

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CIn – CENTRO DE INFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ProCon - Prognóstico de Congestionamento de Tráfego de Redes usando Wavelets

por

Jorge Luiz de Castro e Silva

Tese submetida à avaliação,
como requisito parcial para a obtenção do grau de
Doutor em Ciência da Computação

Prof. Paulo Roberto Freire Cunha

Profa. Marcília Andrade Campos

Orientadores

Recife - PE, maio de 2004

FICHA CATALOGRÁFICA

CASTRO E SILVA, J. L.

ProCon - Prognóstico de Congestionamento de Redes de Computadores usando Wavelets

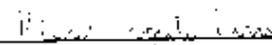
165p.: 37 figuras, 34 quadros e 4 tabelas.

Tese de Doutorado – Universidade Federal de Pernambuco. Programa de Pós-Graduação em Ciência da Computação, Recife, PE, Brasil, 2004. Orientadores: Cunha, Paulo R. F.; Campos, Marcilia A.

1. Engenharia de Tráfego, 2. Congestionamento de Redes, 3. Transformada de Wavelet, 4. Modelo de Tráfego.

I. Castro e Silva, J. L.; II. Título.

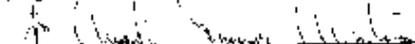
Tese de Doutorado apresentada por Jorge Luiz de Castro e Silva a Pós-Graduação em Ciências da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "ProCon-Prognóstico de Congestionamento de Tráfego de Redes usando Wavelets", orientada pelo Prof. Paulo Roberto Freire Cunha e aprovada pela banca Examinadora formada pelos professores:



Prof. Nelson Souto Rosa
(primeiro examinador)



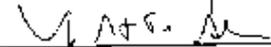
Prof. Paulo Romero Martins Maciel
(segundo examinador)



Prof. José Augusto Suruagy Monteiro
(terceiro examinador)



Prof. Manoel Raimundo de Sena Junior
(quarto examinador)



Prof. Virgílio Augusto Fernandes Almeida
(quinto examinador)

Visto e permitida a impressão
Recife, 29 de junho de 2004



Prof. Jaelson Freire Brelaz de Castro
Coordenador da Pós-Graduação em Ciência da Computação
Centro de Informática da Universidade Federal de Pernambuco

Previsão é difícil, principalmente do futuro

Niels Bohr

Dedico esta tese à minha mãe (*in memoria*),
ao meu pai e a minha família, base de todas
as minhas conquistas.

Agradecimentos

À Universidade Federal de Pernambuco (UFPE) e ao Centro de Informática (Cin), através do Programa de Pós-Graduação em Ciência da Computação, pela oportunidade de realizar este trabalho e pela acolhida.

O meu mais profundo agradecimento à Profa. Dra. Marcilia Andrade Campos, minha co-orientadora, pelos conselhos e orientações em todos os momentos do presente trabalho, particularmente os decisivos, pelas inúmeras correções e sugestões apresentadas e, principalmente, pela amizade da qual tenho o privilégio de compartilhar.

Ao Prof. Dr. Paulo Roberto Freire Cunha, meu orientador, pela orientação, pela acolhida e pelas contribuições apresentadas.

Ao Conselho Nacional de Pesquisa (CNPq) pelo aporte financeiro à realização de parte desta pesquisa.

Aos membros da banca de tese, Prof. Dr. Virgílio Augusto Fernandes Almeida (UFMG), Prof. Dr. José Augusto Suruagy Monteiro (Unifacs), Prof. Dr. Manoel Raimundo de Sena Junior (UFPE), Prof. Dr. Nelson S. Rosa (UFPE) e Prof. Dr. Paulo Romero M. Maciel (UFPE), pelas correções e sugestões apresentadas.

À equipe do Projeto AUTO-SIM, particularmente Arnoldo, Erick, Diogo e Viviane, pela colaboração na minha pesquisa.

À minha esposa, Sônia Maria Martins de Castro e Silva, pelo incentivo que me deu para realização dessa árdua pesquisa, pela solidariedade em todos momentos difíceis e, principalmente, pela compreensão.

Aos meus filhos, Jorge Luiz de Castro e Silva Junior e Lucas Martins de Castro e Silva, pelo apoio que me deram para mais uma conquista.

Às demais pessoas e instituições que, direta ou indiretamente, colaboraram para elaboração deste trabalho.

Sumário

Lista de Figuras	iv
Lista de Quadros.....	vi
Lista de Tabelas.....	viii
Lista de Abreviaturas	ix
Resumo.....	xii
Abstract	xiii
Capítulo 1	1
Introdução.....	1
1.1 O Problema.....	1
1.2 Objetivo Geral da Tese.....	3
1.2.1 Objetivos Específicos	3
1.2.2 Relevância	4
1.3 Trabalhos Relacionados	5
1.3.1 Revisão Bibliográfica sobre Congestionamento e Engenharia de Tráfego.....	5
1.3.2 Revisão Bibliográfica sobre Modelagem de Tráfego.....	15
1.3.3 Revisão Bibliográfica sobre <i>Wavelets</i> na Modelagem de tráfego.....	17
1.4 Contribuições	18
1.5 Estrutura do Trabalho.....	20
1.6 Comentários	21
Capítulo 2	22
Fundamentos Conceituais	22
2.1 Congestionamento de Redes	22
2.1.1 Congestionamento na Internet.....	23
2.1.2 Políticas e Mecanismos de Gerenciamento de Filas	25
2.1.3 Controle de Congestionamento de Fluxos Amigáveis ao TCP (<i>TCP-friendly</i>) .	28
2.1.4 Tecnologias em QoS	29
2.2 Engenharia de Tráfego	30
2.2.1 Problemas com os Protocolos de Roteamento	31
2.2.2 Objetivos de Desempenho da Engenharia de Tráfego	32

2.3 Modelagem de Tráfego de Redes.....	33
2.3.1 Modelagem Tradicional	35
2.3.2 Auto-Similaridade e Novas Abordagens em Modelagem de Tráfego	38
2.4 <i>Wavelet</i>	47
2.4.1 Transformada de Wavelet Discreta	50
2.4.2 Algoritmo Piramidal.....	52
2.4.3 Esquema <i>Lifting</i>	56
2.5 Métodos de Estimação de Tendência em Séries Temporais	58
2.5.1 Teste para Tendência.....	59
2.6 Comentários	61
Capítulo 3	62
Metodologia e Cenário	62
3.1 Localização do Mecanismo de Prognóstico de Congestionamento	62
3.2 Arquitetura do Mecanismo de Prognóstico de Congestionamento	64
3.3 Parâmetros do Mecanismo de Prognóstico de Congestionamento.....	69
3.3.1 Base de <i>Wavelet</i>	69
3.3.2 Política de Redução dos Coeficientes	70
3.3.3 Método de Escolha do Limiar de Redução dos Coeficientes.....	71
3.3.4 Janela de Observação	72
3.3.5 Indicador de Congestionamento.....	72
3.3.6 Resumo dos Parâmetros de Entrada do Mecanismo ProCon	73
3.4 Especificação do Indicador de Tendência.....	73
3.5 Descrição do Tráfego Simulado.....	75
3.6 Descrição dos Programas do Mecanismo ProCon	77
3.7 Cenário de Execução do Mecanismo ProCon	82
3.8 Comentários	84
Capítulo 4	85
Resultados e Discussão	85
4.1 A Escolha dos Parâmetros do Mecanismo	85
4.2 Resultados Usando os Parâmetros Padrões do Mecanismo	87
4.3 Resultados Variando os Parâmetros do Mecanismo	97
4.4 Tempo de Execução do Mecanismo.....	101

4.5 Validação do Mecanismo ProCon.....	103
4.6 Comentários Finais.....	104
Capítulo 5	106
Conclusões	106
5.1 Trabalhos Futuros.....	107
Referências Bibliográficas	109
Anexos.....	122
Apêndices	139

Lista de Figuras

Figura 2.1. Período sem controle	23
Figura 2.2. Comportamento do controle de congestionamento do TCP.....	24
Figura 2.3. Encaminhamento de pacotes: (a) sem engenharia de tráfego; (b) com engenharia de tráfego	32
Figura 2.4. Modelagem de tráfego: $X(t)$, tráfego de chegada; $X_f(t)$, tráfego filtrado, μ - taxa média do tráfego filtrado e $\hat{X}_f(t+h)$, tráfego previsto no tempo $t+h$	33
Figura 2.5. Modelo de tráfego ON/OFF para uma única fonte de voz	36
Figura 2.6. Auto-similaridade em tráfego real (Leland et al. 1994).....	39
Figura 2.7. Função de autocorrelação de processos DCD e DLD.....	44
Figura 2.8. Escalas de uma função ω : (a) $s < 1$; (b) $s = 1$; (c) $s > 1$	47
Figura 2.9. <i>Wavelet</i> de Haar	49
Figura 2.10. Dilatações e translações da <i>wavelet</i> de Haar em $[0,1]$	49
Figura 2.11. “Função de dados” em $[0,1]$	51
Figura 2.12. Algoritmo piramidal	54
Figura 2.13. Exemplo de algoritmo piramidal	55
Figura 2.14. Esquema <i>lifting</i> - divisão recursiva em médias e diferenças	57
Figura 3.1. Arquitetura de um roteador baseado em barramento.....	63
Figura 3.2. Diagrama funcional de um roteador	64
Figura 3.3. Diagrama do laço de execução do mecanismo ProCon.....	66
Figura 3.4. Mecanismo de prognóstico de congestionamento	67
Figuras 3.5a e 3.5b. Política dura e suave com limiar $\lambda = 1$ para a $f(x) = x$	71
Figura 3.6. Processo ON/OFF	76
Figura 3.7. Descrição do mecanismo simulador	76
Figura 3.8. Fluxograma das funções do ProCon	80

Figura 3.9. Tela de execução do programa ProCon	82
Figura 4.1. Autocorrelações	90
Figura 4.2. Série original e série transformada pela base de Daubechies_8	91
Figura 4.3. Séries original e transformada com as primeiras 512 observações	92
Figura 4.4. Saída do mecanismo ProCon – da 1 ^a à 4 ^a janela.....	93
Figura 4.5. Saída do mecanismo ProCon – da 5 ^a à 8 ^a janela.....	94
Figura 4.6. Séries original e transformada – da observação de número 513 a 1024.....	94
Figura 4.7. Saída do mecanismo ProCon – da 9 ^a à 12 ^a janela.....	95
Figura 4.8. Séries original e transformada – da observação de número 1025 a 1536.....	95
Figura 4.9. Saída do mecanismo ProCon – da 13 ^a à 16 ^a janela.....	96
Figura 4.10. Séries original e transformada – da observação de número 1537 a 2048.....	96
Figura 4.11. Saída do mecanismo ProCon com janela de observação de tamanho 512	98
Figura 4.12. Saída do mecanismo usando as bases (a) Spline_2_2 e (b) Daubechies_8	99
Figura 4.13. Séries de 1024 observações com métodos de limiar “quantil” e “universal-1”. 100	
Figura 4.14. Série de 1024 observações com política de filtragem “soft”	101

Lista de Quadros

Quadro 1.1. Grupos de Pesquisa	6
Quadro 1.2. Recomendações do IETF sobre controle de congestionamento.....	10
Quadro 1.3. Recomendações do IETF sobre engenharia de tráfego	11
Quadro 1.4. Recomendações do ITU-T para modelagem de tráfego.....	12
Quadro 1.5. Recomendações do ITU-T para medição de tráfego.....	13
Quadro 1.6. Recomendações do ITU-T para previsão de tráfego.....	14
Quadro 1.7. Recomendações do ITU-T para controle e dimensionamento de tráfego	14
Quadro 1.8. Recomendações do ITU-T para monitoração de desempenho.....	15
Quadro 3.1. Funções do Sistema ProCon.....	78
Quadro 3.2. Programas Auxiliares	81
Quadro 3.3. Amostra do arquivo gerado	83
Quadro 3.4. Bases de <i>wavelet</i> , políticas de filtragem e métodos de limiar.....	84
Quadro 4.1. Fatores escolhidos	86
Quadro 4.2. Bases escolhidas pelo mecanismo.....	88
Quadro 4.3. Tempos de execução	102
Quadro 4.4. Número de acertos do ProCon	103
Quadro 4.5. Número de acertos do modelo FARIMA	104
Quadro A.1. Filtros da base de AdelsonSimoncelliHingorani.....	139
Quadro A.2. Filtros da base de AntoniniBarlaudMatheiuDaubechies_4_4.....	139
Quadro A.3. Filtros da base de BattleLemarie	140
Quadro A.4. Filtros da base de BurtAdelson	141
Quadro A.5. Filtros da base de Coiflet_2.....	141
Quadro A.6. Filtros da base de Coiflet_4.....	141
Quadro A.7. Filtros da base de Coiflet_6.....	142

Quadro A.8. Filtros da base de Daubechies _4	142
Quadro A.9. Filtros da base de Daubechies _6	142
Quadro A.10. Filtros da base de Daubechies _8	143
Quadro A.11. Filtros da base de Daubechies _10	143
Quadro A.12. Filtros da base de Daubechies _12	143
Quadro A.13. Filtros da base de Daubechies _20	144
Quadro A.14. Filtros da base de Haar	144
Quadro A.15. Filtros da base de Pseudocoiflet_4_4	145
Quadro A.16. Filtros da base de Spline_2_2.....	145
Quadro A.17. Filtros da base de Spline_2_4.....	146
Quadro A.18. Filtros da base de Spline_3_3.....	146
Quadro A.19. Filtros da base de Spline_3_5.....	147
Quadro A.20. Filtros da base de Spline_3_7.....	147

Lista de Tabelas

Tabela 4.1. Parâmetro de Hurst dos coeficientes e das séries simuladas e transformadas.....	88
Tabela 4.2. Estatísticos dos coeficientes de <i>wavelets</i>	89
Tabela 4.3. Estatísticos da série simulada com 1024 observações.....	89
Tabela 4.4. Estatísticos da série simulada com 2048 observações.....	90

Lista de Abreviaturas

ABR	<i>Available Bit Rate</i> – (Taxa de Bit Disponível)
AEA	<i>Alisamento Exponencial Adaptativo</i>
AES	<i>Alisamento Exponencial Simples</i>
AQM	<i>Active Queue Management</i> – (Gerenciamento Ativo de Fila)
AR	<i>Auto-Regressive</i> – (Modelo Auto-Regressivo)
ARED	<i>Adaptative RED</i> – (RED Adaptativo)
ARIMA	<i>Auto-Regressive Integrated Moving Average</i> – (Modelo Auto-Regressivo-Integrado-Médias-Móveis)
ATM	<i>Asynchronous Transfer Mode</i> – (Modo de Transferência Assíncrona)
B_ISDN	<i>Broadband - Integrated Services Digital Network</i> – (Rede Digital de Serviços Integrados de Banda Larga)
CAC	<i>Connection Admission Control</i> – (Controle de Admissão de Conexão)
CAIDA	<i>Cooperative Association for Internet Data Analysis</i> – (Associação Cooperativa para Análise de Dados da Internet)
CCSN	<i>Common Channel Signaling Network</i> – (Rede de Sinalização de Canal Comum)
CM	<i>Congestion Manager</i> – (Gerente de Congestionamento)
CNRG	<i>Cornell Networking Research Group</i> – (Grupo de Pesquisa da Universidade de Cornell)
DARPA	<i>Defense Advanced Research Projects Agency</i> – (Agência de Projetos de Pesquisa Avançados de Defesa)
DCD	<i>Dependência de Curta Duração</i>
DiffServ	<i>Differentiated Services</i> – (Serviços Diferenciados)
DLD	<i>Dependência de Longa Duração</i>
ECN	<i>Explicit Congestion Notification</i> – (Notificação de Congestionamento Explícito)
FA	<i>Filtragem Adaptativa</i>
FARIMA	<i>Fractional Auto-Regressive Integrated Moving Average</i> – (Modelo Auto-Regressivo-Fracionário-Integrado-Médias-Móveis)
FBM	<i>Fractional Brownian Motion</i> – (Movimento Browniano Fracionário)
Fda	<i>Função de Distribuição Acumulada</i>
Fdp	<i>Função Densidade de Probabilidade</i>
FGN	<i>Fractional Gaussian Noise</i> – (Ruído Gaussiano Fracionário)
FPQ	<i>Flow Proportional Queueing</i> – (Fila Proporcional ao Fluxo)
FRED	<i>Flow Random Early Drop</i> – (Fluxo com Descarte Prematuro Aleatório)

FTP	<i>File Transfer Protocol</i> – (Protocolo de Transferência de Arquivo)
GMPLS	<i>Generalized Multi-Protocol Label Switching</i> – (Comutação Generalizada de Rótulo Multiprotocolo)
GOS	<i>Grade of Service</i> – (Grau ou Nível de Serviço)
HTTP	<i>Hipertext Transfer Protocol</i> – (Protocolo de Transferência de Hipertexto)
IC	<i>Indicador de Congestionamento</i>
ICIR	<i>The ICSI Center for Internet Research</i> – (Centro para Pesquisa da Internet)
ICSI	<i>International Computer Science Institute</i> – (Instituto Internacional de Ciência da Computação)
IEEE	<i>Institute of Electrical and Electronic Engineers</i> – (Instituto de Engenheiros Elétricos e Eletrônicos)
IETF	<i>Internet Engineering Task Force</i> – (Força Tarefa de Engenharia da Internet)
IGP	<i>Interior Gateway Protocol</i> – (Protocolo Interior do Gateway ou Roteador)
IN	<i>Intelligent Network</i> – (Rede Inteligente)
IntServ	<i>Integrated Services</i> – (Serviço Integrado)
IP	<i>Internet Protocol</i> – (Protocolo Internet)
IPP	<i>Interrupted Poisson Process</i> – (Processo de Poisson Interrompido)
IRP	<i>Interrupted Renewal Process</i> – (Processo de Renovação Interrompido)
IRTF	<i>Internet Research Task Force</i> – (Força Tarefa de Pesquisa da Internet)
ISDN	<i>Integrated Services Digital Network</i> – (Rede Digital de Serviços Integrados)
ITU-T	<i>International Telecommunications Union – Telecommunications Standardization Sector</i> – (União de Telecomunicações Internacionais – Setor de Padronização)
JSAC	<i>Journal on Selected Areas in Communications</i> – (Periódico em Áreas Seleccionadas em Comunicação)
LAN	<i>Local Area Network</i> – (Rede de Área Local)
LRD	<i>Long-Range Dependence</i> – (Dependência de Longa Duração)
MAC	<i>Media Access Control</i> – (Controle de Acesso ao Meio)
MAN	<i>Metropolitan Area Network</i> – (Rede de Área Metropolitana)
MIT	<i>Massachusetts Institute of Technology</i> – (Instituto Tecnológico de Massachusetts)
MMPP	<i>Markov Modulated Poisson Process</i> – (Processo de Poisson Modulado por Markov)
MPLS	<i>Multiprotocol Label Switching</i> – (Comutação de Rótulos Multiprotocolo)
NIC	<i>Network Interface Card</i> – (Cartão de Interface de Rede)
NMN	<i>Network and Mobile Systems</i> – (Rede e Sistemas Móveis)
NSF	<i>National Science Foundation</i> – (Fundação Nacional de Ciências)

N-ISDN	<i>Narrowband – Integrated Services Digital Network</i> – (Rede Digital de Serviços Integrados de Banda Estreita)
PoP	<i>Presence of Point</i> – (Ponto de Presença)
PSTN	<i>Public Switched Telephone Network</i> – (Rede Telefônica Comutada Pública)
QoS	<i>Quality of Service</i> – (Qualidade de Serviço)
QoS/R	<i>Quality of Service Routing</i> – (Qualidade de Serviço de Roteamento)
RDF	<i>Residual Distribution Function</i> – (Função de Distribuição Residual)
RED	<i>Random Early Detection</i> – (Detecção Prematura Aleatória)
RFC	<i>Request For Comments</i> – (Requisição de Comentários)
R/S	<i>Rescaled reajusted range Statistic</i> – (Estatística de Amplitude Reescalada Ajustada)
RSVP	<i>Resource ReSerVation Protocol</i> – (Protocolo de Reserva de Recursos)
RTCL	<i>Real Time Computing Lab</i> – (Laboratório de Computação em Tempo Real)
RTT	<i>Round Trip Time</i> – (Tempo de Viagem de Ida e Volta)
SBC	<i>Sociedade Brasileira de Computação</i>
SBRC	<i>Simpósio Brasileiro de Redes de Computadores</i>
SIGCOMM	<i>Special Interest Group on Data Communications</i> – (Grupo de Interesse Especial em Comunicações de Dados)
SFB	<i>Stochastic Fair Blue</i> – (Técnica Estocástica Justa)
SRD	<i>Short-Range Dependence</i> – (Dependência de Curta Duração)
SS	<i>Signaling Systems</i> – (Sistema de Sinalização)
SS7	<i>Signaling Systems 7</i> – (Sistema de Sinalização 7)
TCP	<i>Transmission Control Protocol</i> – Protocolo de Controle de Transmissão)
TE	<i>Traffic Engineering</i> – (Engenharia de Tráfego)
TES	<i>Transform Expand Example</i> – Exemplo de Expansão de Transformada)
UCP	<i>Unidade Central de Processamento</i>
UCSD	<i>University of California at San Diego</i> – (Universidade da Califórnia em San Diego)
UDP	<i>User Datagram Protocol</i> – (Protocolo de Datagrama de Usuário)
VBR	<i>Variable Bit Rate</i> – (Taxa de Bit Variável)
WAN	<i>Wide Area Network</i> – (Rede de Longa Distância)
WP	<i>Work Party</i> – (Grupo de Trabalho)
WWW	<i>World Wide Web</i> – (Rede Mundial)
4IPP	<i>Four Sources – Interrupted Poisson Process</i> – (Processo de Poisson Interrompido com 4 Fontes)

Resumo

As redes LAN e WAN têm mostrado um comportamento diferente do comumente assumido na modelagem estocástica de tráfego. Este novo padrão de comportamento observado pode ser modelado como processos estocásticos auto-similares para fins de modelagem de tráfego. O comportamento do tráfego das redes atuais é caótico, possuindo propriedades estatísticas que incluem correlações temporais complexas. Neste cenário, um problema ainda não resolvido é o do congestionamento do tráfego. Os algoritmos de controle de congestionamento implementados, por exemplo, no protocolo TCP são reativos e baseados unicamente no descarte de pacotes. Existem esquemas preventivos de controle de congestionamento que tratam o problema nas filas de saída dos nós da rede utilizando mecanismos de gerência ativa de filas, mas, no entanto, não evitam o uso da política de descarte. Nas últimas décadas, tem-se observado que considerável parcela das pesquisas em gerência de redes concentra-se na área de desempenho, principalmente em mecanismos de controle preventivo de congestionamento. Uma possível explicação para isso deve-se ao crescimento das aplicações nas redes, não só em quantidade, mas também em complexidade, e ao número de usuários que utilizam redes, particularmente a Internet.

