1247, 1101, 1004, 1004, 1065, 1223, 1184, 1216, 1180, 1142, 1277, 1206, 1076, 1076, 1189, 1121, 1178, 1031, 1076, 1178, 1209, 1022, 1220, 1070, 1126, 1058, 1216, 1358, 1184, 1083, 1097, 1119, 1097, 1097, 1153, 1153, 1151, 1151, 1151, 1184, 1097, 1043, 1043, 1002, 1152, 1097, 1034, 1002, 989, 1092, 1115, 1115, 1047, 1040, 1038, 1085, 1126, 1058, 1067, 1115, 1263, 1124, 1110, 1097, 1097, 1157, 1000, 991, 995, 1013, 1007, 971, 971, 980, 993, 1043, 1097, 982, 971, 971, 1065, 1022, 1029, 989, 1029, 995, 982, 1090, 980, 971, 957, 989, 966, 989, 1022, 1074, 1110, 1110, 1061, 1151, 1128, 1074, 1043, 1034, 1074, 966, 1027, 1029, 1034, 1065, 989, 1034, 1002, 1128, 1178, 1097, 1142, 1466, 1097, 1137, 1097, 1259, 1313, 1173, 1169, 1173, 1088, 1191, 1146, 1160, 1142, 1128, 1169, 1162, 1115, 1164, 1088, 1079, 1083, 1043, 1110, 1092, 1110, 1047, 1076, 1110, 1043, 1103, 1034, 1074, 1052, 1011, 1097, 1092, 1110, 1115, 1097, 1196, 1115, 1162, 1151, 1142, 1126, 1108, 1187, 1191, 1153, 1254, 1187, 1196, 1331, 1412, 1349, 1290, 1211, 1232, 1166, 1124, 1146, 1079, 1108, 1097, 1106, 1072, 1065, 1128, 1340, 959, 959, 1137, 1133, 1137, 1151, 1117, 1157, 1157, 1133, 1110, 1155, 1189, 1260, 1189, 1151, 1097, 1209, 1130, 1295, 1308, 1250, 1205, 1310, 1250, 1155, 1101, 1100, 1103, 1121, 1121, 1097, 1106, 1259, 1261, 1124, 1196, 1205, 1205, 1119, 1088, 1250, 1094, 1198, 1121, 1164, 1211, 1153, 1146, 1126, 1288, 1175, 1171, 1081, 1133, 1164, 1155, 1155, 1155, 1160, 1094, 1054, 1067, 1044, 948, 1099, 1016, 1065, 1067, 1072, 1076, 1081, 1196, 1196, 1151, 1088, 1128, 1151, 1236, 1216, 1288, 1297, 1182, 1306, 1043, 1184, 1054, 1169, 1043, 980, 1072, 1189, 1151, 1142, 1193, 1151, 1097, 1144, 1097, 1094, 1153, 1108, 935, 1081, 1081, 1097, 1146, 1250, 1151, 1043, 1043, 1043, 1070, 1124, 1137, 1146, 1099, 1054, 1045, 1070, 1142, 1074, 1101, 1220, 1196, 1097, 1207, 1119, 1160, 1151, 1025, 1097, 1137, 1007, 1034, 1043, 1043, 980, 1079, 1169, 1250, 1324, 1209, 1142, 1061, 1000, 1088, 1128, 1142, 1259, 1142, 1148, 1088, 1142, 1119, 1130, 1088, 1250, 1137, 1108, 1110, 1173, 1173, 1196, 1189, 1200, 1351, 1274, 1227, 1310, 1148, 1151, 1151, 1182, 1182, 1151, 1133, 1130, 1151, 1166, 1070, 1200, 1074, 1110, 1292, 1178, 1128, 1097, 1304, 1103, 1259, 1119, 1119, 1119, 1081, 1196, 1085, 1101, 1103, 1146, 1211, 1169, 1144, 1191, 1189, 1182, 1243, 1243, 1227, 1189, 1191, 1155, 1209, 1218, 1211, 1209, 1164, 1135, 1121, 1137, 1254, 1457, 1299, 1277, 1277, 1178, 1270, 1313, 1333, 1270, 1245, 1245, 1211, 1265, 1346, 1346, 1290, 1295, 1286, 1259, 1254, 1421, 1268, 1263, 1335, 1313, 1265, 1319, 1351, 1277, 1317, 1268, 1263, 1112, 1207, 1292, 1205, 1223, 1205, 1153, 1182, 1245, 1205, 1151, 1079, 1151, 1081, 1128, 1209, 1157, 1277, 1259, 1209, 1220, 1184, 1220, 1193, 1247, 1252, 1259, 1299, 1173, 1182, 1180, 1180, 1331, 1207, 1236, 1151, 1182, 1142, 1191, 1259, 1166, 1196, 1241, 1252, 1241, 1252, 1157, 1126, 1164, 1088, 1173, 1252, 1288, 1301, 1286, 1223, 1232, 1184, 1207, 1250, 1256, 1211, 1216, 1209, 1209, 1207, 1151, 1097, 1097, 989, 966, 1047, 1056, 1110, 1290, 1151, 1166, 1196, 1196, 1110, 1110, 1119, 1119, 1074, 1106, 1128, 1218, 1098, 1044, 1056, 1058, 1098, 1043, 1038, 1142, 1142, 1193, 1103, 989, 936, 1142, 1142, 1151, 1151, 1180, 1259, 1196, 1142, 1169, 1196, 1142, 1128, 1043, 1097, 1142, 1205, 1205, 1164, 1160, 1196, 1112, 1169, 1110, 1178, 1133, 1153, 1139, 1155, 1187, 1196, 1220, 1166, 1128, 1101, 1157, 1175, 1142, 1187, 1254, 1198, 1263, 1283, 1252, 1160, 1234, 1234, 1232, 1306, 1205, 1054, 1151, 1108, 1097¹