Este trabalho propõe e implementa um mecanismo para prever possíveis congestionamentos, através da identificação da componente de tendência em séries de tempo que representam o tráfego real de redes, utilizando a transformada de *wavelet* discreta. O mecanismo proposto produziu o efeito desejado, pois detectou possíveis casos de congestionamento.

Abstract

LAN and WAN networks have showed a traffic behavior different from the commonly assumed in the traffic stochastic modelling. This new pattern of observed behavior motivates the use of self-similar stochastic processes for traffic modelling purposes. The behavior of the current networks traffic is chaotic, possessing statistical properties which include complex temporal correlations. In this scenario, a problem not yet solved is related to traffic congestion. The current congestion control algorithms implemented in the TCP protocol, for example, are reactive and based only on packets discard. There are preventive schemes of congestion control that treats with the problem at the output queues of the network nodes, using actives queues management mechanisms. However, they do not avoid the use of the discard policies. In the last decades, it has been observed that considerable research effort in network management has been concentrated on performance issues, mainly in preventive congestion control mechanisms. Main reasons for this is the growth of network applications, not only in amount but also in complexity, and to the number of users that use networks, particularly the Internet.

This work proposes and implements a mechanism for forecasting possible congestions through the identification of the tendency component in time series that represent the network real traffic, using discrete wavelet transform. The proposed mechanism produced the wanted effect, since can detect possible cases of congestion.

Capítulo 1

Introdução

O presente trabalho apresenta a tese, submetida como requisito parcial para obtenção do grau de doutor em Ciência da Computação do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE).

Este capítulo é uma introdução à tese. Inicialmente, a Seção 1.1 expõe o problema que está sendo resolvido, destacando a originalidade da solução. A seção seguinte, 1.2, apresenta o objetivo geral e os objetivos específicos e a relevância da pesquisa. A Seção 1.3 descreve os trabalhos relacionados com: (i) controle de congestionamento e engenharia de tráfego, destacando os principais grupos de pesquisa da área e as recomendações do IETF (*Internet Engineering Task Force*) e do ITU-T (*International Telecommunications Union – Telecommunications Standardization Sector*), (ii) modelagem de tráfego e (iii) o uso de *wavelets* na análise e modelagem do tráfego de redes. As contribuições e a estrutura da tese são apresentadas, respectivamente, nas Seções 1.4 e 1.5. Por fim, a Seção 1.6 encerra o capítulo com os comentários finais.

1.1 O Problema

Um fato indesejável que prejudica o desempenho das redes e constitui uma classe de problema, particularmente importante para engenharia de tráfego, é o do congestionamento. É um dos principais problemas das redes comutadas por pacotes como as redes IP. Este problema poderá se tornar crítico nos próximos anos devido ao crescimento do número de usuários, do número de aplicações, principalmente as de multimídia, e da complexidade das mesmas. Awduche (1999) descreve que congestionamento pode ter duas causas básicas: a insuficiência para acomodar o fluxo de tráfego presente e o excesso entre a taxa de chegada de pacotes e a taxa de serviço em um ou mais nós da rede. Tal excesso ocasiona desbalanceamento do tráfego em nós da rede, onde um conjunto de recursos fica sobrecarregado, enquanto outro conjunto de recursos é sub-utilizado.

O pensamento dominante há quinze anos atrás era que os congestionamentos poderiam ser resolvidos somente com um incremento significativo das velocidades de transmissão dos enlaces (canais), com um acréscimo do poder de processamento dos nós de comunicação e

com a utilização de grandes *buffers* para o armazenamento de pacotes. Porém, Jain (1990) mostrou que esses procedimentos, por si só, não constituem uma solução eficiente.

Com a chegada da Internet, surgiu o problema conhecido como “Colapso por Congestionamento”, que reduziu significativamente o desempenho das redes em situações de congestionamento (JACOBSON; KARELS, 1988). Tal problema foi resolvido através da proposta de algoritmos, os quais efetuam o controle do congestionamento. Todavia, a Internet continua, nos dias de hoje, sua expansão em tamanho, diversidade e alcance, desempenhando um papel sempre crescente na integração de outras redes de acesso. Atualmente, a ocorrência de congestionamento na Internet é devida, principalmente, à natureza imprevisível e caótica do seu tráfego.

Basicamente, os atuais mecanismos de controle de congestionamento podem ser classificados em dois grupos: o primeiro aumenta a disponibilidade dos recursos através da reconfiguração dinâmica deles e o segundo, mais usado, reduz a demanda ao nível da disponibilidade dos recursos.

No primeiro grupo os usuários dos recursos não precisam estar cientes do congestionamento. Somente a rede é responsável pela solução. Esta abordagem é difícil de ser implementada e os custos envolvidos, por serem altos, desestimulam sua utilização.

No segundo grupo os mecanismos são divididos em: (i) aqueles que descartam pacotes, e (ii) aqueles que limitam a taxa de entrada de pacotes. Os que descartam pacotes devem ser evitados, porque geram uma situação que contraria o próprio mecanismo, na medida que os pacotes descartados deverão ser retransmitidos pelo usuário aumentando, conseqüentemente, o congestionamento. A outra alternativa, que é a de limitar a taxa com que novos pacotes são passados para a rede, também deve ser evitada. Sua falha está em não tratar os casos nos quais o congestionamento é gerado pelos vários nós de forma simultânea e propagado por toda a rede. Por esta razão, além do controle local, existe a necessidade de se fazer um controle de congestionamento a nível global da rede.

As alternativas levantadas até aqui não tratam o problema de forma pró-ativa, pois os mecanismos são acionados somente após a ocorrência do congestionamento. Uma das formas de controlar o congestionamento sem o desperdício de recursos é prever o comportamento do tráfego e as situações de congestionamento de modo que se possa tomar medidas de controle, que vão desde o descarte de pacotes ao provisionamento de largura de banda adicional nos canais.

Existem duas atividades distintas relacionadas com o controle de congestionamento: prevenção e recuperação (JACOBSON; KARELS, 1988; JAIN, 1992; LEFELHOCZ et al., 1996). A primeira trata de detectar possíveis condições que levem a congestionamentos futuros e de executar procedimentos para impedi-los. A segunda atua na rede após a ocorrência de congestionamento para resolver o problema. Assim, alternativas de controle que antecipam situações de congestionamento devem incluir modelos que analisem o comportamento de um tráfego, muitas vezes caótico.

As deficiências dos controles de congestionamento anteriormente citadas, a ausência de mecanismos preventivos que antecipam situações de congestionamento e a complexidade envolvida nos atuais modelos de tráfego motivaram uma nova abordagem para o problema, objeto central desta tese, baseada na análise do comportamento do tráfego e em técnicas de estimação de tendências associadas a métodos de transformadas de *wavelet* (DAUBECHIES, 1992).

A tese foi desenvolvida seguindo três princípios. Primeiro, como a maioria dos congestionamentos ocorre em nós intermediários (roteadores, *switches*, etc.), estes devem ser os únicos responsáveis pela monitoração da carga e do congestionamento da rede. Segundo, como o congestionamento somente é percebido após a sua ocorrência pelos atuais controles, deve-se tratá-lo de forma preventiva. Por fim, como a circunstância do congestionamento tem característica estocástica, deve-se considerar as diversas formas de comportamento do tráfego (rajada, contínuo com taxa constante e contínuo com taxa variável) e as chances com que cada tipo de tráfego contribui para a ocorrência do problema.

1.2 Objetivo Geral da Tese

Propor, especificar e implementar um mecanismo para identificar situações que podem causar congestionamento de tráfego em redes, baseado nas funções *wavelets* e nas suas transformadas discretas e inversas, e em técnicas de estimação de tendências de séries temporais.

1.2.1 Objetivos Específicos

- a) Estudar os atuais mecanismos reativos e preventivos de controle de congestionamento de redes;

- b) Identificar situações e fatores, além do aumento da carga de tráfego, que favorecem a ocorrência de congestionamento e as possibilidades de cada um deles;
- c) Analisar modelos estocásticos, principalmente processos auto-similares, para modelar tráfego heterogêneo de redes;
- d) Investigar modelos estocásticos de previsão e suas aplicabilidades no prognóstico de tráfego de redes;
- e) Identificar e analisar processos que podem descrever comportamentos caóticos em tráfego de rede;
- f) Investigar modelos de análise e previsão de série temporais de tráfego no domínio do tempo e da frequência;
- g) Investigar o papel das *wavelets*, com ênfase na transformada discreta direta e inversa, para análise, estimação e síntese de séries temporais.
- h) Especificar e implementar um mecanismo para realizar prognóstico de congestionamento de redes através de técnicas de transformação de séries de dados no domínio *wavelet*;
- i) Validar o mecanismo em termos da corretude e do tempo necessário para a realização do prognóstico.

1.2.2 Relevância

Congestionamento de redes é um problema real que vem sendo estudado por vários grupos de pesquisa. Possivelmente, não será solucionado em um futuro próximo, pois a demanda pelos recursos da rede, às vezes escassos, quase sempre é maior que a oferta. Por exemplo, as redes de acesso à Internet são susceptíveis ao problema do congestionamento, pois são veículos para transferência de áudio, vídeo, imagem e dados, através das redes de comunicação. As redes de acesso à Internet estão entre as de maior demanda por parte dos usuários e de maior consumo de recursos.

O problema do congestionamento é agravado com o crescimento do uso de aplicações multimídia de tempo real com requisitos estritos de atraso. Apesar do aumento das taxas de transmissão dos canais de longa distância, o problema do congestionamento ainda persiste, sobretudo nas redes de acesso.

Portanto, o problema é relevante e precisa de soluções inovadoras para resolvê-lo, principalmente em redes IP, onde este protocolo é utilizado pela maioria das redes e constitui a base da Internet.

1.3 Trabalhos Relacionados

A revisão bibliográfica sobre o assunto está aqui dividida em três partes. A primeira relaciona as pesquisas e os trabalhos de controle de congestionamento e engenharia de tráfego, salientando os principais grupos de estudo da área e as recomendações do IETF e do ITU-T. A segunda descreve os artigos sobre modelagem de tráfego, sobressaindo os modelos auto-similares, e a última mostra os trabalhos que descrevem a função das *wavelets* na análise e modelagem do tráfego de redes.

1.3.1 Revisão Bibliográfica sobre Congestionamento e Engenharia de Tráfego

Pesquisas teóricas sobre congestionamento têm utilizado cenários do tipo um único engarrafamento na rede ou uma fila por fluxo, e realizado simulações para estudar e validar projetos na área de redes (BENMOHAMED; MEERKOV, 1993; OZBAY; KALYANARAMAN; IFTAR, 1998; MASCOLO, 1999; SRIKANT, 2000). Tudo isso tem evidenciado a diferença entre o comportamento real das redes e o simulado.

Todavia, nos últimos anos, progressos têm sido realizados com o uso de modelos analíticos na modelagem do controle de congestionamento (KELLY, 1999; LOW; LAPSLEY, 2000; KUNNIYUR; SRIKANT, 2000; MO; WALRAND, 2000; MISRA; GONG; TOWSLEY, 2000; HOLLOT et al., 2001).

Existem grupos, a maioria sediada em universidades, que estudam as questões relacionadas com congestionamento e como a engenharia de tráfego pode solucionar problemas das redes. O problema do congestionamento está vinculado diretamente com o estudo das características do tráfego, pois estas influem na efetividade dos mecanismos de controle de congestionamento, bem como no dimensionamento dos recursos. O Quadro 1.1 lista os principais grupos de pesquisa na área de engenharia de tráfego e de controle de congestionamento de redes.

Quadro 1.1. Grupos de Pesquisa

Instituição / Grupo	Título do Projeto / Pesquisa	Líder(es) da Pesquisa
Harvard Networking Group	• High-Speed Networks Management	H. T. Kung
Cornell Networking Research Group (CNRG)	• Network Performance Management	S. Keshav
The ICSI Center for Internet Research (ICIR)	• RED (Random Early Detection), • ECN (Explicit Congestion Notification)	Sally Floyd, Vern Paxson, Scott Shenker
U. of Michigan, Real Time Computing Lab (RTCL)	• Sensitive Networks Flow and Congestion Control	Kang G. Shin
MIT Network and Mobile Systems (NMN) Group	• CM – Congestion Manager	Hari Balakrishnan
U. Massachusetts Computer Network Research Group	• Fluid Models for Large Heterogeneous Networks	Jim Kurose, Don Towsley
U. California San Diego (UCSD)	• CAIDA (Cooperative Association for Internet Data Analysis)	
LAND / UFRJ	• ALMADEM e outros	Edmundo S. e Silva
GTA / UFRJ e UNICAMP	• QUARESMA	Otto C. M. B. Duarte

O grupo de pesquisa de redes da Universidade de Harvard está construindo uma estrutura de conexão de alto-desempenho com a Internet para dar apoio a vários projetos. A conexão será utilizada em pesquisas que incluem química atmosférica, astrofísica, bibliotecas digitais, medicina, ciências sociais e redes de dados. Este trabalho está também ajudando a criar um grande ponto de presença (gigaPoP) na cidade de Boston, para prover serviços da próxima geração da Internet em instituições de pesquisa e educacionais. Paralelamente, estão sendo realizadas pesquisas sobre engenharia de tráfego para a construção desta rede de alto-desempenho. Informações sobre o grupo e suas pesquisas são encontradas em: <http://www.deas.harvard.edu/research/computerscience/netandsys/index.html>.

O grupo de pesquisa de redes da Universidade de Cornell (CNRG) trabalha no projeto *Network Performance Management*, que investiga aspectos relacionados com a infra-estrutura da Internet. O grupo tem interesse em protocolos e algoritmos teóricos aplicáveis à Internet de hoje. As pesquisas estão concentradas em gerenciamento de desempenho de redes, topologia,

algoritmos e aplicações multiplanares. Informações sobre o grupo estão no *site* <http://cs.cornell.edu/cnrg>.

A AT&T criou um instituto de pesquisa, chamado ICSI, que funciona em Berkeley, Califórnia. Até o ano 2001, o grupo era conhecido como ACIRI e, a partir daí, passou a se chamar ICIR. Os objetivos do ICIR são: (i) fazer pesquisa sobre a arquitetura Internet e aspectos relacionados à rede, (ii) participar ativamente das pesquisas do SIGCOMM (*Special Interest Group on Data Communication*) e IRTF (*Internet Research Task Force*) e das comunidades de padronização do IETF, (iii) fazer a ligação entre as comunidades acadêmicas de pesquisa da Internet e as de interesse comercial, através da promoção de fóruns para discussão de tópicos técnicos de interesse mútuo. Dentre os projetos de envergadura desenvolvidos, dois são conhecidos pelos nomes de seus mecanismos: RED e ECN. O grupo mantém um *site* em <http://www.aciri.org>.

O mecanismo RED é atualmente o mais difundido e implementado em roteadores com o protocolo TCP/IP. Desde que foi originalmente apresentado (FLOYD; JACOBSON, 1993), várias pesquisas foram realizadas com o propósito de sugerir valores para seus parâmetros (FLOYD, 2000a; ROSOLEN et al., 1999), recomendar o seu uso (BRADEN et al., 1998; FLOYD; FALL, 1999; FLOYD, 2000b) e fazer avaliações (MAY; BONALS; BOLOT, 2000; CHRISTIANSEN et al., 2000; FLOYD, 2001). O Capítulo 2 fornece mais detalhes sobre o RED. O ECN é um mecanismo mais recente descrito em detalhes em Ramakrishanan; Floyd (1999) e Ramakrishanan; Floyd; Black (2001). Avaliações desse mecanismo são encontradas em Bagal et al. (1999) e Salin; Ahmed (2000).

O grupo de pesquisa da Universidade de Michigan desenvolve o projeto denominado “*Sensitive Networks Flow and Congestion Control*”. O trabalho está relacionado com um dos aspectos mais pesquisados em projetos de redes de longa distância que é o controle de fluxo, essencial para o desempenho de serviços garantidos e para o uso flexível e eficiente dos recursos da rede. A pesquisa desenvolve esquemas de controle de fluxo e de congestionamento que: (i) maximizam a utilização da largura de banda; (ii) limitam os atrasos fim-a-fim; (iii) evitam o estouro de *buffer* e (iv) minimizam os requisitos de *buffer*. O grupo tem trabalhado com gerenciamento ativo de filas para controle de congestionamento utilizando os algoritmos Blue e Fair Blue (FENG et al., 2001a; FENG et al., 2001b), em protocolos de transporte baseados no controle da taxa (ZHANG; SHIN, 2001) e em controle

de fluxos de serviços ABR de redes ATM (ZHANG et al., 2002). Informações sobre a pesquisa do grupo estão no *site*: <http://kabru.eecs.umich.edu>.

O grupo de pesquisa do MIT desenvolve um *framework*, denominado CM, para gerenciar e controlar congestionamentos de redes e compartilhar largura de banda, que funciona de forma independente do protocolo de transporte e das aplicações. O protocolo TCP incorpora mecanismos de controle de congestionamento e é, em grande parte, responsável pela estabilidade da Internet. Porém, existem situações onde o TCP falha: (i) quando aumenta o número de aplicações que utilizam o protocolo UDP (*User Datagram Protocol*) e (ii) quando existem muitos fluxos concorrentes que competem entre si pelos recursos da rede. Motivado por este problema, o grupo propôs uma nova arquitetura ao redor do CM. Esta arquitetura tem a capacidade de tratar fluxos concorrentes de tráfego e fluxos baseados no protocolo UDP que não se adaptam às mudanças da rede. Este trabalho (BALAKRISHNAN; SESHAN, 2001) está registrado no IETF como um padrão proposto. Mais informações sobre o grupo estão no *site* <http://nms.lcs.mit.edu>.

O objetivo do grupo de pesquisa da Universidade de Massachussets é desenvolver novas metodologias, baseadas nos modelos de fluídos, para que grandes redes heterogêneas possam manusear vários fluxos concorrentes baseados no TCP e no UDP. Essas metodologias provêm soluções eficientes e rápidas que determinam o desempenho, atrasos médios e taxas de perda de fluxos individuais e agregados em redes de longa distância. Além da análise de fluxos UDP e TCP, a metodologia favorece que pesquisadores avaliem o desempenho de novos protocolos de rede, como o *multicast*, e o impacto desses protocolos em fluxos TCP legados. O projeto é financiado pelo DARPA. Informações sobre a pesquisa do grupo podem ser encontradas em <http://gaia.cs.umass.edu>.

A CAIDA é uma associação de organizações governamentais e comerciais e de setores de pesquisa, que promove a cooperação na área de engenharia e manutenção de uma infraestrutura global da Internet. Os objetivos da CAIDA são: (i) encorajar a criação de métricas de tráfego da Internet e assegurar suas utilidade e aceitação universal, (ii) criar um ambiente de pesquisa colaborativa em que várias formas de dados de tráfego podem ser adquiridas, analisadas e compartilhadas, e (iii) promover o desenvolvimento de técnicas e metodologias para análise, visualização, simulação, caracterização de fluxo e desempenho de tráfego. Novas áreas específicas de pesquisa incluem diagnóstico de roteamento em tempo-real, evolução

para medições da próxima geração da Internet e protocolos de roteamento *multicast* e *unicast*. Informações sobre a CAIDA estão disponíveis em <http://www.caida.org>.

O Laboratório para Modelagem, Análises e Desenvolvimento de Redes (LAND) da Universidade Federal do Rio de Janeiro (UFRJ), localizado na COPPE, faz pesquisas cujos principais tópicos são: (i) desenvolvimento de novos algoritmos para tratar com modelos de sistemas de comunicação, (ii) desenvolvimento de ferramentas para análises de desempenho e confiabilidade de sistemas de comunicação e de computadores e (iii) desenvolvimento de modelos para tráfego e mecanismos de rede multimídia. Um dos produtos de pesquisa do grupo, a ferramenta TANGRAM-II, é um ambiente para modelagem e experimentação de sistemas de comunicação e de computadores, desenvolvido para fins de pesquisa e educacionais. Esta ferramenta provê uma interface baseada no paradigma orientado a objeto e uma série de soluções para se obter as medições de interesse. O ambiente inclui módulos de experimentação de redes, modelagem de tráfego e ferramentas multimídia que auxiliam o processo de modelagem de redes. O *site* do grupo está em <http://araruama.land.ufrj.br>.

O projeto Quaresma (Qualidade de Serviço em Redes, Segurança, Mobilidade e Aplicações) vem sendo desenvolvido por professores e alunos da UFRJ, UNICAMP, UFF e da Universidade de Paris 6, com o apoio do CNPq. O grupo está desenvolvendo soluções para integração e interação de componentes de *middleware* com recursos e mecanismos de redes e com as aplicações. Entre as atividades do grupo, destacam-se: o desenvolvimento de um gerador de tráfego Web parametrizável para comparar resultados simulados com medidas reais, a investigação da natureza multifractal do tráfego Internet e a investigação da efetividade de mecanismos de gerenciamento ativo de filas e de notificação explícita de congestionamento (ECN). Informações sobre a pesquisa do grupo são encontradas em <http://www.gta.ufrj.br/quaresma>.

Além desses grupos de pesquisa, existem organismos internacionais, como IETF, que procuram manter um registro do que está sendo feito nas áreas de engenharia de tráfego e controle de congestionamento. O IETF mantém uma página na WEB (<http://www.ietf.org>) com as recomendações e propostas nestas áreas. O Quadro 1.2 mostra as recomendações do IETF sobre controle de congestionamento e o Quadro 1.3 as relacionadas com engenharia de tráfego. As recomendações nestes quadros estão dispostas em ordem cronológica inversa.

Quadro 1.2. Recomendações do IETF sobre controle de congestionamento

Número	Título	Autor	Data	Status
RFC3649	HighSpeed TCP for Large Congestion Windows	S. Floyd	Dezembro 2003	Experimental
RFC3540	Robust Explicit Congestion Notification (ECN) Signaling with Nonces	N. Spring, D. Wetherall, D. Ely	Junho 2003	Experimental
RFC3465	TCP Congestion Control with Appropriate Byte Counting (ABC)	M. Allman	Fevereiro 2003	Experimental
RFC3168	The Addition of (ECN) Explicit Congestion Notification to IP	K. Ramakrishnan, S. Floyd, D. Black	Setembro 2001	Padrão proposto
RFC3124	The Congestion Manager (CM)	H. Balakrishnan, S. Seshan	Junho 2001	Padrão proposto
RFC2914	Congestion Control Principles	S. Floyd	Setembro 2000	Informativo
RFC2884	Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks	J. Hadi Salim, U. Ahmed	Julho 2000	Informativo
RFC2861	TCP Congestion Window Validation	M. Handley, J. Padhye, S. Floyd	Junho 2000	Experimental
RFC2581	TCP Congestion Control	M. Allman, V. Paxson, W. Stevens	Abril 1999	Padrão proposto
RFC2309	Recommendations on Queue Management and Congestion Avoidance in the Internet	B. Braden, D. Clark, B. Davie, J. Crowcroft, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang	Abril 1998	Informativo
RFC1254	Gateway Congestion Control Survey	A. Mankin, K. Ramakrishnan	Agosto 1991	Informativo

A RFC 1254 do IETF faz uma revisão sobre controles de congestionamento em *gateway*. A RFC 2581 define quatro algoritmos de controle de congestionamento (*slow start*, *congestion avoidance*, *fast retransmit* e *fast recovery*) para o TCP e substitui a de referência 2001. A RFC 2309 recomenda o gerenciamento ativo de filas em roteadores para melhorar o desempenho da Internet. As recomendações 2861, 3465 e 3649 tratam da validação e de modificações do tamanho da janela de congestionamento do TCP. Já a RFC 2914 descreve os princípios gerais de controle de congestionamento e a 3124 um *framework* para controlar congestionamentos. Por fim, as recomendações 2884, 3168 e 3540 descrevem o mecanismo ECN e avaliam seu desempenho.

Quadro 1.3. Recomendações do IETF sobre engenharia de tráfego

Número	Título	Autor	Data	Status
RFC3564	Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering	F. Le Faucheur, W. Lai	Julho 2003	Informativo
RFC3477	Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)	K. Kompella, Y. Rekhter	Janeiro 2003	Padrão proposto
RFC3473	Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions.	L. Berger	Janeiro 2003	Padrão proposto
RFC3346	Applicability Statement for Traffic Engineering with MPLS	J. Boyle, V. Gill, A. Hannan, D. Cooper, D. Awduche, W.S. Lai B. Christian	Agosto 2002	Informativo
RFC3272	Overview and Principles of Internet Traffic Engineering	D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao	Mai 2002	Informativo
RFC2702	Requirements for Traffic Engineering Over MPLS	D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus	Setembro 1999	Informa

A recomendação 3272 descreve os princípios da engenharia de tráfego na Internet. As RFC 2702 e 3346 tratam dos requisitos e da aplicabilidade da engenharia de tráfego com MPLS. As recomendações 3473 e 3477 descrevem a engenharia de tráfego com RSVP e a 3564 apresenta os requisitos de Serviços Diferenciados com engenharia de tráfego.

Uma outra organização internacional, ITU-T, responsável pela produção de padrões internacionais na área de redes de telecomunicações e de longa distância, também mantém grupos de trabalho para elaborar recomendações relacionadas à engenharia de tráfego. A série “E” produzida pelo grupo de estudo WP 3/2 do ITU-T constitui o corpo principal das recomendações na área de engenharia de tráfego. As principais tarefas da engenharia de tráfego para o ITU-T compreendem: (i) caracterização de demanda de tráfego, (ii) controle e dimensionamento de tráfego e (iii) monitoração de desempenho. Os Quadros 1.4, 1.5, 1.6, 1.7 e 1.8 relacionam essas recomendações.

A caracterização de demanda de tráfego é feita através de modelos que aproximam o comportamento do tráfego da rede em grandes populações de usuários. Esta tarefa está subdividida em: modelagem, medição e previsão de tráfego. O controle e dimensionamento de tráfego são entradas no processo de engenharia de tráfego para o projeto e a operação da rede. O dimensionamento da rede tem por objetivo assegurar recursos suficientes para atender a demanda de tráfego e os requisitos de qualidade de serviço (QoS), enquanto que o controle assegura que o desempenho da rede seja mantido mesmo em condições de sobrecarga e de falhas. Por fim, a tarefa de monitoração de desempenho é realizada para corrigir situações de sobrecarga ou de falhas não consideradas no dimensionamento do tráfego.

Quadro 1.4. Recomendações do ITU-T para modelagem de tráfego

Nº. Rec.	Data	Título
E.711	10/92	User demand modelling
E.712	10/92	User plane traffic modelling
E.713	10/92	Control plane traffic modelling
E.716	10/96	User demand modelling in Broadband-ISDN
E.760	03/00	Terminal mobility traffic modelling

As recomendações E.711 e E.716 explicam como caracterizar conexões em redes digitais de serviços integrados (ISDN), através de um conjunto de atributos (por exemplo, modo de transferência da informação, configuração da comunicação, taxa de transferência, requisitos de QoS, etc.) e de variáveis de tráfego (número de eventos que ocorre ao longo de uma conexão e os tempos entre esses eventos). As recomendações E.712 e E.713 explicam como modelar o tráfego e a E.760 trata do problema de modelagem de tráfego em redes móveis.

Quadro 1.5. Recomendações do ITU-T para medição de tráfego

Nº. Rec.	Data	Título
E.490	06/92	Traffic measurement and evaluation – general survey
E.491	05/97	Traffic measurement by destination
E.500	11/98	Traffic intensity measurement principles
E.501	05/97	Estimation of traffic offered in the network
E.502	02/01	Traffic measurement requirements for digital telecommunication Exchanges
E.503	06/92	Traffic measurement data analysis
E.504	11/88	Traffic measurement administration
E.505	06/92	Measurements of the performance of common channel signalling Network
E.743	04/95	Traffic measurements for SS No.7 dimensioning and planning
E.745	03/00	Cell level measurement requirements for the B-ISDN

As recomendações E.490, E.491, E.503 e E.504 fazem referência aos aspectos mais gerais e operacionais do tráfego, como a utilidade da medição (descrita na E.491), os procedimentos operacionais para executar medições (E.504) e a análise dos dados medidos (E.503). A E.500 e E.501 descrevem aspectos mais técnicos, como os critérios para se medir a intensidade do tráfego e os métodos de estimação. As recomendações E.502, E.505, E.743 e E.745 tratam dos requisitos para se medir tráfego e desempenho em redes específicas, como as redes telefônicas tradicionais (PSTN), as redes digitais de serviços integrados de banda estreita e de banda larga (N-ISDN e B-ISDN) e as redes de sinalização de canal comum (CCSN).

Quadro 1.6. Recomendações do ITU-T para previsão de tráfego

Nº. Rec.	Data	Título
E.506	06/92	Forecasting international traffic
E.507	11/88	Models for forecasting international traffic
E.508	10/92	Forecasting new telecommunication services

As recomendações E.506 e E.507 estão relacionadas à previsão de serviços tradicionais para os quais existem dados históricos. A E.506 orienta na escolha dos dados (de tráfego, econômicos, sociais e demográficos) para previsão e a E.507 provê uma introdução das técnicas matemáticas existentes para previsão, tais como ajustamento de curvas, modelos auto-regressivos e modelos ARIMA (BOX; JENKINS, 1976). Embora ambas façam referência ao tráfego internacional, também podem ser aplicadas ao tráfego de um país. A recomendação E.508 trabalha com previsão de novos serviços de telecomunicações para os quais não existem dados históricos.

Quadro 1.7. Recomendações do ITU-T para controle e dimensionamento de tráfego

Nº. Rec.	Data	Título
E.735	05/97	Framework for traffic control and dimensioning in B-ISDN
E.736	05/97	Methods for cell level traffic control in B-ISDN
E.737	05/97	Dimensioning methods for B-ISDN

As recomendações para dimensionamento e controle de tráfego podem ser classificadas como: as que tratam de redes comutadas de pacotes (listadas no Quadro 1.7), as voltadas para redes comutadas de circuitos e as que se referem às redes estruturadas inteligentes (IN) e de sinalização (SS nº. 7). As voltadas para redes comutadas de pacotes descrevem métodos que podem ser aplicados em redes B-ISDN, utilizando tecnologia ATM. Todavia, a maioria desses métodos também pode ser aplicada em outras redes, como as baseadas no protocolo IP, nas quais é possível controlar a admissão de conexões. A recomendação E.735 fornece conceitos sobre controle de admissão de conexão (CAC) em redes B-ISDN, define o que é conexão e recurso e analisa estratégias para configuração de redes lógicas. A E.736 e E.737 descrevem diversos métodos para controle e dimensionamento de tráfego em redes B-ISDN.

Quadro 1.8. Recomendações do ITU-T para monitoração de desempenho

Nº. Rec.	Data	Título
E.492	02/96	Traffic reference period
E.493	02/96	Grade of service (GOS) monitoring

As recomendações E.492 e E.493 descrevem parâmetros de medição de desempenho. Existem outras recomendações do ITU-T não classificadas nos quadros citados, como a E.600, que descrevem os termos e definições da engenharia de tráfego. Atualmente, o ITU-T trabalha em recomendações com ênfase na área de engenharia de tráfego para comunicações pessoais, para redes SS7 e para redes baseadas no protocolo IP.

1.3.2 Revisão Bibliográfica sobre Modelagem de Tráfego

Apesar de Mandelbrot (1982) ter lançado as primeiras noções sobre auto-similaridade em 1982, somente a partir de 1993, com o artigo de Leland et al. (1993), as pesquisas sobre o comportamento do tráfego de redes tomaram um outro rumo. Até então, os modelos que existiam tentando explicar o comportamento do tráfego, eram os modelos de renovação e de Markov (ROSS, 1985), o modelo de fluidos (JAGERMAN; MELAMED; WILLINGER, 1996), os modelos estocásticos lineares (BOX; JENKINS, 1976) e o modelo TES (*Transform Expand Sample*) (MELAMED, 1993).

Os antigos paradigmas de explicar o comportamento do tráfego de redes através das leis Poissonianas, ou de forma mais geral obedecendo regras Markovianas, foram colocados num cenário altamente questionável após o trabalho de Leland et al. (op.cit.) e a partir daí as pesquisas sobre modelagem de tráfego, segundo Park; Willinger (2000), podem ser classificadas em quatro categorias: (i) modelagem baseada em medições, (ii) modelagem física, (iii) análise de filas e (iv) controle de tráfego.

Os trabalhos de Garret; Willinger (1994), Huang et al. (1995), Leland et al. (1994) e Willinger et al. (1997), enquadrados na primeira categoria, utilizaram *traces* de tráfego de redes para detectar, identificar e quantificar características pertinentes, como a auto-similaridade. Em particular, Leland et al. (op.cit.) demonstraram a auto-similaridade em um ambiente LAN *Ethernet*. Paxson; Floyd (1995) mostraram manifestação de auto-similaridade no tráfego IP de redes WAN e Crovella; Bestavros (1997) no tráfego WWW. Esses trabalhos constituem, coletivamente, fortes evidências de que a invariância do tráfego em diferentes

escalas de tempo não é um fenômeno isolado, mas uma característica persistente em vários ambientes de redes. Artigos como os de Willinger; Paxson (1998a) e Willinger; Paxson (1998b) têm focado a pesquisa na diversidade e no volume das medições de tráfego, enquanto outros têm utilizado inferência estatística na caracterização do tráfego auto-similar (GILBERT; WILLINGER; FELDMANN, 1999; TAQQU; TEVEROVSKY; WILLINGER, 1995).

Os trabalhos classificados na segunda categoria tentam explicar as causas físicas da auto-similaridade baseadas nos mecanismos de rede, como protocolos e algoritmos de controle de fluxo e de congestionamento, e nas propriedades de sistemas distribuídos, estabelecidas empiricamente. O padrão de chegada do tráfego gerado por uma única fonte de dados, como por exemplo vídeo com uma taxa de bit variável (VBR), é a causa mais comum (BERAN et al., 1995; GARRET; WILLINGER, 1994). Uma segunda causa, de origem estrutural, é encontrada em tráfego no qual as fontes geradoras produzem arquivos e objetos com tamanhos que podem ser ajustados a uma distribuição de cauda pesada (PARK; KIM; CROVELLA, 1996). O modelo ON/OFF de Willinger et al. (1997) mostra que a superposição de um número grande de fontes independentes com períodos ON/OFF de cauda pesada conduz à auto-similaridade do processo agregado. O modelo ON/OFF tem sua raiz em um processo de renovação introduzido por Mandelbrot (1969) e estudado depois por Taqqu; Levy, 1996), que provê os fundamentos teóricos de trabalhos recentes em modelagem física de tráfego de redes, como os de Baugh (2000), Huang (2000) e Baugh et al. (2001). A fundamentação teórica, juntamente com a evidência empírica de distribuições de cauda pesada em períodos ON/OFF, explica a causalidade física da auto-similaridade, distinguindo o modelo ON /OFF de outros modelos matemáticos de tráfego auto-similar.

Na terceira categoria estão os trabalhos que provêm modelos matemáticos para modelar tráfego dependente de longa duração ou de longo alcance no âmbito da teoria das filas (ADAS; MUKHERJEE, 1995; ADDIE; ZUKERMAN; NEAME, 1995; LIKHANOV et al., 1995; NORROS, 1994; TSYBAKOV; GEORGANAS, 1998). Esses trabalhos investigaram o comportamento das filas com entrada dependente de longa duração e constataram que a distribuição do tamanho da fila em sistemas de *buffer* infinito decai lentamente em contraste com o decaimento exponencial da fila com entrada dependente de curta duração. De fato, dependendo do modelo de fila considerado, uma entrada dependente de longa duração pode

dar origem a uma distribuição de Weibull (NORROS, 1994) ou polinomial (TSYBAKOV; GEORGANAS, 1998).

Por fim, na quarta categoria estão os trabalhos relacionados com controle de congestionamento de tráfego auto-similar. Tuan; Park (1999) e Tuan; Park (2000) mostraram que a estrutura de correlação do tráfego em grandes escalas de tempo pode ser utilizada para obter ganhos de desempenho de processamento, utilizando tanto o controle de congestionamento do TCP quanto o controle baseado em taxa.

1.3.3 Revisão Bibliográfica sobre *Wavelets* na Modelagem de tráfego

Apesar da teoria sobre a transformada de *wavelet* ter sido estabelecida para processos determinísticos finitos, a literatura tem mostrado que essa transformada pode ser aplicada a processos estocásticos (MASRY, 1993). Particularmente, Delbeke (1998) e Pesquet-Popescu (1999) demonstraram que os coeficientes de *wavelets* reproduzem processos auto-similares através de determinadas propriedades. Pode ser visto em Flandrin (1993) que a correlação entre seus coeficientes em diferentes posições é menor que a do processo que lhe deu origem. Flandrin (op. cit.) mostrou que o decaimento das correlações pode ser controlado pelo aumento do tamanho da série. Os processos dependentes de longo alcance ou estacionários de segunda ordem são também reproduzidos por *wavelets* (ABRY; GONÇALVES; FLANDRIN, 1995).

Abry et al. (2002) desenvolveram técnicas baseadas em *wavelet* para estimar o parâmetro de Hurst (HURST, 1955), isto é, o grau de auto-similaridade. Existe também um sistema, baseado em *wavelet*, que estima o parâmetro de Hurst de modo *online* em um roteador, para fazer modificações adaptativas no controle de congestionamento ou para prover informações atualizadas sobre a dinâmica do tráfego (ROUGHAN, M.; VEITCH, D.; ABRY, P., 1998).

A introdução de técnicas baseadas em *wavelets* para analisar *traces* de tráfego (ABRY; VEITCH, 1998) representa um passo na direção do desenvolvimento de alternativas de modelagem, em especial, em tráfego não estacionário. Riedi et al. (1999) mostraram simulações de tráfego de rede usando *wavelets*, com resultados que assemelham-se visualmente (usando gráficos) e estatisticamente com *traces* reais de tráfego *Ethernet*.

Qiao; Skicewicz; Dinda (2003) estudaram empiricamente a previsibilidade de coeficientes *wavelet*, gerados a partir de tráfego real, para uso na determinação de tempos de transferência

de mensagens. O sistema WIND (HUANG; FELDMANN; WILLINGER, 2001) usa análises de escala baseada em *wavelet* para detectar problemas de desempenho de rede.

Uma classe de processos, chamada de multifractais, tem sido proposta por Peltier; Lévy-Véhel (1997) e relacionada com *wavelet* por Feldmann; Gilbert; Willinger (1998) que, na prática, investigaram a natureza multifractal do tráfego Internet e utilizaram *wavelets* para detectar, identificar e descrever comportamentos caóticos.

As causas físicas da auto-similaridade no tráfego de rede e o impacto da propriedade multifractal em mecanismos de controle de congestionamento do protocolo TCP foram investigadas em Feldman et al. (1999). Ma; Ji (2001) modelaram o tráfego auto-similar gerado por essas causas no domínio da *wavelet*.

1.4 Contribuições

Esta tese apresenta uma nova abordagem para prognósticos de congestionamentos que pode ser utilizada por mecanismos de controle de congestionamento situados nos nodos intermediários das redes. A concepção do mecanismo consiste em usar técnicas de transformação de séries temporais, partindo do princípio que o tráfego das redes não é estacionário e apresenta forte autocorrelação temporal e propriedade auto-similar. A idéia é utilizar *wavelets*, ou seja, ao invés de prever o comportamento do tráfego diretamente no domínio do tempo, modelam-se as propriedades estatísticas da série no domínio *wavelet* e, em seguida, utilizam-se técnicas de estimação de tendências para indicar possíveis casos de congestionamento.

A motivação principal na utilização de *wavelet* para prognóstico é a sua capacidade em reduzir a autocorrelação temporal dos coeficientes obtidos pela transformada, favorecendo a alteração e a aplicação deles na reconstrução do sinal. Os coeficientes de *wavelet* apresentam uma autocorrelação menor que a autocorrelação dos processos temporais que lhe deram origem. Este resultado foi obtido empiricamente aqui na tese. Trabalhos como os de Flandrin (1992) e Kaplan; Kuo (1993) validam este resultado e mostram que um processo com dependência de longa duração e com parâmetro de Hurst, $0.5 < H < 1$, depois de transformado por uma *wavelet* Haar (DAUBECHIES, 1992), apresenta coeficientes onde a soma das autocorrelações converge para uma constante. As investigações realizadas na estrutura da autocorrelação de coeficientes de *wavelet* indicam que a transformada de *wavelet* reduz drasticamente a dependência temporal. Assim, modelos de previsão, que não são adequados

para processos com dependência de longa duração, no domínio do tempo, podem ser utilizados nos coeficientes *wavelets*, uma vez que estes apresentam dependência de curto alcance.

Uma vez tendo-se os coeficientes gerados pela transformada de *wavelet*, estes podem ser utilizados para reconstruir a série original, isto é, o tráfego que está sendo analisado. Para isso é importante aplicar filtros para obter coeficientes menores do que um determinado limiar. Esta é, talvez, a mais importante aplicação das *wavelets* a uma técnica estatística. As principais contribuições nessa área foram dadas por Donoho; Johnstone (1995).

Após a reconstrução da série é possível identificar tendências crescentes que indicam congestionamento ou a falta de tendências no tráfego. Isto só é viável devido ao comportamento suave da série obtida pelos procedimentos de transformação direta, aplicação de filtros e transformação inversa. Revelada a tendência, o mecanismo efetua prognóstico do comportamento futuro do tráfego, através de testes de tendência de séries temporais.

Portanto, as principais contribuições deste trabalho estão: (i) na proposta desse mecanismo e na sua aplicabilidade prática e (ii) no estudo da transformada de *wavelet* com propósitos de realizar prognósticos de congestionamentos.