¹Dados com início em 622 dC, obtidos de [R Development Core Team 2005].

Apêndice B

Distribuições de Probabilidade

Foram utilizadas nesta tese as seguintes funções densidade de probabilidade:

Nome	Função
Normal	$\Pr\{X \leq x\} = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2/2\sigma^2}, \sigma > 0$
Lognormal	$\Pr\{X \leq x\} = \frac{1}{x\sigma\sqrt{2\pi}}e^{-(\ln x - \mu)^2/2\sigma^2}, \mu > 0, \sigma > 0$
Weibull	$\Pr\{X \leq x\} = \frac{bx^{b-1}}{a^b}e^{-(x/a)^b}, a > 0, b > 0$
Pareto	$\Pr\{X \leq x\} = 1 - \left(\frac{\beta}{x}\right)^\alpha, \beta \leq x$

Referências Bibliográficas

- [Abry e Veitch 1998] Abry, P. e Veitch, D. (1998). Wavelet analysis of long-range dependent traffic. *IEEE Trans. on Info. Theory*, 44(1):2–15. 18
- [Adas 1997] Adas, A. (1997). Traffic models in broadband networks. *IEEE Communications Magazine*, 37(7):82–89. 6, 20, 21, 22
- [Alonso 2004] Alonso, S. (2004). Improving aggregate flow control in differentiated services networks. *Elsevier Journal of Computer Networks*, 44(4):499 – 512. 55
- [Andersen e Nielsen 1998] Andersen, A. e Nielsen, B. (1998). A markovian approach for modeling packet traffic with long-range dependence. *IEEE JSAC*, 16(5):719–732. 27
- [Arlitt e Jin 2000] Arlitt, M. e Jin, T. (2000). A workload characterization study of the 1998 World Cup Web site. *IEEE Network*, 14:30–37. 33, 38, 41, 44
- [Avrachenkov et al. 2004] Avrachenkov, K., Ayesta, U., Brown, P., e Nyberg, E. (2004). Differentiation between short and long tcp flows: Predictability of the response time. In *INFOCOM*. 56
- [Azeem et al. 2000] Azeem, F., Rao, A., e Kalyanaraman, S. (2000). A TCP-friendly traffic marker for IP differentiated services. In *IwQoS'2000, Pittsburgh, June 2000*. 52, 53, 62
- [Banks et al. 2001] Banks, J., Carson, J., Nelson, B., e Nicol, D. (2001). *Discrete-event system simulation*. Prentice Hall, New Jersey, 3th edition. 22, 65
- [Barford e Crovella 1998] Barford, P. e Crovella, M. (1998). Generating representative web workloads for network and server performance evaluation. In *Joint International Conference on Measurement and Modeling of Computer Systems - Performance Evaluation Review (SIGMETRICS '98/PERFORMANCE '98)*. 29, 33, 34, 36, 38, 42, 47, 49, 62