Além desta tese, os principais resultados (artigos) obtidos foram:

- Minicurso publicado em anais de conferência nacional (SBC) que trata dos modelos estocásticos de tráfego em redes de alta velocidade (CAMPOS; SILVA; CUNHA, 2001) – anexo 1.
- Artigo publicado em anais de conferência internacional que descreve a simulação de uma rede TCP e o comportamento dos pacotes descartados quando o ambiente fica congestionado (SILVA; CAMPOS; SOUZA; CUNHA, 2002b) – anexo 2.
- Artigo publicado em anais de conferência nacional (SBRC) que investiga o comportamento de pacotes descartados em redes TCP, através de simulação (SILVA; CAMPOS, SOUZA; CUNHA, 2002a) – anexo 3.
- Artigo submetido ao periódico IEEE - JSAC que trata do emprego do mecanismo proposto nesta tese em redes empresariais (SILVA; CAMPOS; CUNHA, 2004b) – anexo 4.

- Artigo em fase de elaboração a ser submetido ao periódico *Performance Evaluation*, intitulado “Wavelet-Domain Modeling of Congestion Control in Computer Networks”. Aborda o modelo FARIMA e a transformada de *wavelet* em prognósticos de congestionamento de redes.

Além desses artigos, outros resultados obtidos, desde o início do doutorado, foram:

- Trabalho publicado em conferência internacional: (SILVA; CUNHA, 2001a).
- Trabalhos publicados em conferências nacionais: (SILVA; CUNHA, 2000), (SILVA; CAMPOS; CUNHA, 2001b), (SILVA; CAMPOS; CUNHA, 2002d), (SILVA; CAMPOS; CUNHA, 2003)
- Trabalhos publicados em conferências regionais (SILVA; CAMPOS, 2001c), (SILVA; CAMPOS; CUNHA, 2002c).

1.5 Estrutura do Trabalho

Esta tese é constituída de 4 capítulos, além deste, das referências bibliográficas, de anexos e de apêndices.

As definições e os conceitos teóricos utilizados no presente trabalho estão no Capítulo 2, que se inicia com o problema do congestionamento de redes e as soluções encontradas, destacando as tecnologias para qualidade de serviço (QoS). Este capítulo também trata das questões de avaliação e otimização do desempenho das redes através da engenharia de tráfego e mostra em que consiste um tráfego de rede e quais os modelos de tráfego, ressaltando os processos auto-similares. Apresenta ainda a transformada de *wavelet* e suas formas de decomposição e de reconstrução de séries finitas, bem como os modelos de estimação de tendências de séries temporais.

Constam no terceiro capítulo a localização, a arquitetura e as especificações do mecanismo proposto. Inicialmente, são descritos a localização (onde o artefato vai executar suas funções), a arquitetura, as funções e os procedimentos que o mecanismo realiza. Em seguida, são relacionados os programas que compõem o mecanismo e suas funções. Na seção seguinte, descreve-se o tráfego simulado e especificam-se os parâmetros do mecanismo (bases de *wavelet*, políticas de redução de ruído, tipos de limiares de coeficientes e percentuais de redução) e dos indicadores de tendência de congestionamento. Ao final, são mostrados os cenários de execução dos programas do mecanismo.

Os resultados obtidos nesta tese estão todos concentrados no Capítulo 4. Neste capítulo apresentam-se os modelos de tráfego usados nos testes do mecanismo, as medidas avaliadas e os resultados obtidos. É feita também análise dos resultados mudando os parâmetros e dos tempos obtidos pelo mecanismo.

Finalmente, o Capítulo 5 dedica-se às conclusões, dando ênfase aos resultados que podem ser úteis para o desenvolvimento de mecanismos preventivos de controle de congestionamento de redes. Também são enumerados alguns dos trabalhos que estão em andamento e os possíveis trabalhos futuros.

Os anexos desta tese são resumos de trabalhos publicados e/ou submetidos para publicação e o código fonte do mecanismo proposto.

1.6 Comentários

Este capítulo expôs inicialmente o problema que se quer resolver, os objetivos que devem ser alcançados com a conclusão da tese e a relevância do problema para as redes atuais. A seção 1.3, apresentou os trabalhos relacionados com controle de congestionamento e engenharia de tráfego, modelagem de tráfego e a utilização *wavelets* na análise e na modelagem de tráfego de redes. Também nesta seção, expôs-se os principais grupos de pesquisa de redes que atuam na área de controle de congestionamento e engenharia de tráfego e as recomendações do IETF e do ITU-T para estas áreas. Por fim, foram apresentadas as contribuições esperadas e a estrutura da tese.

Capítulo 2

Fundamentos Conceituais

Este capítulo apresenta os conceitos teóricos utilizados no presente trabalho. Inicialmente, a Seção 2.1 trata do problema do congestionamento em redes e as soluções encontradas, destacando as tecnologias para qualidade de serviço (QoS). A seção seguinte, 2.2, é concernente com questões de avaliação e otimização do desempenho das redes, através da engenharia de tráfego. A Seção 2.3 mostra em que consiste um tráfego de rede, os modelos de tráfego, com ênfase em processos dependentes de curta duração e de longa duração, e os processos auto-similares. A Seção 2.4 trata da transformada de *wavelet* e suas formas de decomposição e reconstrução de séries finitas. Por fim, as Seções 2.5 e 2.6 apresentam, respectivamente, os modelos de estimação de tendências de séries e os comentários deste capítulo.

2.1 Congestionamento de Redes

Congestionamento em redes de computadores é basicamente um problema de compartilhamento de recursos (YANG; REDDY, 1995), que ocorre quando equipamentos ou nodos terminais inserem na rede uma quantidade maior de pacotes do que ela é capaz de tratar. Existem duas atividades distintas relacionadas ao controle ou gerenciamento de congestionamento (JACOBSON; KARELS, 1988; JAIN, 1992; LEFELHOCZ et al., 1996). A primeira é a prevenção de congestionamento que trata de detectar possíveis condições que levem a congestionamentos futuros e de executar procedimentos para impedi-los. A segunda é a recuperação de congestionamento que atua na rede após a ocorrência de congestionamento para que ela volte ao seu estado normal.

Particularmente na Internet, a ocorrência de congestionamento é devido à natureza imprevisível do tráfego. O serviço oferecido pelo protocolo IP na Internet, conhecido como serviço de “melhor esforço” (*best effort*), além de não dar garantias a respeito da entrega dos pacotes, não incorpora qualquer tipo de controle de admissão e reserva de recursos para limitar o tráfego inesperado. Congestionamento é, portanto, uma consequência natural dos princípios utilizados na arquitetura Internet. Por ser inevitável, os fluxos de tráfego podem se beneficiar quando participam ativamente do controle de congestionamento, pois não é

economicamente viável prover a rede de recursos para satisfazer às demandas de pico (LEFELHOCZ et al., 1996). No entanto, resultados confiáveis podem ser obtidos com o serviço de melhor esforço, se políticas de gerenciamento de filas e técnicas de controle de congestionamento apropriadas forem utilizadas (BRADEN et al., 1998).

2.1.1 Congestionamento na Internet

Na Internet o controle de congestionamento é realizado fim a fim, na camada de transporte, pelo protocolo TCP. A taxa de transmissão de dados do protocolo TCP é controlada pelas condições da rede e por isso os fluxos TCP são chamados de “compreensivos” ou “responsáveis”, pois respondem positivamente às notificações de congestionamentos (BRADEN, op. cit.). O UDP, por outro lado, não realiza qualquer tipo de controle de congestionamento. Em situações de congestionamento, os fluxos UDP não diminuem a taxa de transmissão de dados, a não ser que as aplicações o façam (BRADEN, op cit.), por isso são chamados de “agressivos” ou “não compreensivos”. Como cerca de 90% dos pacotes na Internet pertencem a fluxos TCP (THOMSON; MILLER; WILDER, 1997), em geral os congestionamentos são resolvidos de maneira adequada, apesar de várias implementações TCP existentes estarem incorretas (PAXSON et al., 1999).

Para resolver o problema de congestionamento, o protocolo TCP utiliza uma abordagem baseada na adaptação do emissor, que consiste na redução da taxa de transmissão. O modelo de adaptação do emissor tem trabalhado bem na Internet, porém alguns desafios conhecidos, como a dificuldade de se detectar congestionamento e o tempo requerido entre a percepção do congestionamento e a taxa de ajustamento, precisam ser tratados. Esses problemas resultam em períodos de perda de pacotes não controlados pela rede, conforme ilustrado na Figura 2.1. Contudo, poderiam ser resolvidos se as adaptações fossem movidas para os nós que estão no interior da rede.

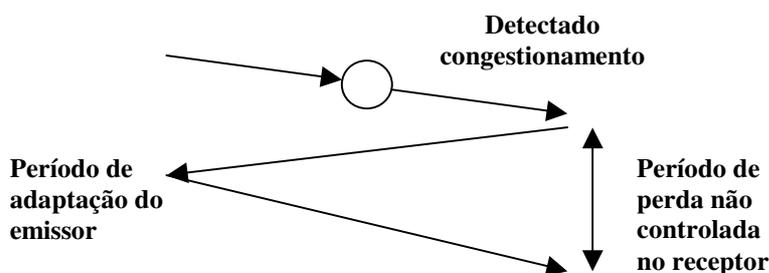


Figura 2.1. Período sem controle

Em 1986, a Internet experimentou uma série de problemas, conhecidos como “Colapso por Congestionamento”, que reduziu, significativamente, o desempenho das redes em situações de congestionamento (JACOBSON; KARELS, 1988). Tais problemas, apesar de raros, podem ocorrer quando o tráfego é intenso e a rede está congestionada, levando ao descarte de pacotes nos roteadores e, conseqüentemente, a ocorrência de retransmissões, aumentando assim o RTT (*Round Trip Time*). Este fato levou Jacobson; Karels (op.cit.) a desenvolverem e a introduzirem um conjunto de melhoramentos para o TCP, que estão consubstanciados nos algoritmos *Slow Start* e *Congestion Avoidance*. A partir daí, duas novas melhorias foram acrescentadas, os algoritmos *Fast Retransmit* e *Fast Recovery*. Esses quatro algoritmos estão padronizados na RFC 1122 e apresentados com mais detalhes na RFC 2581 (ALLMAN; PAXSON; STEVENS, 1999). O objetivo global dos mesmos é levar a rede a um estado estável de plena utilização, permitindo a introdução de um novo pacote à medida que um outro é retirado. Desse modo, os recursos da rede não são desperdiçados, ao mesmo tempo em que os congestionamentos são evitados. A Figura 2.2 ilustra o comportamento dos algoritmos *Slow Start* e *Congestion Avoidance*, mostrando o aumento exponencial do tamanho de uma janela de transmissão até atingir a capacidade da rede (estágio *Slow Start*). Quando atinge a capacidade da rede, ou seja, em situações de congestionamento (sinalizadas por perda de pacotes) a janela é reduzida. Do contrário, a janela aumenta linearmente. Esse estágio é chamado de *Congestion Avoidance*.

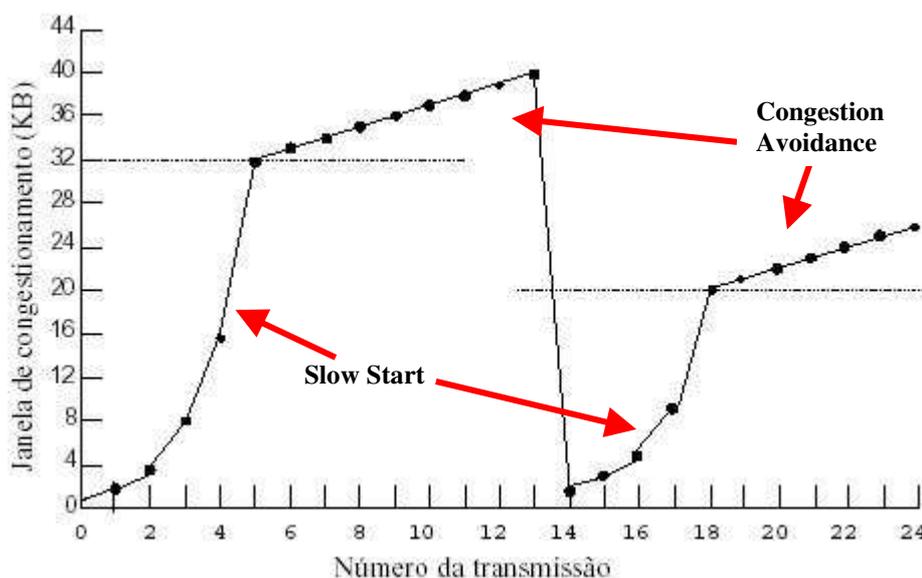


Figura 2.2. Comportamento do controle de congestionamento do TCP

Os mecanismos de controle de congestionamento do TCP ainda não são suficientes para prover serviços de boa qualidade em várias situações, como também não impedem que o congestionamento ocorra. Desta forma, novos mecanismos de controle foram desenvolvidos para serem implementados em nós intermediários (roteadores), monitorando a fila de saída destes nós, podendo assim detectar congestionamentos e identificar os fluxos de tráfego que mais contribuem para o problema. Dentre os mecanismos de prevenção destacam-se os algoritmos de gerenciamento de filas que controlam o tamanho da fila, através do descarte ou da marcação de pacotes.

Políticas de gerenciamento de filas, como a *Tail Drop*, e mecanismos de gerenciamento ativo de filas, como RED (*Random Early Detection*), ARED (RED adaptativo), FPQ (*Flow Proportional Queueing*), FRED (*Flow Random Early Drop*), Blue e SFB (*Stochastic Fair Blue*) são apresentados a seguir.

2.1.2 Políticas e Mecanismos de Gerenciamento de Filas

Tail Drop é a forma mais tradicional de gerenciamento do tamanho das filas nos roteadores (BRADEN et al., 1998). Esta política adota um valor máximo (quantidade de memória disponível) para o tamanho da fila, em termos da variável aleatória X número de pacotes. Os pacotes são aceitos pelo roteador enquanto existe espaço na fila, ou seja, até que o valor máximo de X seja atingido. Depois disto, os pacotes seguintes são descartados.

Esta técnica, por ser simples, foi utilizada por muitos anos na Internet. Porém, problemas tais como a injustiça dos descartes de pacotes, a monopolização e a ocupação total das filas, inviabilizam sua utilização. Se as filas estiverem sempre ou quase cheias, não sobra espaço suficiente para absorver rajadas de tráfego. Aumentar o tamanho da fila não é uma boa alternativa, pois aumentaria o atraso. Desta forma, o tamanho da fila deve espelhar o tamanho da maior rajada que se deseja absorver.

Com o crescimento da Internet e, conseqüentemente, do tráfego, os problemas oriundos da política *Tail Drop* tornaram-se mais críticos. Duas alternativas foram apresentadas para resolver o problema do monopólio: descarte do primeiro pacote da fila ou descarte aleatório de um pacote (BRADEN, op cit.). Contudo, estas políticas não resolvem o problema das filas cheias. Para solucionar este outro problema, é necessário manter o tamanho médio da fila pequeno que suporte tráfego em rajadas e, por conseguinte, minimize o número de pacotes descartados.

Na política *Tail Drop*, o mecanismo do TCP emissor só reduz sua taxa de transmissão quando é notificado através do descarte de pacotes, o que só ocorre quando a fila está cheia. O intervalo entre o momento do descarte feito no roteador e o momento em que o emissor descobre a perda do pacote é considerável. Durante esse período de tempo, que não tem qualquer controle pelo lado do roteador, o emissor continua transmitindo pacotes que, possivelmente, também serão descartados, dado o congestionamento existente. O ideal seria o emissor receber uma notificação sobre a dimensão da fila para diminuir sua taxa de transmissão antes que haja um transbordo.

Para manter o tamanho médio da fila pequeno, pacotes devem ser descartados aleatoriamente antes que o tamanho da fila atinja o seu máximo, notificando assim os nós emissores sobre a ocorrência de congestionamento. O descarte aleatório é realizado também para que os diversos fluxos TCP, que utilizam o mesmo roteador/canal congestionado, reduzam o seu tráfego melhorando a equidade (*fairness*). Os mecanismos descritos a seguir, denominados AQM (*Active Queue Management*), têm o objetivo de controlar quando e como o roteador deve descartar pacotes.

RED – *Random Early Detection*

É atualmente o algoritmo mais difundido e mais implementado (BRADEN et al., 1998). Foi proposto por Floyd; Jacobson (1993), com os seguintes objetivos:

- a) Manter um limite máximo do tamanho médio da fila para prevenir congestionamento. RED atinge este objetivo através da comparação do tamanho médio da fila com dois limiares;
- b) Prevenir monopolização das filas e a penalização do tráfego em rajadas. Descartes de pacotes são proporcionais à utilização da fila por um ou mais fluxos.

O valor do tamanho médio da fila é comparado com dois limiares (*thresholds*), *min* e *max*, que são utilizados para definir três zonas. Se o tamanho médio da fila estiver abaixo do *min*, o algoritmo está na zona normal de operação, ou seja, todos os pacotes que chegam são aceitos. Caso o tamanho médio esteja entre os dois limiares (*thresholds*), o algoritmo está na zona de prevenção e cada pacote que chega pode ser descartado com uma probabilidade p . Se o tamanho médio estiver acima de *max*, o algoritmo está na zona de controle e todos os pacotes que chegam são descartados.

Apesar de ter muitas vantagens em relação à política *Tail Drop*, o algoritmo RED ainda apresenta várias desvantagens:

- a) Injustiça é a maior desvantagem. As deficiências que originam este problema são:
 - Penaliza conexões com janelas pequenas. Conexão com janela pequena tem que esperar por um tempo relativamente grande (mais de um RTT) para se recuperar de uma perda;
 - Não sensível a tráfego não-adaptativo. Enquanto o tráfego adaptativo reduz sua taxa de transmissão, o não-adaptativo (UDP) não reduz, conseguindo assim maior banda passante.
- b) Dificuldade na determinação correta dos limiares, *min* e *max*, pois depende do tamanho médio da fila que se deseja manter e este, por sua vez, depende do tipo de tráfego. Como a priori não se sabe as características do tráfego que se vai trabalhar, é impossível determinar valores ótimos dos limiares. Este algoritmo só atinge um ponto operacional ideal quando possui *buffers* suficientes e quando seus limiares estão corretamente definidos;
- c) Degradação do comportamento do algoritmo quando trabalha com grande quantidade de fluxos ativos. Muitos fluxos ativos provocam um tráfego agregado em rajadas. Isso pode fazer com que o tamanho médio da fila oscile rapidamente antes que o algoritmo RED possa reagir, provocando muitas perdas de pacotes e, conseqüentemente, baixa utilização do enlace (FENG, 1999).

ARED – Adaptive RED

Foi proposto inicialmente por Feng (1999) e modificado por Floyd; Gummadi; Shenker (2001). A idéia deste algoritmo é inferir os dois limiares, *min* e *max*, através da verificação do tamanho médio da fila. Quando o número de fluxos ativos é pequeno, o tamanho médio da fila deve oscilar ao redor de *min*, levando a diminuição do valor deste limiar. Esta medida evita a baixa utilização do enlace. Por outro lado, se o número de fluxos ativos é muito grande, o tamanho médio da fila deve oscilar ao redor de *max*, o que significa que o valor de *max* deve ser aumentado. Desta forma, evita-se o descarte de todos os pacotes, quando o tamanho médio da fila atinge o valor de *max*.

FRED – *Flow Random Early Drop*

É um algoritmo que resolve o problema da injustiça do RED decorrente de uma grande quantidade de fluxos ativos. Enquanto RED faz a indicação de congestionamento de fluxos escolhidos de forma aleatória, o algoritmo FRED o faz de forma seletiva e atinge somente os fluxos que usam as filas mais intensamente (LIN; MORRIS, 1997).

FRED mantém o estado dos fluxos ativos, ou seja, dos fluxos que possuem pacotes enfileirados no roteador. É um algoritmo que protege conexões com janelas pequenas e consegue lidar com fluxos não-adaptativos. O preço destas vantagens é uma maior complexidade e um custo computacional considerável.

Existem algoritmos de AQM que armazenam informações da fila de saída por fluxo. Isso permite um controle justo do tamanho médio da fila, uma vez que esses algoritmos têm controle sobre a participação de cada fluxo na ocupação da fila. Com esse controle é possível penalizar fluxos que ocupam mais a fila, protegendo conexões com janelas pequenas.

2.1.3 Controle de Congestionamento de Fluxos Amigáveis ao TCP (*TCP-friendly*)

Não são todas as aplicações na Internet que usam o protocolo TCP e seguem o conceito da equidade para compartilhar a largura de banda disponível. O efeito indesejável das aplicações não-TCP não tem tido, por enquanto, um forte impacto, já que a maioria do tráfego Internet utiliza protocolos baseados no TCP, tais como HTTP, FTP etc. Contudo, o número de aplicações de áudio/vídeo, telefonia IP, vídeo-conferência e outras de tempo real está constantemente crescendo e uma consequência será o aumento na percentagem do tráfego não-TCP. Essas aplicações não usam os mecanismos de controle de congestionamento do TCP, competem com fluxos TCP e monopolizam a banda passante em caso de engarrafamento. Por esta razão, é desejável a definição de regras que compatibilizem as taxas de transmissão do tráfego não-TCP, isto é, aplicações que apesar de usarem o protocolo UDP respondem ao congestionamento, reduzindo o tráfego de pacotes de forma semelhante ao TCP.

Fluxo *TCP-friendly* é definido por Floyd; Kevin (1999) como aquele cuja capacidade de vazão (*throughput*), a longo prazo, não excede a capacidade de uma conexão TCP sob as mesmas condições. Aplicações *TCP-friendly* podem usar os mesmos esquemas de controle de congestionamento do TCP. Esses esquemas podem ser baseados em janela de congestionamento ou em taxa de transmissão (WIDRNER; DENDA; MAUVE, 2001). A

abordagem baseada em janela controla o fluxo de transmissão através de “créditos” provenientes da rede ou do receptor. No esquema baseado em taxa, o fluxo de transmissão é inferido constantemente, através de medições realizadas no emissor.

2.1.4 Tecnologias em QoS

O surgimento de novas aplicações multimídia fez com que as redes necessitassem de mais largura de banda. Enquanto as aplicações tradicionais (telnet, FTP, WWW) são menos sensíveis aos retardos e intolerantes às perdas de pacotes, as novas aplicações multimídia compensam as perdas de pacotes, porém são rigorosas com relação aos altos retardos. Isso significa que aplicações que utilizam largura de banda pequena ou instável podem ser interrompidas e provocar congestionamentos na rede devido às retransmissões. Por isso, para essas aplicações são necessários novos conceitos que garantam a qualidade de serviço (QoS) desejada.

Para atender à demanda de QoS das atuais aplicações multimídia, novas tecnologias foram desenvolvidas, destacando-se: (i) as arquiteturas *DiffServ* (Diferenciação de Serviços) e *IntServ* (Integração de Serviços), (ii) MPLS (*Multi-Protocol Label Switching*) e (iii) QoS (Roteamento baseado em QoS). Essas tecnologias atendem os requisitos de QoS de formas distintas. As duas arquiteturas estão relacionadas com a alocação de recursos e com o tipo de serviço que as aplicações necessitam e que podem obter, enquanto que MPLS e QoS estão relacionadas com a infra-estrutura necessária para garantir que as arquiteturas *IntServ* e *DiffServ* possam satisfazer os requisitos das aplicações, otimizando o uso dos recursos da rede (WANG, 2001). Além destas tecnologias existe algo mais amplo, conhecido como engenharia de tráfego, que faz uso do *DiffServ*, *IntServ* e/ou MPLS para atender às demandas de QoS (AWDUCHE, et al., 2000).

Embora haja sobreposição de interesses entre as abordagens apresentadas e apesar de servirem para solucionar problemas distintos, é possível combiná-las em uma única arquitetura integrada para prover QoS. A distinção entre as funções de cada uma das tecnologias apresentadas facilita a compreensão de como elas podem interagir.

As arquiteturas *IntServ* e *DiffServ* representam o rompimento com o modelo tradicional do melhor esforço para oferecer serviços garantidos. Representam visões distintas dos mecanismos que devem ser utilizados. A *IntServ* é baseada em reserva de recursos e pode oferecer garantias rígidas a determinados fluxos de dados. Nesta arquitetura, as aplicações definem o caminho pelo qual seus pacotes devem passar e fazem reserva dos recursos

necessários antes de transmitir, de fato, seus dados. O empecilho para sua utilização em larga escala é a sua falta de escalabilidade. Na arquitetura *DiffServ* não há reserva de recursos, mas oferecimento de garantias para agregações de fluxos. Os pacotes são marcados de acordo com classes de serviços pré-definidos e os pertencentes às mesmas classes recebem o mesmo tratamento. É considerada escalável e por isso é atualmente a abordagem que, provavelmente, será usada para QoS. No entanto *IntServ* pode ser utilizada com sucesso em segmentos de rede onde a escalabilidade não representa um grande problema.

MPLS é uma técnica de encaminhamento de pacotes, não necessariamente direcionada para o provimento de QoS. Os pacotes são rotulados ao entrarem em um domínio com suporte a MPLS e a classificação e o posterior encaminhamento dos mesmos por uma determinada rota são definidos baseados nestes rótulos. Pode ser utilizada para construir caminhos escolhidos explicitamente, que têm recursos suficientes para sustentar as necessidades de desempenho de certos fluxos de dados. Os pacotes que trafegam por esses caminhos podem receber tratamento diferenciado dentro dos roteadores. MPLS representa um rompimento (ou uma evolução) com o mecanismo tradicional de encaminhamento de pacotes, chamado de salto a salto (*hop-by-hop*).

QoSR é uma técnica de roteamento, que encontra caminhos que atendem às necessidades de QoS de determinados fluxos. Esta técnica apenas indica qual a rota mais adequada para que os níveis de QoS possam ser mantidos. Deve ser utilizada em conjunto com outro mecanismo de provimento de QoS, como *IntServ* ou *DiffServ*. Não é um mecanismo de encaminhamento, ou seja, pode descobrir novas rotas, mas não tem como forçar pacotes de certos fluxos a seguirem obrigatoriamente essas rotas. Portanto, sua utilização faz mais sentido quando se modifica o mecanismo de encaminhamento básico, por exemplo, com MPLS.

2.2 Engenharia de Tráfego

Engenharia de tráfego não é uma técnica e sim um processo de organizar convenientemente o fluxo de tráfego, para que congestionamentos causados pela utilização desigual da rede possam ser evitados. Trata das questões de avaliação e otimização de desempenho de redes operacionais. Em geral, engloba a aplicação de princípios tecnológicos e científicos para medir, modelar, caracterizar e controlar o tráfego, e de técnicas e conhecimentos para atingir determinados objetivos de desempenho (AWDUCHE, 1999; AWDUCHE et al., 1999). Um

fenômeno indesejável que prejudica o desempenho de redes operacionais e constitui uma classe de problemas, particularmente importante para engenharia de tráfego, é o do congestionamento.

Atualmente, a engenharia de tráfego é uma função indispensável em redes de longa distância e empresariais por causa do alto custo dos equipamentos e da natureza comercial e competitiva entre as empresas. Pode ser implementada através de diversos mecanismos, como configuração manual, utilização de características específicas das tecnologias de transmissão da camada de enlace de dados, determinação de rotas adequadas através de QoS e fixação de rotas com MPLS. Pode ser implementada tanto em redes convencionais, que oferecem o serviço de melhor esforço, quanto em redes que oferecem níveis de QoS, com *IntServ* ou *DiffServ*. Na realidade, não se deve utilizar ao mesmo tempo todas as arquiteturas e tecnologias citadas como forma de organização do fluxo de tráfego na rede, pois isto causaria diminuição da relação custo/benefício. Alguns exemplos de implementações conjuntas de arquiteturas, tecnologias e engenharia de tráfego são: (i) engenharia de tráfego com MPLS (AWDUCHE et al., 1999) e RSVP (protocolo de reserva de recursos) (LI; REKHTER, 1998), para *IntServ* e *DiffServ* (AWDUCHE et al., 2000); (ii) engenharia de tráfego com QoS (CRAWLEY et al., 1998); (iii) interoperabilidade entre redes *IntServ* e *DiffServ* (BERNET et al., 2000) e (iv) utilização de *DiffServ* em uma rede MPLS (FAUCHER et al., 2002).

2.2.1 Problemas com os Protocolos de Roteamento

Na internet, a engenharia de tráfego é extremamente importante, principalmente porque os protocolos de roteamento interno, como o IGP (*Interior Gateway Protocol*), usam sempre o menor caminho para encaminhar o tráfego e isso contribui significativamente para aumentar os problemas de congestionamento. Fatores, como a largura de banda disponível e as características do tráfego, não são levados em consideração nas decisões de roteamento. Conseqüentemente, sempre irá ocorrer congestionamento quando:

- (i) O caminho mais curto de várias fontes de dados convergir para determinados enlaces.
- (ii) O tráfego de uma determinada fonte de dados é roteado, através de um enlace que não tem suficiente largura de banda para encaminhá-lo adequadamente.

Esse tipo de cenário se manifesta mesmo quando existem caminhos alternativos com capacidade ociosa suficiente para encaminhar o tráfego excedente. Isto é, a utilização dos protocolos de roteamento, baseados no menor caminho, tende a degradar o desempenho

observado pelos fluxos de dados, mesmo quando existem recursos suficientes para tratar todo o tráfego.

A Figura 2.3 ilustra o tratamento que a engenharia de tráfego dá a certos tipos de tráfego, encaminhando pacotes por caminhos diferentes, dependendo das políticas adotadas. Em um ambiente sem engenharia de tráfego todos os pacotes são encaminhados pelo caminho mais curto, que na figura corresponde ao caminho do meio.

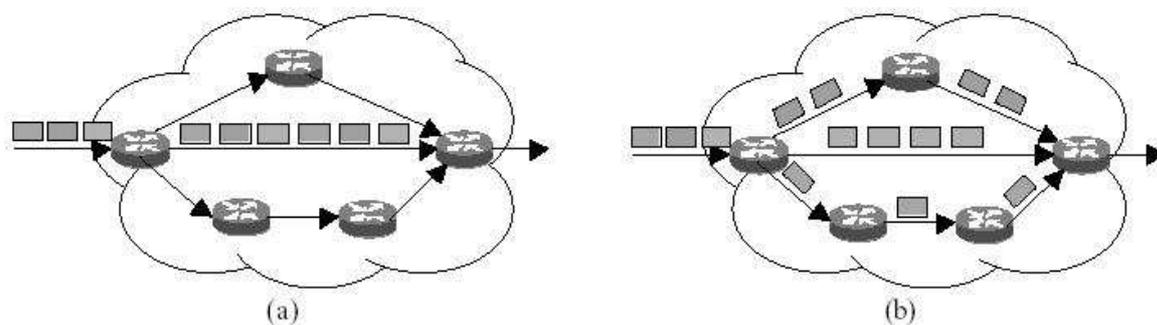


Figura 2.3. Encaminhamento de pacotes: (a) sem engenharia de tráfego; (b) com engenharia de tráfego

2.2.2 Objetivos de Desempenho da Engenharia de Tráfego

Otimizar o desempenho de redes operacionais envolve objetivos orientados ao tráfego e à recursos (AWDUCHE et al., 1999). Os objetivos de desempenho orientados ao tráfego incluem os aspectos relacionados à melhoria dos níveis de QoS oferecidos aos fluxos de dados. Na Internet atual, esses objetivos incluem: minimização da perda de pacotes, minimização do atraso e maximização da vazão. Em uma rede de serviços diferenciados, com várias classes de serviços que necessitam limite máximo e mínimo para métricas de QoS, os objetivos orientados ao tráfego assumem uma importância ainda maior. Principalmente, porque, em última análise, o que mais importa é como os usuários da rede sentem o desempenho dos serviços utilizados. Em uma rede de melhor esforço, a noção de desempenho assume critérios subjetivos para o usuário. Em uma rede que oferece níveis de QoS, o usuário tem como mensurar e exigir o desempenho contratado.

Os objetivos orientados a recursos incluem aspectos relacionados à otimização da utilização dos recursos da rede. O gerenciamento eficiente dos recursos é a melhor maneira de alcançá-los. Mais especificamente, é importante garantir que os recursos de alguns segmentos da rede não fiquem superutilizados e congestionados, enquanto que outros segmentos com

caminhos alternativos para o tráfego tenham recursos ociosos. Uma das principais funções da engenharia de tráfego é gerenciar o recurso mais importante das redes atuais, a largura de banda. Existe uma discussão sobre até que ponto a engenharia de tráfego deve interferir no encaminhamento de pacotes na rede, uma vez que as modernas tecnologias estão oferecendo cada vez mais largura de banda a um custo menor. No entanto, até o presente momento, a solução com o melhor custo/desempenho para grandes redes é realizar alguma forma de engenharia de tráfego ao invés de simplesmente aumentar a largura de banda.

2.3 Modelagem de Tráfego de Redes

Um tráfego de rede consiste de chegadas de entidades (bytes, pacotes, células, etc.) que formam um fluxo caracterizado por uma seqüência de observações $\dots, X(t_{n-1}), X(t_n), X(t_{n+1}), \dots$ nos instantes de tempo $\dots, t_{n-1}, t_n, t_{n+1}, \dots$, as quais descrevem, por exemplo, o número de pacotes ou de bytes que chegam a um servidor da rede durante um período especificado.

A Figura 2.4 mostra a chegada de pacotes em um nó com capacidade finita de *buffer* (fila de entrada) e procedimentos de modelagem de tráfego baseados em medições e em séries temporais. O processo de chegada na fila de entrada do nó da rede pode sofrer, por exemplo, a ação de um filtro (de passa alta) que corta as freqüências baixas, isto é, valores pequenos da série (estabelecidos de acordo com algum critério). Neste caso, somente os valores grandes poderão servir para fins de modelagem.

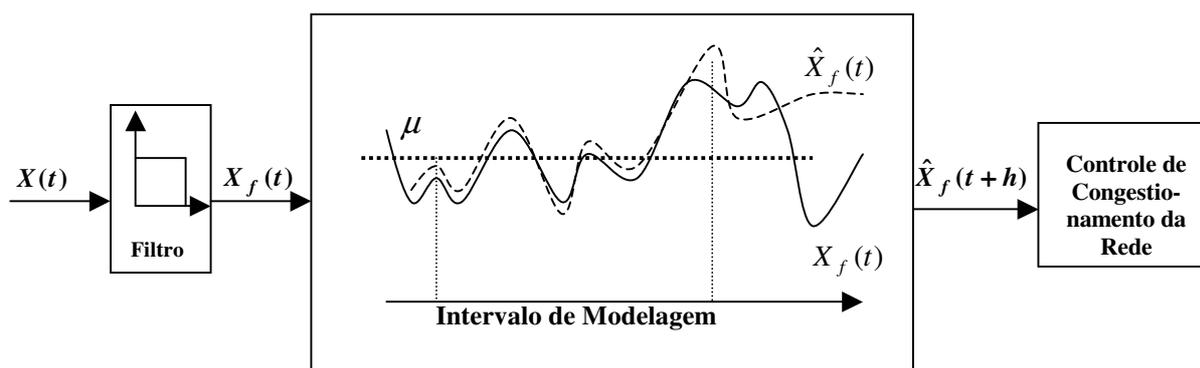


Figura 2.4. Modelagem de tráfego: $X(t)$, tráfego de chegada; $X_f(t)$, tráfego filtrado, μ - taxa média do tráfego filtrado e $\hat{X}_f(t+h)$, tráfego previsto no tempo $t+h$.

A modelagem de tráfego de redes pode ter como objetivo a representação do tráfego real por um processo estocástico. O processo selecionado é útil quando: (i) pode gerar *traces* que espelham a realidade; (ii) é parametrizado com o menor número possível de variáveis e (iii) pode ser manipulado matematicamente para prover indicadores de desempenho da rede tais como perdas de pacotes, atrasos (*delay*) e variação de atrasos (*jitter*).

Os primeiros modelos matemáticos de tráfego foram usados nos sistemas de telefonia tradicional. Nesses sistemas, as probabilidades de bloqueio e enfileiramento da chamada telefônica, dois importantes índices de desempenho, são fornecidas pelas fórmulas de Erlang B e C, respectivamente (GALLAGER; BERTSEKAS, 1992). Essas fórmulas tiveram tanto sucesso que foram tabuladas e utilizadas pelas Companhias de Telecomunicações do mundo todo durante muitos anos. O sucesso das fórmulas está baseado no fato de que os modelos eram uma boa representação da realidade, ou seja, as chamadas iniciadas seguiam um processo de Poisson e suas durações uma distribuição exponencial. Essas distribuições estavam descrevendo exatamente o comportamento humano, o qual era muito previsível.

Redes comutadas por pacotes começaram a aparecer nos anos 60 como uma maneira de transmitir dados. Como a maioria dos dados é menos sensível ao atraso, suas transmissões são mais eficientes, em termos de recursos, se eles forem divididos em pacotes que são processados independentemente pela rede. Desta forma, um circuito reservado para somente uma conexão de telefone, por exemplo, pode ser usado por muitas conexões de dados multiplexadas. Com as redes comutadas de pacotes, as conversas são feitas principalmente pelos computadores ao invés dos humanos, cujos comportamentos são mais previsíveis. O envolvimento de computadores complica a situação, pois as dinâmicas do tráfego são mais influenciadas, não somente pelo comportamento humano, mas também pelos diferentes protocolos de comunicação e suas implementações e pelos equipamentos e suas configurações. A explosão da Internet causada pela popularidade das aplicações Web tem complicado ainda mais a caracterização do tráfego.

Na próxima seção são descritos os modelos de tráfego mais comumente usados no passado e o modelo auto-similar utilizado na modelagem do tráfego atual.

2.3.1 Modelagem Tradicional

Processo de Poisson

O processo de Poisson (ROSS, 1985) foi um dos primeiros a ser usado na modelagem de tráfego. Descreve o número de chegadas de entidades no intervalo de tempo $(0, t]$ através da seguinte expressão:

$$P(X(t) = x) = \frac{(\lambda t)^x e^{-\lambda t}}{x!}, \quad \lambda > 0, \quad x = 0, 1, \dots \quad (1)$$

com taxa média de chegada (número médio de chegadas por unidade de tempo) igual a λ . O número de chegadas em dois intervalos de tempo disjuntos são independentes. Algumas propriedades importantes do processo de Poisson são:

- (i) O n -ésimo tempo, X_n , entre duas chegadas sucessivas é descrito por uma distribuição exponencial com parâmetro $\lambda > 0$:

$$P(X_n \leq t) = 1 - e^{-\lambda t}, \quad t \geq 0. \quad (2)$$

- (ii) A soma de processos de Poisson fornece um processo de Poisson com taxa λ igual à soma das taxas dos processos individuais.

Se um sistema segue uma distribuição exponencial então não tem memória, isto é, $P(X > s + t | X > s) = P(X > t)$ para s, t reais positivo. Em outras palavras, o tempo para a próxima chegada é independente de quando a chegada anterior ocorreu. Esta propriedade foi usada amplamente em teoria das filas (KLEINROCK, 1975), particularmente nas filas M/M/1.

O modelo de Poisson pode ser usado para representar a chegada de pacotes em um nó da rede, desde que o tráfego observado não apresente qualquer autocorrelação. Outras possíveis aplicações são: (i) a chegada de usuários para utilizar alguma facilidade (aplicação) do computador e (ii) o fluxo de comandos editados por um usuário no computador, desde que não se observe dependência proveniente do passado.

Modelo ON/OFF com taxa de saída constante

Um dos modelos de tráfego mais simples, aplicado em tráfego de voz, é o processo ON/OFF representado na Figura 2.5. O estado OFF representa o estado do recurso quando não existe atividade de voz, enquanto o estado ON representa a fonte quando existe um surto de

conversa. A distribuição do tempo gasto em cada estado é exponencial com média $1/\lambda$ para o estado OFF e $1/\mu$ para o estado ON. A taxa para a transição OFF \rightarrow ON é λ , enquanto para a transição reversa é μ , como indicado na figura.

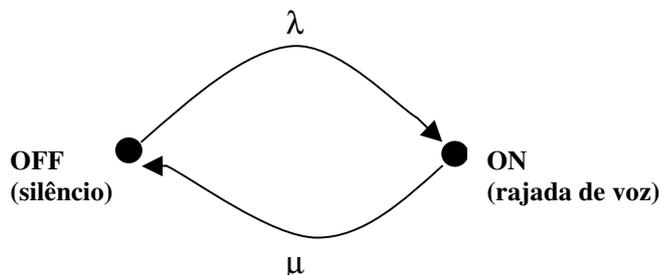


Figura 2.5. Modelo de tráfego ON/OFF para uma única fonte de voz

Este modelo foi sucesso na modelagem em voz principalmente por duas razões: i) experimentos atuais com uma única fonte de voz têm mostrado que a distribuição do silêncio e os períodos ativos podem ser representados por distribuições exponenciais e ii) o modelo de fonte única pode ser expandido para um processo de nascimento e morte multi-estado para representar voz multiplexada.

O processo ON/OFF também pode ser usado para simular processos com comportamento explosivo, como o tráfego de redes, onde os dados são transferidos dentro de uma sucessão de períodos ON separados por períodos de inatividade OFF. O modelo IPP (*Interrupted Poisson Process*) (ANDERSEN; NIELSEN, 1998) é um caso especial desse processo, onde os períodos ON/OFF e a taxa entre chegadas durante o período ON são mutuamente independentes.

Modelo de Poisson modulado por uma Cadeia de Markov (MMPP)

O primeiro passo para descrever dependências entre os $X(t)$ é dado pelos processos de Markov (ROSS, 1985). Em um modelo de tráfego que utiliza Markov, cada transição de estado é interpretada como sinalização de uma chegada, de modo que os intervalos entre chegadas são exponencialmente distribuídos e os parâmetros de taxa de chegada são dependentes dos estados onde a cadeia estava antes da mudança. Isto resulta na dependência entre intervalos de chegadas. Em outras palavras, as chegadas ocorrem segundo uma distribuição de Poisson com uma taxa que varia de acordo com um estado k da cadeia de Markov.

O processo de Poisson modulado por Markov é semelhante ao processo de nascimento e morte descrito no modelo ON/OFF. A principal diferença é que, em um estado ativo, o modelo MMPP gera Poisson ao invés de tráfego constante e à medida que o estado, na qual a cadeia se encontra, se altera a taxa do processo de Poisson também se modifica. A principal desvantagem deste modelo é a quantidade de parâmetros que precisa ser estimada, isto é, o número de estados da cadeia. Este modelo pode ser utilizado para modelar situações reais onde as observações dependem somente do valor anteriormente observado e onde a taxa de chegada varia com o passar do tempo, como por exemplo: (i) o comportamento do usuário em um terminal, onde a próxima ação é determinada pela ação anterior; (ii) a falha / mudança do estado da rede ou do sistema e (iii) o tráfego de rede que apresenta autocorrelação zero ou próxima de zero. Heffes; Lucantoni (1986) e Nagarajan; Kurose; Towsley (1991) utilizaram este modelo para modelar fontes de tráfego de voz.

Modelo Auto-Regressivo (AR)

A especificação de um modelo auto-regressivo (MORETTIN, 1985) é dada pela equação

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + e_t, \quad (3)$$

onde x_t corresponde à observação da série temporal no tempo t , ϕ_p é o parâmetro de ordem p e e_t representa o erro de eventos aleatórios que não podem ser explicados pelo modelo, isto é, representa o ruído branco Gaussiano com média zero e variância σ^2 .