- [Blake et al. 1998] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., e Weiss, W. (1998). An Architecture for Differentiated Service. RFC 2475 (Informational). Updated by RFC 3260. 53, 54
- [Box et al. 1994] Box, G., Jenkine, G., e Reineel, G. (1994). *Time Series Analysis*. Prentice-Hall, New York, 3th edition. 22, 23, 24, 25
- [Brakmo e Peterson 1995] Brakmo, L. S. e Peterson, L. L. (1995). TCP vegas: End to end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480. 55
- [Breslau et al. 2000] Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., e Yu, H. (2000). Advances in network simulation. *IEEE Computer*, 33(5):59–67. 34, 47, 54, 62
- [Cao et al. 2003] Cao, J., Andersson, M., Nyberg, C., e Kihl, M. (2003). Web server performance modeling using an M/G/1/K*PS queue. In *International Conference on Telecommunication (ICT 2003)*. 29, 32, 33
- [Cao et al. 2004a] Cao, J., Cleveland, W. S., Gao, Y., Jeffay, K., Smith, F. D., e Weigle, M. (2004a). Stochastic Models for Generating Synthetic HTTP Source Traffic. In *IEEE Infocom*. 2, 32, 35
- [Cao et al. 2004b] Cao, J., Cleveland, W. S., Gao, Y., Jeffay, K., Smith, F. D., e Weigle, M. C. (2004b). Stochastic models for generating synthetic HTTP source traffic. In *INFOCOM*. 67
- [Chait et al. 2005] Chait, Y., Hollot, C., Misra, V., Towsley, D., Zhang, H., e Cui, Y. (2005). Throughput differentiation using coloring at the network edge and preferential marking at the core. *IEEE/ACM Transactions on Networking*, 13(4):743– 754. 56
- [Chen e Heidemann 2003] Chen, X. e Heidemann, J. S. (2003). Preferential treatment for short flows to reduce web latency. *Computer Networks*, 41(6):779–794. 56
- [Chiu e Jain 1989] Chiu, D. e Jain, R. (1989). Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Journal of Computer Networks and ISDN*, 17(1):1–14. 55
- [Crovella e Bestavros 1995] Crovella, M. e Bestavros, A. (1995). Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846. 2, 6, 9, 20, 33, 34, 36, 47, 52

- [Crovella et al. 1998] Crovella, M., Taqqu, M., e Bestavros, A. (1998). *A Practical Guide to Heavy Tail*, chapter 1, pages 3–26. Chapman & Hall. 2, 62
- [Cunha et al. 1995] Cunha, C., Bestavros, A., e Crovella, M. (1995). Characteristics of World Wide Web Client-based Traces. Technical Report BUCS-TR-1995-010, Boston University, CS Dept, Boston, MA 02215. 62
- [de Rezende 1999] de Rezende, J. (1999). Assured service evaluation. In *IEEE GLOBECOM*. 54
- [Fall e Floyd 1996] Fall, K. e Floyd, S. (1996). Simulation-based comparisons of Tahoe, Reno and SACK TCP. *Computer Communication Review*, 26(3):5–21. 52
- [Fielding et al. 1999] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., e Berners-Lee, T. (1999). Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard). Updated by RFC 2817. 2, 32, 62, 67
- [Floyd e Jacobson 1993] Floyd, S. e Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413. 52, 57
- [Garrett e Willinger 1994] Garrett, M. W. e Willinger, W. (1994). Analysis, modeling and generation of self-similar vbr video traffic. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 269–280, New York, NY, USA. ACM Press. 26
- [Gong et al. 2005] Gong, W.-B., Liu, Y., Misra, V., e Towsley, D. F. (2005). Self-similarity and long range dependence on the internet: a second look at the evidence, origins and implications . *Computer Networks*, 48(3):377–399. 20, 28, 43, 49
- [Graps 1995] Graps, A. (1995). An introduction to Wavelets. *IEEE Comput. Sci. Eng.*, 2(2):50–61. 16
- [Greca e Gielow 2004] Greca, L. V. e Gielow, R. S. (2004). *Simulador de Redes de Petri Estocásticas*. Trabalho de conclusão do curso de Bacharelado em Engenharia de Computação da Pontifícia Universidade Católica do Paraná. 4
- [Harchol-Balter et al. 2003] Harchol-Balter, M., Schroeder, B., Bansal, N., e Agrawal, M. (2003). Size-based scheduling to improve web performance. *ACM Transactions on Computer Systems (TOCS)*, 21(2). 55, 56