Os processos auto-regressivos definem a próxima observação de uma seqüência como uma função explícita das observações calculadas anteriormente, dentro de uma janela de tempo determinada, fazendo com que o processo seja indicado para modelagem de tráfego autocorrelacionado. Tal modelagem foi utilizada por Maglaris et al. (1988) para modelar a taxa de bits de uma fonte de vídeo comprimido. A natureza dos quadros de vídeo é tal que quadros sucessivos dentro de uma mesma cena têm uma pequena variabilidade (vídeo de alta qualidade tem aproximadamente 30 quadros por segundo). As vantagens deste modelo são sua simplicidade e a necessidade de se estimar parâmetros somente através do ajustamento da média e da correlação da taxa real de bits. A desvantagem está em não poder representar, no caso da fonte de vídeo, mudanças de cena, pois estas podem provocar alterações bruscas na taxa de bits de quadros.

2.3.2 Auto-Similaridade e Novas Abordagens em Modelagem de Tráfego

Mandelbrot (1982) lançou as primeiras noções sobre auto-similaridade e fractais. Porém, somente em 1994 surge, como fruto de uma pesquisa, o artigo de Leland et al. (1994) e a partir daí o mundo da engenharia de tráfego e dos modelos de caracterização de tráfego de dados nunca mais seria o mesmo (STALLINGS, 1997; WILLINGER; PAXSON, 1998a). Os antigos paradigmas de considerar o comportamento do tráfego de redes como obedecendo às leis Poissonianas ou, de forma mais geral, obedecendo regras Markovianas, foram colocados num cenário altamente questionável.

Nos estudos analíticos de amostras empíricas de dados realizados por Leland et al. (op. cit.) mostrou-se, de maneira irrefutável, que o comportamento do tráfego das redes Ethernet possui a característica de preservar determinadas propriedades estatísticas em escalas de tempo distintas. Mostrou-se também que o comportamento do tráfego está longe de apresentar uma regularidade estatística e de convergir no limite para um processo Gaussiano ordinário (ruído branco Gaussiano) (TAQQU; WILLINGER; SHERMAN, 1997; WILLINGER et al., 1997). Segundo esses estudos, as características estatísticas atuais do tráfego de LAN/WAN, diferem de maneira surpreendente dos modelos tradicionais citados por Heffes; Lucantoni (1986). O tráfego de dados coletado por Leland et al. (op. cit.), representado na Figura 2.6, apresentou rajadas em várias escalas de observação, isto é, as características estatísticas são degeneradas lentamente de forma quase independente da escala de observação. Esta figura mostra a estrutura do tráfego real em diferentes escalas de tempo (de 0,01s a 100s) e compara com o tráfego gerado por um processo de Markov. Observa-se na figura que a estrutura do tráfego real permanece semelhante (auto-similar) para escalas de tempo diferentes, indicando a persistência da explosividade do tráfego, enquanto que a estrutura do tráfego Markoviano modifica-se, tornando-se muito suave para grandes escalas de tempo.

Estudos posteriores têm constantemente detectado a natureza auto-similar em outros tipos de tráfego (GUO; CROVELLA; MATTA, 2000). Portanto, existe hoje o consenso de que os modelos de tráfego devem capturar esta característica fundamental para que se possa gerar indicadores de desempenho de redes ou fazer previsões confiáveis de seu comportamento.

A principal implicação que isto traz é no desenvolvimento de algoritmos para evitar o congestionamento de tráfego de dados. Caso eles sejam baseados nos modelos tradicionais, praticamente a vazão ou perda de dados será considerável (WILLINGER, PAXSON, 1998),

conseqüentemente, isto gerará uma degradação na qualidade de serviço (QoS) fornecida pela rede e requisitada pelas fontes finais de transmissão e recepção de dados.

(Tráfego real)

(Tráfego gerado)

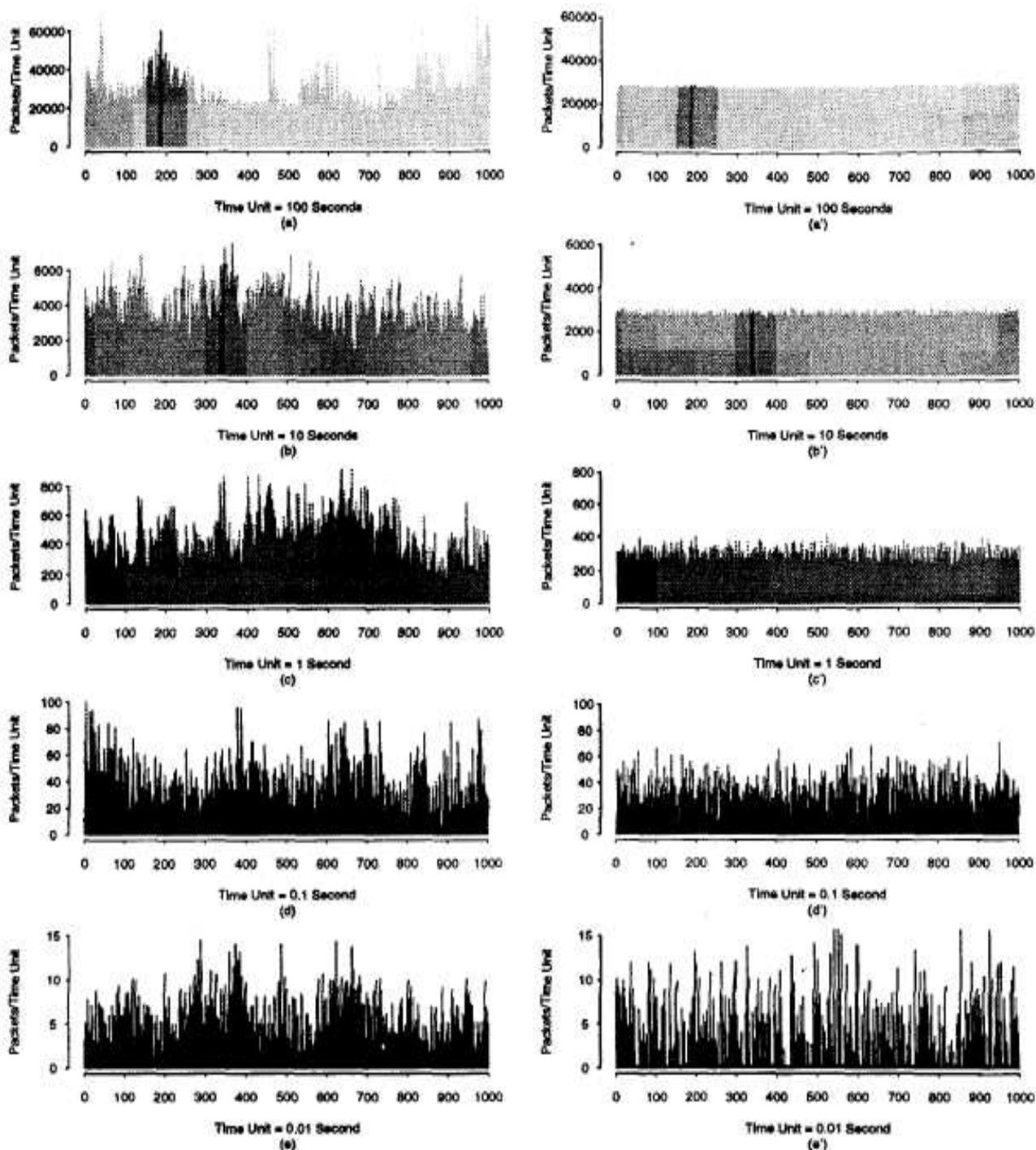


Figura 2.6. Auto-similaridade em tráfego real (Leland et al. 1994)

2.3.2.1 Processos Estocásticos Estacionários de Segunda Ordem

Para fins de modelagem de tráfego, a escolha de processos estocásticos estacionários de segunda ordem e de tempo discreto é apropriada pelo fato de não limitar o estudo dos modelos de tráfego de redes, uma vez que as chegadas de pacotes no tráfego das redes de alta velocidade, durante um intervalo de tempo, são discretas e usualmente considerados como tendo variância finita com algum tipo de estacionaridade (PARK; WILLINGER, 2000).

Assim, os processos estocásticos de tempo discreto estacionário, neste contexto as séries temporais $X(t)$, são interpretados aqui como o volume de tráfego medido em pacotes, bytes ou bits, no instante de tempo t . Estacionaridade é uma propriedade de determinados processos estocásticos e tem o significado de “invariância com o tempo”. Os conceitos de estacionaridade e auto-similaridade são explicados a seguir.

Definição 1. *Seja $\{X(t), t \in T\}$ um processo estocástico. $\{X(t), t \in T\}$ é estritamente estacionário se e somente 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 $\forall k, n \in \mathbb{Z}$.*

Em palavras, a distribuição conjunta de ordem n é invariante sob mudanças da origem do tempo. Para esse processo, a média, $E(X(t))$, e a variância, $V(X(t))$, independem do tempo e a covariância, $\gamma(t, t + k)$, depende apenas da diferença dos tempos.

Definição 2. *Um processo estocástico é fracamente estacionário (ou estacionário de segunda ordem) se e só se $\forall t \in T$, $E(X(t)) = \mu$, $V(X(t)) = \sigma^2 < \infty$ e $\gamma(t, t + k) = \gamma(k)$.*

Seja $X = \{X_t\}, t = 0, 1, 2, \dots$ um processo estocástico estacionário com variância $V(X)$ e função de autocorrelação $\rho(k), k \geq 0$. Sejam $\{X_k^{(m)}\}$ novas séries obtidas pela divisão proporcional da série original em blocos não sobrepostos de tamanho m , ($m = 1, 2, \dots$),

$$X_k^{(m)} = \frac{1}{m}(X_{km-m+1} + \dots + X_{km}), \quad k = 1, 2, \dots \quad (4)$$

onde as séries $X_k^{(m)}$ têm variância $V(X_k^{(m)})$ e função de autocorrelação $\rho^{(m)}(k)$. O processo agregado, $X_k^{(m)}$, é obtido dividindo a série de tempo original em m blocos não sobrepostos. O índice k etiqueta o bloco e cada bloco tem um comprimento m . Assim a equação (4) pode ser reescrita como

$$X_k^{(m)} = \frac{1}{m} \left[\sum_{i=(k-1)m+1}^{km} X_i \right], \quad k \geq 1. \quad (5)$$

A Eq. (5) representa a nova série de tempo (suavizada) e cada elemento dessa série corresponde ao valor médio empírico calculado para cada bloco m . Portanto, o processo $X_k^{(m)}$ é equivalente ao processo original X em uma escala de tempo que foi m vezes reduzida. É possível mostrar que para $m > 1$ o processo $X_k^{(m)}$ é estritamente estacionário (BERAN et al., 1995; PAXSON, 1997).

Definição 3: Um processo estritamente estacionário é chamado de auto-similar com parâmetro de auto-similaridade $0 < H < 1$ se

$$m^{1-H} X^{(m)} \stackrel{d}{=} X, \quad m = 1, 2, \dots \quad (6)$$

isto é, a distribuição de probabilidade é a mesma, tanto para a soma agregada ($X^{(m)}$) quanto para a série original (X). Se X é auto-similar, então X e $X^{(m)}$ têm a mesma função de autocorrelação

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2},$$

onde $\gamma(k)$ é a autocovariância da série e $\gamma(0)$ a sua variância.

Definição 4: Um processo $X(t)$ é chamado exatamente auto-similar de segunda ordem com parâmetro de auto-similaridade H ($0,5 < H < 1$) se

$$\rho(k) = \frac{1}{2} \left((k+1)^{2H} - 2k^{2H} + (k-1)^{2H} \right), \quad k \geq 1. \quad (7)$$

Definição 5: $X(t)$ é chamado assintoticamente auto-similar de segunda ordem se

$$\lim_{m \rightarrow \infty} \rho^{(m)}(k) = \frac{1}{2} \left((k+1)^{2H} - 2k^{2H} + (k-1)^{2H} \right), \quad k \geq 1. \quad (8)$$

Processo estritamente estacionário auto-similar implica em auto-similaridade de 2ª ordem exata, enquanto o oposto não é verdadeiro, exceto se o processo for Gaussiano (PARK; WILLINGER, 2000). Se o processo $X(t)$ for Gaussiano e exatamente auto-similar de 2ª ordem com média zero, então ele é estritamente auto-similar. A partir da Definição 4, um processo assintoticamente auto-similar de 2ª ordem aproxima-se de um processo exatamente

auto-similar de 2ª ordem quando o tamanho do bloco médio m aumenta. Portanto, auto-similaridade de 2ª ordem captura a propriedade de que a estrutura de correlação é exatamente ou assintoticamente preservada pelo processo agregado (PARK; WILLINGER, 2000).

Exemplo 1: *Ruído Gaussiano Fracionário e Movimento Browniano Fracionário*

O processo auto-similar mais simples é o Ruído Gaussiano Fracionário (FGN) que foi primeiro apresentado por Mandelbrot; Van Ness (1968). FGN é um processo Gaussiano estacionário com o coeficiente de correlação dado na Eq. (7). Leland et al. (1994) propuseram o FGN como um dos modelos apropriados para modelagem de tráfego LAN. Norros (1994 e 1995) especificou um modelo de tráfego para os dados capturados pelos Laboratórios Bellcore, usando o FGN mais uma constante. O modelo especificado por Norros (op. cit.) era apropriado para tráfego auto-similar suavizado, mas inadequado para tráfego mais explosivo. FGN é o processo de incremento do Movimento Browniano Fracionário (FBM) com parâmetro de auto-similaridade $H=0,5$.

Definição 6: *Uma variável aleatória X segue uma distribuição com cauda pesada (JOHNSON; KOTZ; BALAKRISHNAM, 1994) quando*

$$P(X > x) \sim L(x)x^{-a}, \quad a > 0, x \rightarrow \infty, \quad (9)$$

onde $L(x)$ é uma função que varia lentamente ao infinito quando $\lim_{x \rightarrow \infty} \frac{L(cx)}{L(x)} = 1, \forall c > 0$.

Exemplo 2: *Distribuição de Pareto*

É a distribuição de cauda pesada mais comumente referenciada nos artigos sobre tráfego de redes LAN / WAN. Sua função densidade de probabilidade (fdp) é

$$f(x) = \frac{ak^a}{x^{a+1}}, \quad a, k > 0, x \geq k.$$

Além de k ser maior que 0, ele é o mínimo do conjunto dos possíveis valores de x . A função de distribuição acumulada (fda) é

$$P(X > x) = \left(\frac{k}{x}\right)^a, \quad a > 0, k > 0, x \geq k. \quad (10)$$

Variáveis aleatórias de Pareto têm variância infinita se $a < 2$ e média infinita se $a < 1$. Em modelagem de tráfego foi proposta essa distribuição para modelar os intervalos de tempo de

chegada de pacotes observados em aplicações Telnet (PAXSON; FLOYD, 1995). Taqqu; Teverovsky (1998) e Willinger et al. (1997) mostraram que quando fontes ON/OFF são multiplexadas, com durações ON e OFF distribuídas por uma função de cauda pesada, o processo de contagem relacionado (número de pacotes que chegam por unidade de tempo) é assintoticamente auto-similar de segunda ordem. Gordon (1995) mostrou que se o intervalo de tempo entre pacotes segue uma distribuição de Pareto, então o processo de contagem de pacotes é assintoticamente auto-similar de segunda ordem com $a = 3 - 2H$, quando $0 < a < 1$ e $1 < a < 2$.

2.3.2.2 Processos com Dependência de Curta Duração e de Longa Duração

A Eq. (4) é usada na prática para inferir se um determinado processo estocástico possui características de dependência de longa duração ou de curta duração.

Definição 7: *Seja f a função de densidade espectral¹ de um processo estocástico estacionário. Um processo estocástico estacionário de segunda ordem, de tempo discreto e com uma função de autocorrelação $\rho(k), k \geq 0$, é dito possuir dependência de curta duração (DCD) se satisfaz às seguintes condições:*

$$(a) \sum_{k=0}^{\infty} \rho(k) < +\infty,$$

$$(b) f(0) < +\infty,$$

$$(c) \lim_{m \rightarrow \infty} \{mV[X_k^{(m)}]\} < +\infty.$$

Estas condições são (essencialmente) equivalentes, cada uma implica na outra, isto é, (a) \rightarrow (b) \rightarrow (c), porém o contrário não é mais válido, ou seja, (c) não implica (b) ou (a) (BERAN, 1994).

Um processo DCD é geralmente caracterizado por um decaimento do tipo exponencial da função de autocorrelação, o que implica na convergência da $\sum_{k=0}^{\infty} \rho(k)$. Portanto, esta condição faz com que o comportamento seja semelhante a de um ruído branco puro Gaussiano de segunda ordem. Isto explica uma suposição comumente feita na modelagem de tráfego

¹ Função de densidade espectral de um processo estocástico estacionário é definida como a transformada inversa de Fourier da sua função de autocorrelação.

tradicional que é a independência de observações, ou seja, na análise clássica de séries temporais, as observações separadas por um atraso (*lag*) temporal longo são consideradas praticamente independentes, apresentando um típico decaimento exponencial na sua função de autocorrelação.

Por outro lado, existe outra classe de processos os quais possuem as características de dependência de longa duração, como definidas a seguir.

Definição 8: Um processo estacionário no sentido amplo é chamado de dependente de longa duração (DLD) quando sua função de autocorrelação tem a seguinte forma

$$\rho(k) \sim L(k)k^{-a}, \quad a > 0, \quad \text{quando } k \rightarrow \infty \quad (11)$$

onde $L(k)$ é uma função que varia lentamente.

A Figura 2.7 mostra uma comparação da função de autocorrelação de um processo DLD com um processo dependente de curta duração. Observa-se que o decaimento da função de correlação de um processo DLD é mais lento que um DCD.

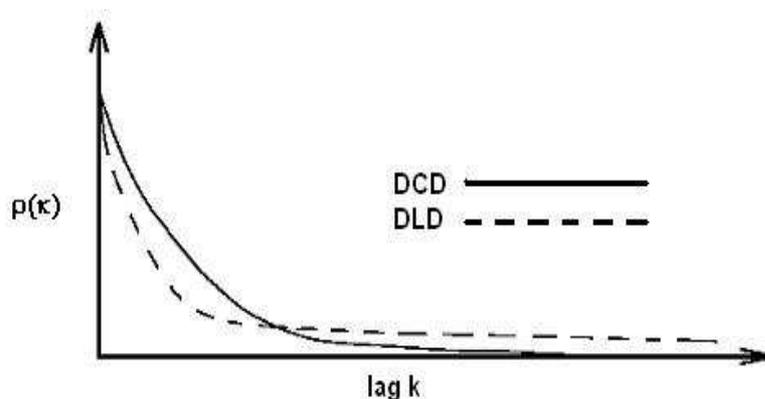


Figura 2.7. Função de autocorrelação de processos DCD e DLD

Um processo estocástico estacionário com função de autocorrelação $\rho(k), k \geq 0$ e função de densidade espectral f possui dependência de longa duração se satisfaz às seguintes condições:

(a) $\sum_{k=0}^{\infty} \rho(k) = +\infty,$

(b) $f(0) = +\infty,$

(c) $\lim_{m \rightarrow \infty} \{mV[X_j^{(m)}]\} = +\infty.$

O fato da função de autocorrelação ser não somável sugere que valores das autocovariâncias para atrasos longos são individualmente pequenos, porém o seu efeito acumulativo é considerável. As funções de autocorrelação para processos DLD decaem mais lentamente do que exponencialmente; este comportamento é conhecido também como decaimento hiperbólico.

Considerando a condição (a), é possível mostrar que essa somatória pode ser aproximada pela seguinte relação

$$\sum_{k=0}^{\infty} \rho(k) \approx Cm^{1-\alpha}, \quad (12)$$

onde $0 < \alpha < 1$. Um exemplo de uma função com decaimento hiperbólico é aquela com a seguinte estrutura matemática $\rho(k) \approx C_{\rho(k)} |k|^{-\alpha}$ para $k \rightarrow \infty$.

O modelo auto-similar pode ser caracterizado por um único parâmetro, denominado de parâmetro de Hurst (HURST, 1955) e definido por $H = 1 - \frac{\beta}{2}$. Processos dependentes de longa duração têm valores de H entre 0,5 e 1, onde o grau de auto-similaridade cresce quando H cresce. Pode-se estimar o grau de auto-similaridade através de diferentes métodos, entre eles a representação gráfica (variância x tempo) e a análise da estatística R/S (*rescaled readjusted range statistic*). A estatística de R/S está baseada em um enfoque gráfico heurístico (LELAND et al., 1994). Basicamente, mediante a utilização do processo agregado dado pela Eq. (5) calcula-se a razão dada por $R(n)/S(n)$ para cada bloco de comprimento m não sobreposto da série agregada (LELAND et al., 1994; HUANG et al., 1995). Matematicamente a razão por $R(n)/S(n)$ é definida como

$$R(n)/S(n) = 1/S(n)[\max(0, W_1, W_2, \dots, W_n) - \min(0, W_1, W_2, \dots, W_n)],$$

com $W_k = (X_1 + X_2 + \dots + X_k) - kX(n)$, $k = 1, 2, \dots, n$. O termo dado por $S(n)$ é a variância amostral de X_1, \dots, X_n .

Hurst (1955) observou que muitas séries temporais de ocorrência natural satisfaziam a relação

$$E\left(\frac{R(n)}{S(n)}\right) \sim cn^H, \quad n \rightarrow \infty, \quad (13)$$

onde c é finita positiva e independe de n e H . Naquela época (anos 50), isto era contraditório porque, segundo a teoria, essas séries temporais deveriam satisfazer a relação dada por

$$E\left(\frac{R(n)}{S(n)}\right) \sim cn^{0.5}, \quad n \rightarrow \infty. \quad (14)$$

A Eq. (14) é a definição clássica para o movimento Browniano (passeio aleatório) e a Eq. (13) é uma generalização deste processo estocástico conhecido como movimento Browniano fracionário. A discrepância entre a Eq. (13) e a Eq. (14) é conhecida como efeito de Hurst ou fenômeno de Hurst e pela estatística R/S o comportamento de um processo com DLD obedece à Eq. (13), enquanto que para um processo DCD o comportamento é governado pela Eq. (14).

O estimador R/S, nos processos auto-similares, apresenta a seguinte característica quando n é grande:

$$\left(\frac{R(n)}{S(n)}\right) \sim \left(\frac{n}{2}\right)^H, \quad \text{para } H > 0.5.$$

donde,

$$H \cong \frac{\log\left(\frac{R(n)}{S(n)}\right)}{(\log n - \log 2)}. \quad (15)$$

Pelo que foi exposto, observa-se que existem processos auto-similares que não são dependentes de longa duração e vice-versa. Por exemplo, o movimento Browniano fracionário (FBM) é um processo auto-similar que utiliza o ruído branco Gaussiano como incremento estacionário, sendo este último um processo não dependente de longa duração. De modo inverso, existem séries temporais baseadas no modelo FARIMA (*Fractional Auto-Regressive Integrated Moving Average*) geradas com dependência de longa duração que não são auto-similares. Porém, no caso de séries assintoticamente auto-similares de 2ª ordem, devido à restrição (por definição) do parâmetro de Hurst, $\frac{1}{2} < H < 1$, auto-similaridade implica em dependência de longa duração e vice-versa. Por esta razão e pelo fato de que processos auto-similares de 2ª ordem são empregados como a forma canônica de modelos de tráfego, é que muitas vezes se usam os termos auto-similaridade e dependência de longa duração de modo intercambiável, desde que o contexto permita.

2.4 Wavelet

Na análise de Fourier, os senos e cossenos são funções bases usadas para representar outras funções no domínio da frequência. A idéia da transformada, tanto em Fourier como em qualquer outra base, é aproximar uma função através de uma combinação linear da função mãe. Assim, toda série de quadrado somável, $\left(\sum_{i=1}^{\infty} x_i^2 < +\infty\right)$ pode ser construída a partir de uma única função ω (MORETTIN, 1997).

Dada uma função f não periódica, a localização de suas frequências pode ser feita através de uma função de modulação $\omega(x)$ que não tenha escala fixa e que seja localizada no tempo. Seja \mathfrak{R} o conjunto dos números reais. Define-se:

$$\omega_s(x) = |s|^{-p} \omega\left(\frac{x}{s}\right) = \frac{1}{|s|^p} \omega\left(\frac{x}{s}\right), \quad (16)$$

onde $s \in \mathfrak{R}$, $s \neq 0$ e $p \geq 0$.

Para localizar cada função ω_s no tempo, define-se para cada $t \in \mathfrak{R}$ a função

$$\omega_s^t(x) = \omega_s(x-t) = |s|^{-p} \omega\left(\frac{x-t}{s}\right) = \frac{1}{|s|^p} \omega\left(\frac{x-t}{s}\right).$$

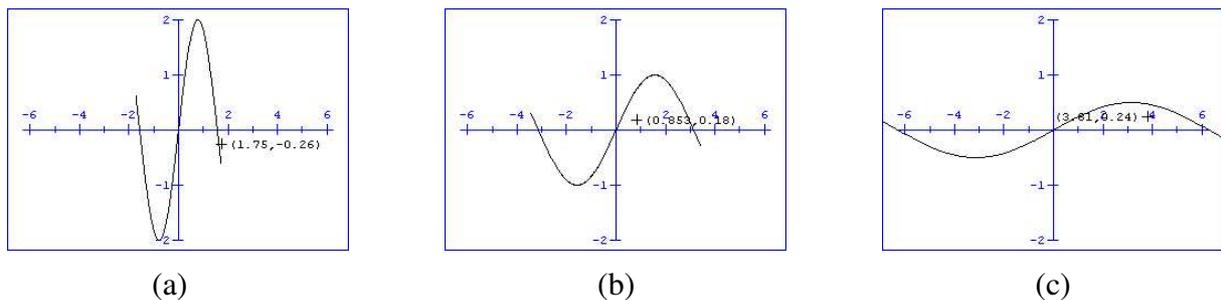


Figura 2.8. Escalas de uma função ω : (a) $s < 1$; (b) $s = 1$; (c) $s > 1$

Observa-se que a modulação da função ω pelo fator $\left(\frac{1}{|s|^p}\right)$ na equação (16) faz com que sua amplitude aumente quando a escala s diminui e vice-versa. Em termos de frequência, ω

possui altas frequências para s pequeno, ao passo que, se s aumenta a frequência de ω_s diminui. A Figura 2.8 ilustra estas observações.

Wavelets (DAUBECHIES, 1992) podem ser consideradas como sendo funções da forma $\omega_a^b(x)$, geradas a partir de translações (ou deslocamentos) e dilatações (ou contrações) de uma única função $\omega(x)$, chamada de função mãe ou base, que permitem a representação, tanto no domínio das abscissas (escala) quanto no da frequência, dos elementos de determinados subespaços do conjunto de todas funções. A variável independente t representa o tempo e seu domínio é referido com domínio do tempo. A forma geral de uma “família” de *wavelets* é

$$\omega_a^b(x) = \frac{1}{\sqrt{a}} \omega\left(\frac{x-b}{a}\right), a > 0, -\infty < b < \infty. \quad (17)$$

A função original, $\omega(x)$, corresponde ao caso em que $a = 1$ e $b = 0$. O procedimento de dilatações binárias e translações diádicas consiste em representar operações aplicadas à função original, de acordo com

$$\omega_j^k(x) = 2^{-j/2} \omega(2^{-j}x - k), j, k \in Z,$$

fazendo em (17) $a = 2^{-j}$, $b = k2^j$, onde $j, k \in Z = \{0, \pm 1, \dots\}$. 2^{-j} representa a dilatação que caracteriza a função $\omega(x)$ em diferentes escalas e k representa a translação no tempo. Assim, a função ω_j^k é obtida pela dilatação e translação da função mãe $\omega(x)$, tendo, portanto, a mesma estrutura.

A *wavelet* de Haar (GOMES; VELHO; GOLDSTEIN, 1999), uma função passo (*step*) que assume os valores 1 e -1 , nos intervalos $[0, 1/2)$ e $[1/2, 1)$, respectivamente, é utilizada neste trabalho para mostrar dilatações e translações sobre a função original $\omega(x)$. O gráfico da *wavelet* de Haar é dado na Figura 2.9. As funções $\omega_0^0, \omega_1^0, \omega_1^1, \omega_2^0, \omega_2^1, \omega_2^2, \omega_2^3$ da *wavelet* de Haar são mostradas na Figura 2.10.

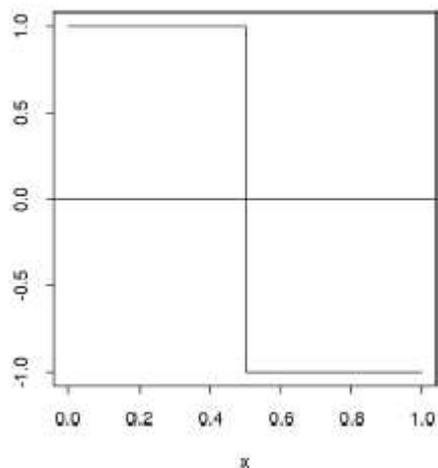


Figura 2.9. *Wavelet de Haar*

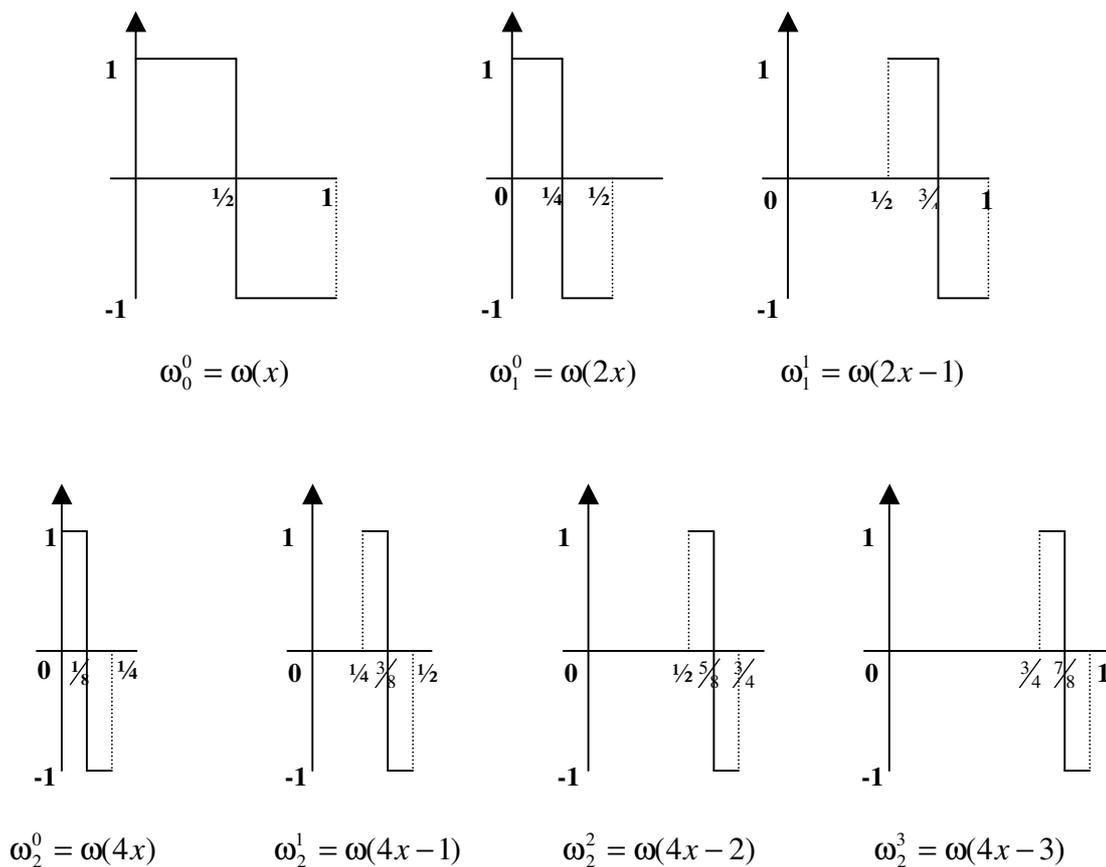


Figura 2.10. Dilatações e translações da *wavelet de Haar* em $[0,1]$

Existem muitos trabalhos na literatura que tratam de *wavelets* como uma ferramenta contemporânea com usos relevantes em processos estocásticos, processamento de sinais, codificação e compressão de imagens, etc. Porém o primeiro livro em português sobre o assunto só saiu em julho de 1997, no 21º. Colóquio Brasileiro de Matemática (GOMES; VELHO; GOLDSTEIN, 1997).

É importante salientar, que as *wavelets* estão localizadas no tempo (ou espaço), diferentemente do ocorre com as funções trigonométricas. Este comportamento torna-as adequadas para análise de séries não-estacionárias, contendo transitoriedades e estruturas tipo fractais (MORETTIN, 1997).

2.4.1 Transformada de Wavelet Discreta

Seja $x = (x_0, x_1, \dots, x_T)$ um vetor de dados de tamanho 2^n , isto é, $T = 2^n - 1$, $n \in \mathbb{Z}^+ = \{1, 2, \dots\}$. Define-se a transformada de *wavelet* discreta de x , com respeito à *wavelet* mãe ω , como

$$d_j^k = \sum_{t=0}^T x_t \omega_j^k(t/T), \quad j = 0, 1, \dots, n-1 \text{ e } k = 0, 1, \dots, 2^j - 1, \quad (18)$$

onde os d_j^k são chamados de coeficientes. O somatório de (18) gera $2^n - 1$ coeficientes quando deveria gerar 2^n , pois o vetor contém 2^n observações. Donoho et al. (1995) denotam o coeficiente não referenciado em (18) por d_{-1}^0 .

A transformada da Eq. (18) pode ser escrita na forma matricial $d = WX$ (MORETTIN, 1997). Sob determinadas condições, pode-se inverter a forma matricial para $X = W'd$, onde W' indica a transposta de W . Contudo, na prática, a transformada de *wavelet* discreta não é obtida através desta multiplicação matricial, mas sim através de um algoritmo, chamado piramidal, com complexidade $O(T)$.

O vetor de dados x pode ser associado a uma função constante f sobre o intervalo $[0, 1)$ gerada por x , da seguinte forma:

$$f(t) = \sum_{k=0}^{2^n-1} x_k \cdot I_{\{k2^{-n} \leq t < (k+1)2^{-n}\}},$$

onde I é a função indicadora.

Considerando, por exemplo, a *wavelet* de Haar, a decomposição de f tem a forma

$$f(t) = d_{-1}^0 \phi(t) + \sum_{j=0}^{2^n - 12^{j-1}} \sum_{k=0}^{2^n - 12^{j-1}} d_j^k \omega_j^k(t), \quad (19)$$

onde $\phi(t) = 1, 0 \leq t < 1$, é a função escala e d_{-1}^0 é o coeficiente de escala. A soma com respeito a j é finita porque f é uma função passo e pode ser exatamente descrita pela resolução até o nível $(2^n - 1)$. Para cada nível, a soma com respeito a k é também finita porque o domínio de f é finito.

Exemplificando (19), utiliza-se a seguir $f(t) = (1, 0, -3, 2, 1, 0, 1, 2)$ para realizar a decomposição da *wavelet* de Haar. A função f correspondente é dada pela Figura 2.11.

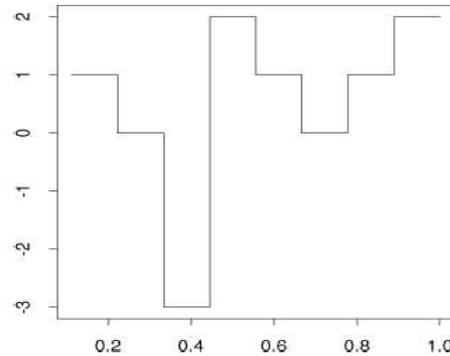


Figura 2.11. “Função de dados” em $[0,1]$

A seguinte equação matriz mostra a conexão entre $f(t)$ e os coeficientes da *wavelet* Haar, denotando as constantes $2^j = 1, \sqrt{2}, 2$ nos níveis de resolução ($j = 0, 1, 2$) correspondentes.

$$\begin{bmatrix} 1 \\ 0 \\ -3 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \sqrt{2} & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & \sqrt{2} & 0 & -2 & 0 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & -2 & 0 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & 2 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & -2 & 0 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & 2 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & -2 \end{bmatrix} \cdot \begin{bmatrix} d_{-1}^0 \\ d_0^0 \\ d_1^0 \\ d_1^1 \\ d_2^0 \\ d_2^1 \\ d_2^2 \\ d_2^3 \end{bmatrix} \quad (20)$$

O vetor solução de (20) é

$$\begin{bmatrix} d_{-1}^0 \\ d_0^0 \\ d_1^0 \\ d_1^1 \\ d_2^0 \\ d_2^1 \\ d_2^2 \\ d_2^3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} \\ \frac{1}{4} \\ -\frac{5}{4} \\ \frac{1}{4} \\ -\frac{1}{4} \end{bmatrix}.$$

Portanto,

$$f = \frac{1}{2}\phi - \frac{1}{2}\omega_0^0 + \frac{1}{2\sqrt{2}}\omega_1^0 - \frac{1}{2\sqrt{2}}\omega_1^1 + \frac{1}{4}\omega_2^0 - \frac{5}{4}\omega_2^1 + \frac{1}{4}\omega_2^2 - \frac{1}{4}\omega_2^3.$$

A solução é fácil de se encontrar. Por exemplo, quando $t \in [0, \frac{1}{8})$, então $f(t) = \frac{1}{2} - \frac{1}{2} \cdot 1 + \frac{1}{2\sqrt{2}} \cdot \sqrt{2} + \frac{1}{4} \cdot 2 = 1$. Contudo, a transformada de *wavelet* discreta não é obtida através desta multiplicação matricial, mas através de um esquema piramidal de multiresolução (MALLAT, 1989) e a transformada inversa através de um algoritmo de reconstrução dos dados (síntese), a partir dos coeficientes (MALLAT, op. cit.).

2.4.2 Algoritmo Piramidal

Como citado anteriormente, existem funções que podem ser escritas como uma combinação linear de funções base, como por exemplo, $\sqrt{2}\phi(2x - k)$. Assim, a função ϕ , conhecida como função escala ou *wavelet* pai, pode ser escrita da seguinte forma

$$\phi(x) = \sqrt{2} \cdot \sum_k l_k \phi(2x - k) \quad (21)$$

onde l_k são coeficientes definidos em $\langle \phi(x), \sqrt{2} \cdot \phi(2x - k) \rangle$ e dados pela chamada equação de dilatação

$$l_k = \sqrt{2} \int_{-\infty}^{\infty} \phi(x) \cdot \phi(2x - k) dx. \quad (22)$$

A função da Eq. (21) pode gerar uma família de funções, a partir das operações de dilatação e translação no tempo citadas anteriormente, do tipo

$$\phi_j^k(x) = 2^{\frac{j}{2}} \phi(2^j x - k), \quad j, k \in \mathbb{Z}. \quad (23)$$

A função ω também pode ser obtida de ϕ através de

$$\omega(x) = \sqrt{2} \sum_k h_k \phi(2x - k), \quad (24)$$

onde h_k são coeficientes definidos em $\langle \omega(x), \sqrt{2} \cdot \phi(2x - k) \rangle$.

Na realidade, l_k e h_k são coeficientes de filtros de passa-baixa e passa-alta usados para calcular a transformada de *wavelet* discreta. Por exemplo, como foi visto anteriormente, a função ω na *wavelet* de Haar assume os seguintes valores

$$\omega(x) = \begin{cases} 1, & 0 \leq x < 1/2, \\ -1, & 1/2 \leq x < 1, \\ 0, & \text{caso contrário,} \end{cases}$$

para a qual a função escala $\phi(x) = 1, 0 \leq x \leq 1$ e a família de funções geradas a partir de ω assume os valores abaixo

$$\omega_j^k = \begin{cases} 2^{j/2}, & 2^{-j} k \leq x < 2^{-j} (k + 1/2), \\ -2^{j/2}, & 2^{-j} (k + 1/2) \leq x < 2^{-j} (k + 1), \\ 0, & \text{caso contrário.} \end{cases}$$

Para a *wavelet* Haar, a equação (21) fica

$$\phi(x) = \sqrt{2} \cdot [l_0 \cdot \phi(2x) + l_1 \cdot \phi(2x - 1)] = \frac{1}{\sqrt{2}} \sqrt{2} \phi(2x) + \frac{1}{\sqrt{2}} \sqrt{2} \phi(2x - 1),$$

logo $l_0 = l_1 = 1/\sqrt{2}$. Do mesmo modo, $h_0 = -h_1 = 1/\sqrt{2}$.

O procedimento do algoritmo piramidal para cálculo da transformada de *wavelet* discreta está ilustrado na Figura 2.12, onde L indica o filtro de passa-baixa, H o filtro de passa-alta e $\downarrow 2$ indica a operação de decimação por 2, ou seja, a cada duas saídas do filtro, despreza-se uma. A operação começa com um vetor de dados $c_n = (c_n^0, \dots, c_n^{2^n - 1})$ de tamanho $2^n, n = 1, 2, \dots$. No j -ésimo passo, o algoritmo calcula c_j^k e d_j^k a partir dos coeficientes suaves do nível $j - 1, c_{j-1}^k$, através de

$$c_j^k = \sum_n l_{2k-n} \cdot c_{j-1}^n,$$

$$d_j^k = \sum_n h_{2k-n} \cdot c_{j-1}^n.$$

A saída do algoritmo é o conjunto dos coeficientes de detalhe d_j , $j = 0, 1, \dots, n-1$, e dos coeficientes suaves $c_j = (c_j^0, \dots, c_j^{2^n-1})$.