- [Heinane e Guerin 1999] Heinane, J. e Guerin, R. (1999). A Single Rate Three Color Marker. RFC 2697 (Informational). 57, 59, 68
- [Hernandez-Campos et al. 2003] Hernandez-Campos, F., Jeffay, K., e Smith, F. (2003). Tracing the evolution of the web traffic: 1995-2003. In *IEEE/ACM MASCOTS 2003 – The 11th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. 34, 42
- [Horvath e Telek 2002] Horvath, A. e Telek, M. (2002). A markovian point process exhibiting multi fractal behavior and its application to traffic modeling. In *4th Int. Conference on Matrix-Analytic Methods in Stochastic Models*. 27
- [Hosking 1981] Hosking, J. (1981). Fractional differencing. *Biometrika*, 68:165–176. 25
- [Ibanez e Nichols 1999] Ibanez, J.-A. e Nichols, K. (1999). Preliminary simulation evaluation of an assured service. 54
- [Jain 1991] Jain, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley & Sons. 34, 61
- [Jain e Routhie 1986] Jain, R. e Routhie, S. (1986). Packet trains - measurements and a new model for computer network traffic. *IEEE Journal on Selected Areas in Communication*, 4(6):986 – 995. 10
- [Kaplan e Glass 1998] Kaplan, D. e Glass, L. (1998). *Understanding non Linear Dynamics*. Springer, New York, 1st edition. 11, 12
- [Kim e Shroff 2005] Kim, H. S. e Shroff, N. B. (2005). The notion of end-to-end capacity and its application to the estimation of end-to-end network delays. *Computer Networks*, 48(3):475–488. 2
- [Koutsoyiannis 2004] Koutsoyiannis, D. (2004). *Hydrologic persistence and the Hurst phenomenon*. *The Encyclopedia of Water*. J. H. Lehr, Wiley. 14
- [Leland et al. 1994] Leland, W., Qaqqu, M., Willinger, W., e Wilson, D. (1994). On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15. 1, 8, 9, 20, 33

- [Liu et al. 1999] Liu, J., Shu, Y., Zhang, L., Xue, F., e Yang, O. (1999). Traffic modeling based on FARIMA models. In *IEEE 1999 Canadian Conference on Electrical and Computer Engineering*, Edmonton, Alberta, Canada. 26
- [Mah 1997] Mah, B. A. (1997). An empirical model of http network traffic. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 592, Washington, DC, USA. IEEE Computer Society. 34, 38, 42
- [Mallat 1989] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693. 18
- [Melo e Fonseca 2005] Melo, C. A. V. e Fonseca, N. L. S. D. (2005). Envelope process and computation of the equivalent bandwidth of multifractal flows. *Computer Networks*, 48(3):351–375. 2, 27
- [Menascé e Almeida 1998] Menascé, D. A. e Almeida, V. A. F. (1998). *Capacity planning for Web performance*. Prentice Hall. 54
- [Mendes et al. 2002] Mendes, D., Fonseca, K., e Pedroso, C. (2002). Bandwidth fairness of a single rate three color marker algorithm implementation. In *IEEE International Conference on Communication Singapore, ICCS'2002*, Singapore. 3
- [Muniz e Kamogawa 2003] Muniz, A. e Kamogawa, H. (2003). *Método para otimização da resposta de servidores*. Trabalho de conclusão do curso de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica do Paraná. 4
- [Murata 1989] Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580. 21
- [Muscarello et al. 2004] Muscarello, L., Mellia, M., Meo, M., e Ajmone-Marsan, M. (2004). An MMPP-based hierarchical model of internet traffic. In *IEEE international conference on communications ICC2004*. 20, 27, 28, 33
- [Nandy et al. 2001] Nandy, B., Etheridge, J., Lakas, A., e Chapman, A. (2001). Aggregate flow control: improving assurances for differentiated services network. In *Conference on Computer Communications (IEEE InfoCom)*. 55

- [Nossenson e Attiya 2004] Nossenson, R. e Attiya, H. (2004). Evaluating self-similar processes for modeling Web servers. In *Symposium on Performance Evaluation of Computer Telecommunication Systems (SPECTS 2004)*. 20, 60
- [Nuzman et al. 2002] Nuzman, C., Saniee, I., e Weiss, A. (2002). A compound model for TCP connection arrivals for LAN and WAN applications. *Elsevier Journal of Computer Networks*, 3(40):319 – 337. 25, 28, 33
- [Park et al. 2005] Park, C., Hernández-Campos, F., Marron, J. S., e Smith, F. D. (2005). Long-range dependence in a changing internet traffic mix. *Computer Networks*, 48(3):401–422. 2
- [Paxson e Floyd 1995] Paxson, V. e Floyd, S. (1995). Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244. 1, 6, 7
- [Pedroso e Fonseca 2002] Pedroso, C. e Fonseca, K. (2002). Modeling WRR packet scheduler with petri nets. In *IEEE International Conference on Communication Singapore, ICCS'2002*, Singapore. 3
- [Pedroso e Fonseca 2004a] Pedroso, C. M. e Fonseca, K. (2004a). Improving the performance of web servers. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 5
- [Pedroso e Fonseca 2004b] Pedroso, C. M. e Fonseca, K. (2004b). Improving the performance of Web servers. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 5
- [Pedroso e Fonseca 2004c] Pedroso, C. M. e Fonseca, K. (2004c). Modelos de tráfego em redes de comunicações digitais. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 5
- [Pedroso e Fonseca 2004d] Pedroso, C. M. e Fonseca, K. (2004d). A queue management discipline to protect aggregate traffic systems against bad behaving data streams. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 5
- [Pedroso e Fonseca 2004e] Pedroso, C. M. e Fonseca, K. (2004e). Traffic modeling of an e-learning server with ARMA model: why this server traffic is not self similar? Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 5, 25

- [Pedroso e Fonseca 2004f] Pedroso, C. M. e Fonseca, K. (2004f). Web server workload generation by clustering the server files. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 4
- [Pedroso e Fonseca 2005a] Pedroso, C. M. e Fonseca, K. (2005a). Improving the average response time of web servers as perceived by the clients. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 4
- [Pedroso e Fonseca 2005b] Pedroso, C. M. e Fonseca, K. (2005b). A method to improve average client perceived response time. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 5
- [Pedroso e Fonseca 2005c] Pedroso, C. M. e Fonseca, K. (2005c). A simple method to improve web servers average time response and availability. Technical Report - CPGEI - Universidade Tecnológica Federal do Paraná. 5
- [Pedroso e Fonseca 2005d] Pedroso, C. M. e Fonseca, K. (2005d). Um método para melhorar o desempenho de servidores web que apresentam perfil de tráfego altamente variável. In *4th International Information and Telecommunication Technologies Symposium (I2TS2005)*, Florianópolis, Brasil. 4
- [Pedroso e Fonseca 2006] Pedroso, C. M. e Fonseca, K. (2006). Um modelo para avaliação de desempenho de servidores web utilizando classificação de conteúdo. In *XXIV Simpósio Brasileiro de Redes de Computadores (SBRC)*, Curitiba, Brasil. 4
- [Pedroso et al. 2005] Pedroso, C. M., Kotelok, M., e Fonseca, K. (2005). Um modelo para avaliação de desempenho de servidores web utilizando classificação de conteúdo [short paper]. In *4th International Information and Telecommunication Technologies Symposium (I2TS2005)*, Florianópolis, Brasil. 3
- [R Development Core Team 2005] R Development Core Team (2005). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. 22, 24, 72
- [Rai et al. 2005] Rai, I. A., Biersack, E. W., e Urvoy-Keller, G. (2005). Size-based scheduling to improve the performance of short TCP flows. *IEEE Network*, 19(1):12–17. 53, 56, 68

- [Ribeiro et al. 2005] Ribeiro, V. J., Zhang, Z.-L., Moon, S. B., e Diot, C. (2005). Small-time scaling behavior of internet backbone traffic. *Computer Networks*, 48(3):315–334. 2
- [Roberts 2000] Roberts, J. (2000). *Self-similar network traffic and performance evaluation*, chapter Engineering for quality of service, pages 401–420. John Wiley & Sons, Inc. 36, 60, 61, 62
- [Rolls et al. 2005] Rolls, D. A., Michailidis, G., e Hernández-Campos, F. (2005). Queueing analysis of network traffic: methodology and visualization tools. *Computer Networks*, 48(3):447–473. 2
- [Roughan et al. 1998] Roughan, M., Veitch, D., e Abry, P. (1998). On-line estimation of the parameters of long-range dependence. In *Proceedings Globecom '98*, volume 6, pages 3716–3721, Sydney. 47
- [R.Scorsin e da Silva 2003] R.Scorsin e da Silva, D. (2003). Servidor de video sob demanda com trafego condicionado. Trabalho de conclusão do curso de Bacharelado em Engenharia de Computação. 4
- [Sarvotham et al. 2005] Sarvotham, S., Riedi, R. H., e Baraniuk, R. G. (2005). Network and user driven alpha-beta on-off source model for network traffic. *Computer Networks*, 48(3):335–350. 2
- [Sommers e Barford 2004] Sommers, J. e Barford, P. (2004). Self-configuring network traffic generation. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 68–81, New York, NY, USA. ACM Press. 35
- [Stoev et al. 2005] Stoev, S., Taqqu, M. S., Park, C., e Marron, J. S. (2005). On the wavelet spectrum diagnostic for hurst parameter estimation in the analysis of internet traffic. *Computer Networks*, 48(3):423–445. 2
- [Taylor e Karlin 1998] Taylor, H. e Karlin, S. (1998). *An Introduction to Stochastic Modeling*. Academic Press, 3rd edition. 20, 21
- [Toussoun 1925] Toussoun, O. (1925). *M'emoire sur l'Histoire du Nil*, volume 18. M'emoires a l'Institut d'Egypte. 14
- [Veitch et al. 2005a] Veitch, D., Hohn, N., e Abry, P. (2005a). Multifractality in TCP/IP traffic: the case against. *Computer Networks*, 48(3):293–313. 2, 27

- [Veitch et al. 2005b] Veitch, D., Hohn, N., e Abry, P. (2005b). Multifractality in tcp/ip traffic: the case against. *Computer Networks*, 48(3):293–313. 2
- [Veres et al. 2003] Veres, A., Kenesi, Z., Molnár, S., e Vattay, G. (2003). TCP’s role in the propagation of self-similarity in the internet. *Computer Communications*, 26(8):899–913. 52
- [Véhel e Riedi 1997] Véhel, J. L. e Riedi, R. (1997). Fractional Brownian motion and data traffic modeling: The other end of the spectrum. *Fractals in Engineering*, pages 185–202. Submitted. 27
- [Wessels e Claffy 1996] Wessels, D. e Claffy, K. (1996). Evolution of the NLANR cache hierarchy: Global configuration challenges. Technical report, NLANR, October 1996. <http://www.nlanr.net/Papers/Cache96/>. 33, 38
- [Willinger e Park 2000] Willinger, W. e Park, K. (2000). *Self-similar network traffic and performance evaluation*. John Wiley & Sons, New York, 1st edition. 6, 7, 8, 9, 10, 11, 13, 14, 20, 29, 33, 36, 47
- [Xue e Lee 1999a] Xue, F. e Lee, T. (1999a). Approximate analysis of queueing system with self-similar input process. In *15th Triennial Conference of IFORS*, Beijing, China. 26
- [Xue e Lee 1999b] Xue, F. e Lee, T. (1999b). Modeling and prediction long-range dependent traffic with FARIMA processes. In *International Symposium on Communications*, Kaohsiung, Taiwan. 26
- [Zukerman et al. 2003] Zukerman, M., Neame, T., e Addie, R. (2003). Internet traffic modeling and future technology implications. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 1, pages 587 – 596. 6

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)