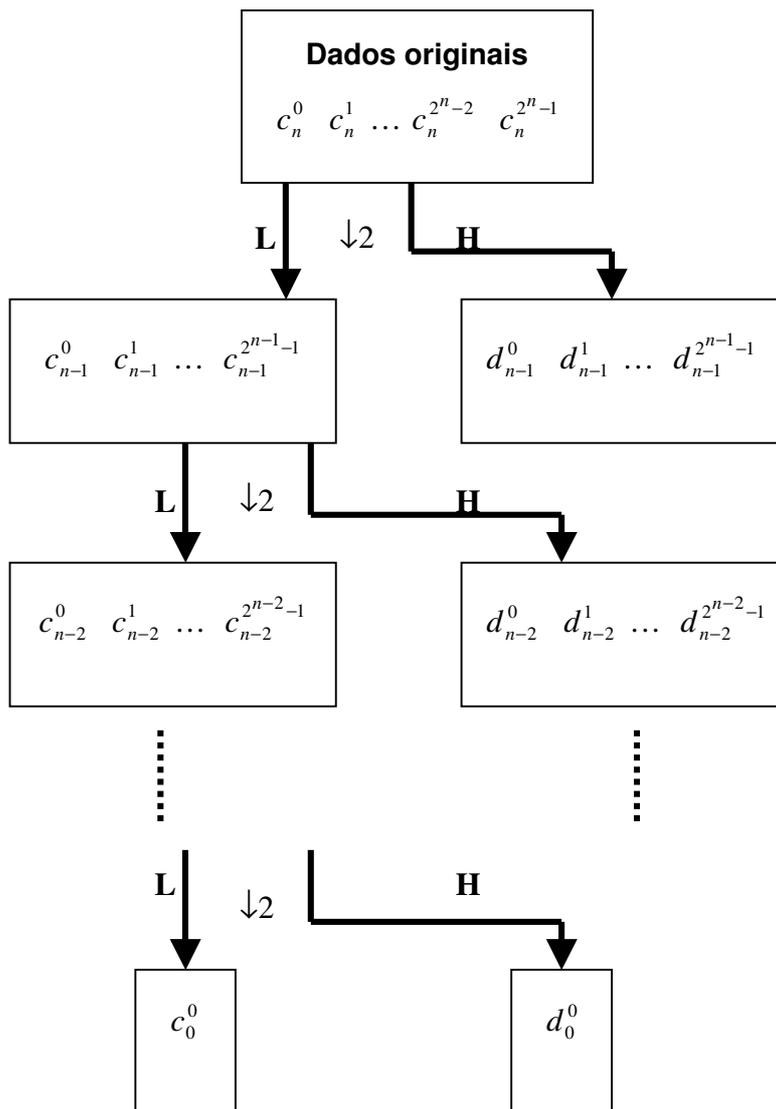


Figura 2.12. Algoritmo piramidal

Apresenta-se, a seguir, um exemplo numérico da transformada de *wavelet* discreta, mostrando inicialmente o procedimento do algoritmo piramidal na Figura 2.13 e, em seguida, a forma matricial (25) de decomposição da *wavelet* de Haar.

Exemplo 3: Considere o vetor de dados (4, -1, 3, 2, 1, 4, -2, 2). Para a base de Haar os coeficientes dos filtros são:

$$L_0 = \frac{\sqrt{2}}{2}; \quad L_1 = \frac{\sqrt{2}}{2}; \quad H_0 = \frac{\sqrt{2}}{2}; \quad H_1 = -\frac{\sqrt{2}}{2}.$$

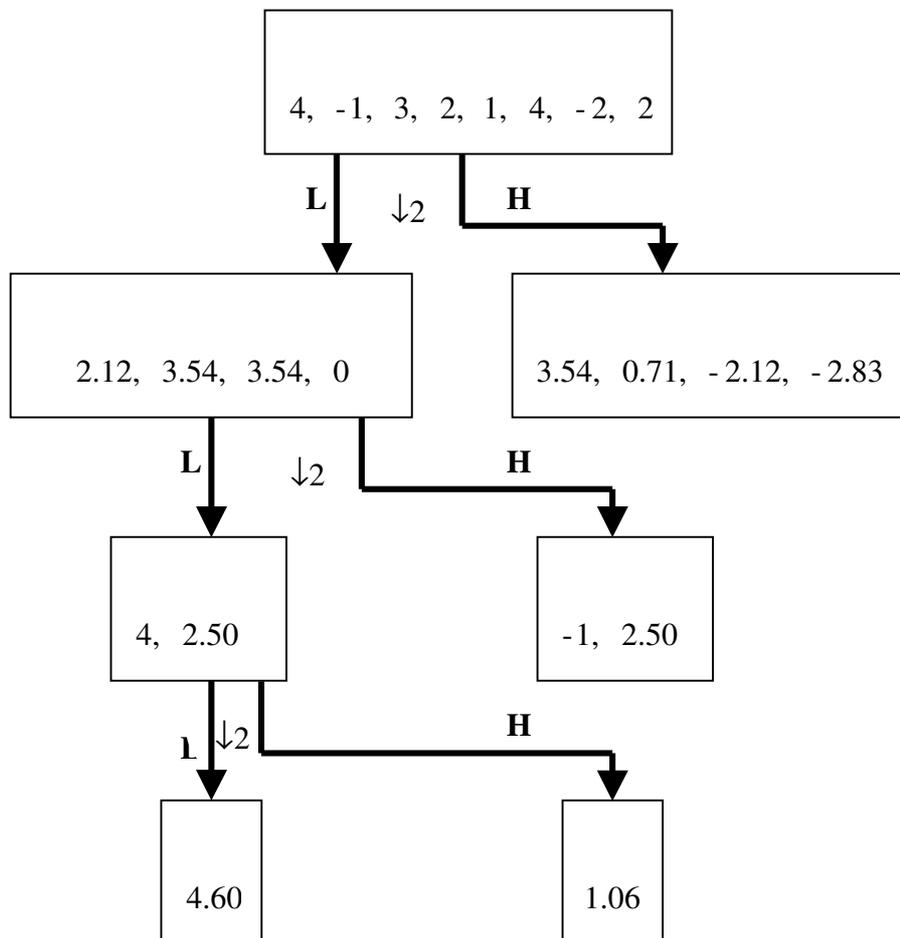


Figura 2.13. Exemplo de algoritmo piramidal

Deste modo, tem-se a seguinte forma matricial para a decomposição

$$\begin{bmatrix} c_j^0 \\ d_j^0 \\ c_j^1 \\ d_j^1 \\ c_j^2 \\ d_j^2 \\ c_j^3 \\ d_j^3 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} c_{j+1}^0 = 4 \\ c_{j+1}^1 = -1 \\ c_{j+1}^2 = 3 \\ c_{j+1}^3 = 2 \\ c_{j+1}^4 = 1 \\ c_{j+1}^5 = 4 \\ c_{j+1}^6 = -2 \\ c_{j+1}^7 = 2 \end{bmatrix} \quad (25)$$

Um fato importante que se pode observar na Figura 2.13 é que, quanto maior é a variação entre os números da seqüência, maior é o coeficiente de *wavelet*. Isto reflete claramente, neste exemplo, que a transformada de *wavelet* detecta exatamente as variações da série. Ainda neste exemplo pode-se ver que os coeficientes de escala extraem a média enquanto os coeficientes de *wavelet* extraem a diferença ou detalhes.

2.4.3 Esquema *Lifting*

O esquema *lifting* (SWELDENS, 1997), desenvolvido em 1994, foi inspirado no trabalho de Donoho (1992) relacionado com a interpolação da transformada de *wavelet*. A forma de explicar este esquema, nesta tese, é feita através de um exemplo simples da *wavelet* de Haar.

Sejam dois números a e b consecutivos de uma amostra qualquer de uma série de tamanho 2^n . Se os números a e b têm alguma correlação, pode-se fazer uma análise deles através de uma transformação linear simples, substituindo a e b por sua média s e por sua diferença d :

$$\begin{aligned} s &= \frac{a+b}{2}, \\ d &= b-a. \end{aligned} \tag{26}$$

Se a e b são correlacionados, o valor absoluto esperado da sua diferença d será pequeno ou zero no caso de $a = b$. Essa transformação não perde qualquer informação porque, dado s e d , sempre é possível recuperar a e b através de:

$$\begin{aligned} a &= s - \frac{d}{2}, \\ b &= s + \frac{d}{2}. \end{aligned}$$

Seja s_n uma seqüência de tamanho 2^n , isto é, com 2^n de valores s_n^k , tal que,

$$s_n = \{s_n^k \mid 0 \leq k \leq 2^n - 1, n = 1, 2, \dots\},$$

e considere a transformação da Eq. (26). Aplicando-se a transformação da média e da diferença sobre cada par $a = s^{2k}$ e $b = s^{2k+1}$ consecutivos, obtém-se 2^{n-1} pares de médias s_{n-1}^k e de diferenças d_{n-1}^k , $k = 0, \dots, 2^{n-1} - 1$, onde:

$$s_{n-1}^k = \frac{s_n^{2k} + s_n^{2k+1}}{2},$$

$$d_{n-1}^k = s_n^{2k+1} - s_n^{2k}.$$

A série de entrada s_n , que tem 2^n amostras, é dividida em duas séries: (i) s_{n-1} com 2^{n-1} médias denotadas por s_{n-1}^k e (ii) d_{n-1} com 2^{n-1} diferenças denotadas por d_{n-1}^k . A reconstrução da série original s_n pode ser feita através das médias s_{n-1} e das diferenças d_{n-1} .

As médias s_{n-1} , nesta transformação, representam a resolução “grossa” da série s_n e as diferenças d_{n-1} representam as informações necessárias para se reconstruir a série original a partir da representação grossa.

Aplicando-se a mesma transformação na série s_{n-1} (série grossa) e tomando-se as médias e as diferenças, obtém-se uma nova série grossa s_{n-2} e uma outra série de diferenças d_{n-2} , onde cada uma delas contém 2^{n-2} amostras. Essas transformações podem ser feitas n vezes até que a última série obtida tenha somente uma amostra, como ilustrado na Figura 2.14.

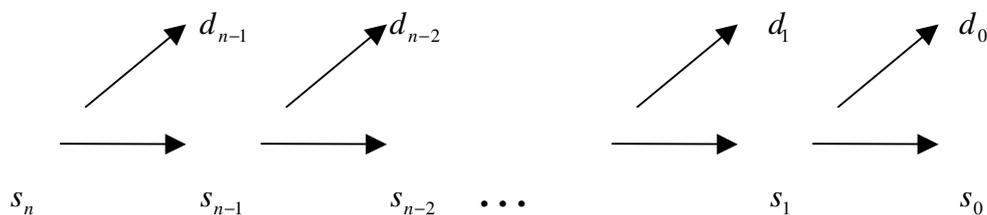


Figura 2.14. Esquema *lifting* - divisão recursiva em médias e diferenças

Uma das vantagens do esquema *lifting* sobre outras formas de apresentação da transformada de *wavelet*, como a descrita na seção anterior, é a sua simplicidade. Em muitos casos, a transformada de *wavelet* é apresentada usando a estrutura clássica no domínio da frequência, com muitos detalhes matemáticos. Outra vantagem do esquema *lifting* é a sua simplicidade de implementação em uma linguagem de programação. A implementação pode ser feita em dois passos: (i) calcula-se a diferença, $d = b - a$, e armazena em b ; (ii) como o

valor original de b foi perdido, calcula-se a média por $s = a + \frac{d}{2}$. Isto fornece o mesmo

resultado porque $a + \frac{d}{2} = a + \frac{b-a}{2} = \frac{(a+b)}{2}$.

A implementação em dois passos não requer no código do programa armazenamento auxiliar, tornando-o mais conciso, uma vez que o valor de b é sobreposto por d e o valor de a por s . Uma implementação na linguagem C é dada por

$$\begin{aligned} b &= a, \\ a &= b/2. \end{aligned}$$

As aplicações das *wavelets* para análise, estimação e síntese de dados em escala são inúmeras. A presença do comportamento do tráfego de redes em escala é encontrada nos trabalhos de Beran et al. (1995), Leland et al. (1994) e Willinger et al. (1997). Técnicas baseadas em *wavelets* para analisar *traces* de tráfego de redes (ABRY; VEITCH, 1998) têm sido investigadas devido à habilidade das *wavelets* em representar fenômenos em diferentes tipos de escala. Em função dessa habilidade de localizar um dado sinal no tempo e na escala, as *wavelets* podem descrever, identificar e detectar tráfego multifractal (FELDMAN; GILBERT; WILLINGER, 1998). Kaplan; Kuo (1993) investigaram extensivamente a estrutura de correlação de processos FGN, dependentes de longa duração, e mostraram que estes processos, depois de transformados por uma *wavelet* de Haar, apresentam uma série de coeficientes, onde a soma das correlações converge para uma constante. Estudos teóricos feitos por Kaplan; Kuo (1993) e Flandrin (1993) sobre a estrutura de autocorrelação dos coeficientes destes processos indicam que a transformada de *wavelet* reduz a dependência temporal.

2.5 Métodos de Estimação de Tendência em Séries Temporais

Uma série X_1, \dots, X_N pode ser escrita como a soma de quatro componentes individuais: tendência, ciclo, sazonalidade e aleatoriedade. Wheelwright; Makridakis (1985) expressam o relacionamento entre as componentes não observáveis da série temporal, através da equação

$$Z_t = f(S_t, T_t, C_t, E_t) \quad (27)$$

onde, considerando o período t , S_t corresponde à componente sazonal; T_t é a tendência; C_t é a componente de ciclo e E_t é a componente aleatória.

A tendência representa o aumento ou declínio gradual nos valores das observações da série. Em geral pode ser representada por uma função “suave” do tempo t . As componentes ciclo e sazonal representam as flutuações da série de acordo com algum fator. Removendo-se as componentes de sazonalidade, ciclo e tendência, a componente aleatória ou residual fica determinada. A suposição usual é que E_t seja uma série puramente aleatória ou ruído branco, com média zero e variância constante.

Wheelwright; Makridakis (1985) afirmam que vários procedimentos para a decomposição de séries temporais foram desenvolvidos, cada qual tentando isolar as componentes não observáveis da série da forma mais exata possível. O modelo de decomposição é dito aditivo quando uma componente não depende das outras. Caso contrário, o modelo é chamado de multiplicativo. O objetivo desses procedimentos consiste em remover cada uma das componentes, permitindo que o comportamento da série temporal seja melhor compreendido e, conseqüentemente, prognosticar valores futuros mais apropriados.

Seja X_1, \dots, X_N uma série temporal. Considerando o modelo de decomposição aditivo e removendo-se as componentes de ciclo e sazonalidade, a série pode ser representada por

$$X_t = T_t + E_t. \quad (28)$$

Há vários métodos de se estimar T_t . Os mais utilizados consistem em:

- (i) ajustar a uma função do tempo, como um polinômio, uma exponencial ou outra função “suave” de t (métodos paramétricos);
- (ii) suavizar (ou filtrar) os valores da série ao redor de um ponto para estimar a tendência naquele ponto (método não-paramétrico);
- (iii) utilizar diferenças (método não-paramétrico).

Estimando-se a tendência através de \hat{T}_t , pode-se obter a série ajustada para tendência ou livre de tendência devido à validade de (28)

$$X_t - \hat{T}_t = \hat{E}_t.$$

2.5.1 Teste para Tendência

A inspeção gráfica de uma série temporal pode revelar características importantes como tendência, observações aberrantes (*outliers*) etc. Além do gráfico, testes de hipótese são

utilizados para verificar se realmente existe tendência na série. Isto pode ser feito antes ou depois da estimação de T_t . As hipóteses a serem testadas são de existência (H_1) ou não (H_0) de tendência, com base nas observações X_t , $t = 1, \dots, N$.

Os testes antes da estimação de T_t são todos não-paramétricos, que, em geral, se baseiam na suposição de que as observações constituem uma amostra de uma população e, portanto, independentes.

O teste de seqüências de Wald-Wolfowitz (MORETTIN; TOLOI, 1987) utiliza a mediana da série, M , para comparar com cada valor X_t , atribuindo um símbolo, (A) por exemplo, se $X_t \geq M$ e (B) se $X_t < M$. A estatística do teste é: E_1 = número total de seqüências de símbolos iguais. Rejeita-se H_0 se há poucas seqüências, ou seja, se E_1 é pequeno.

O teste do sinal de Cox-Stuart (MORETTIN; TOLOI, 1987) agrupa as observações em pares $(X_1, X_{1+c}), (X_2, X_{2+c}), \dots, (X_{N-c}, X_N)$, onde $c = \frac{N}{2}$, se N é par e $c = \frac{N+1}{2}$, se N é ímpar. A cada par associa-se o sinal + se $X_i < X_{i+c}$ e o sinal - se $X_i > X_{i+c}$, eliminando os empates. A estatística do teste é: E_2 = número de pares com sinal +. Rejeita-se (ou não) a existência de tendência baseado na distribuição binomial, com parâmetros n e $p = 1/2$ se $n \leq 20$, ou na normal com parâmetros $\mu = n/2$ e $\sigma^2 = n/4$, onde n é o número de observações, para um dado nível de significância α .

O último teste não-paramétrico, baseado no coeficiente de correlação de Spearman (MORETTIN; TOLOI, 1987), consiste na identificação do posto "natural" de cada X_t e do posto após a classificação dos X_t em ordem ascendente, dentre as N observações. O teste é feito em função do coeficiente de correlação de Spearman dado por

$$\rho = 1 - \frac{6E_3}{N(N^2 - 1)}.$$

A estatística do teste é dada por

$$E_3 = \sum_{j=1}^N (R_j - j)^2,$$

onde $j = 1, \dots, N$ são os postos “naturais” e R_j os postos após a classificação dos X_j . Rejeita-se H_0 se $E_3 \leq c_1$ ou $E_3 \geq c_2$, onde os valores de c_1 e c_2 estão tabelados em Conover (1971).

2.6 Comentários

Este capítulo expõe os conceitos relacionados com congestionamentos de redes e as soluções existentes para o problema, destacando as políticas preventivas e os mecanismos de gerenciamento de filas, as tecnologias usadas para atender requisitos de qualidade de serviço e os processos de organização de fluxos de tráfego que tratam das questões de avaliação e otimização de redes. Foram apresentados também os modelos de tráfego usuais, sobressaindo os processos auto-similar e os dependentes de longa duração. Em seguida foi exposto o conceito de *wavelet* e de transformada discreta, destacando as formas de decomposição e reconstrução de séries finitas. Por fim, apresentaram-se os modelos de estimação de tendência de séries temporais, utilizados na revelação de características de uma série de dados.

O assunto exposto neste capítulo é por demais extenso, não se esgotando aqui o conhecimento que existe, através de publicações, livros etc., sobre o domínio da matéria.

Capítulo 3

Metodologia e Cenário

Este capítulo apresenta a arquitetura e as especificações do mecanismo de prognóstico de congestionamento proposto nesta tese, denominado de ProCon, e o cenário de execução. O capítulo inicia com a descrição da localização do mecanismo de prognóstico de congestionamento, ou seja, em que lugar da rede o ProCon vai executar suas funções. A seção seguinte, 3.2, descreve a arquitetura, as funções e os procedimentos executados pelo ProCon. As especificações dos parâmetros usados pelo mecanismo de prognóstico, tais como bases de *wavelet*, políticas de redução de ruído, tipos de limiares de coeficientes, percentuais de redução e indicadores de congestionamento, são expostas na Seção 3.3. A seção seguinte, 3.4, especifica como a tendência do tráfego é inferida. A Seção 3.5 descreve o tráfego utilizado nos testes do mecanismo de prognóstico de congestionamento e os parâmetros do programa gerador de tráfego simulado. Vale salientar que, face à dificuldade de se obter tráfego real congestionado, optou-se pela utilização de tráfego simulado para fazer funcionar experimentalmente o mecanismo citado. Os programas que compõem o mecanismo de prognóstico de congestionamento e os que foram desenvolvidos para auxiliar os testes do ProCon, bem como suas funções estão descritas na Seção 3.6. As duas últimas seções, 3.7 e 3.8, descrevem, respectivamente, os cenários de execução e os comentários finais.

3.1 Localização do Mecanismo de Prognóstico de Congestionamento

Os atuais mecanismos de controle de congestionamento baseados em gerenciamento ativo de filas, como o já citado RED (BRADEN et al., 1998), estão localizados nos nodos intermediários das redes (roteadores, *switches*, etc.), onde acontece a maioria dos congestionamentos. Na realidade, a maioria dos congestionamentos ocorre exatamente nas portas de saída dos roteadores, em situações como a que está ilustrada na Figura 3.1. As linhas tracejadas na figura mostram que os fluxos de tráfego provenientes das várias portas de entrada estão todos concentrados em uma única porta de saída. Neste caso, se a fila do *buffer* da porta de saída não comportar os fluxos provenientes das portas de entrada, o mecanismo de gerenciamento começa a descartar pacotes até atingir um ponto de equilíbrio entre pacotes que chegam e pacotes que saem e/ou que são descartados. Isto gera uma situação que

contraria o próprio mecanismo, à medida que os pacotes descartados deverão ser retransmitidos pelo usuário aumentando, conseqüentemente, o congestionamento.

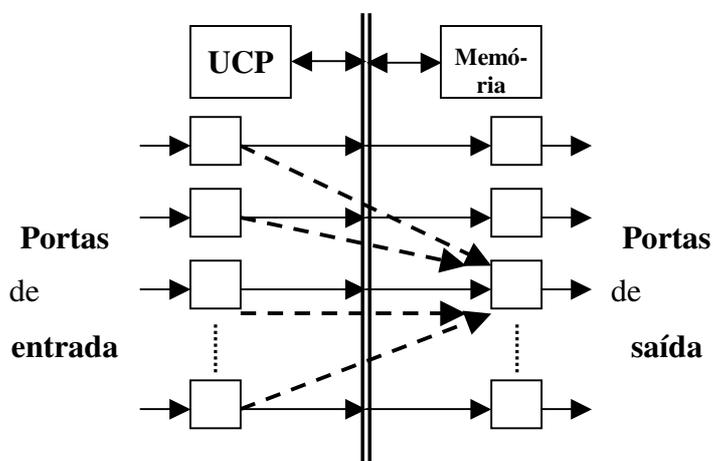


Figura 3.1. Arquitetura de um roteador baseado em barramento

O problema do congestionamento é mais grave em roteadores baseados em barramento com um único processador. Neste caso, os pacotes que chegam nas *interfaces* de entrada (NIC – *Network Interface Card*) são enviados a UCP (Unidade Central de Processamento) e esta, por sua vez, determina seus endereços de destino, enviando-os para as *interfaces* de saída apropriadas. Os dados desses roteadores são comumente armazenados em uma memória centralizada (BRYANT; BRASH, 1993; GUPTA; LIN; MCKEOWN, 1998) e, por isso, cruzam o barramento duas vezes, criando a possibilidade de engarrafamento do sistema.

O problema também ocorre em roteadores com tecnologia mais recente, baseados em malha de comutação (*switch fabric*), como os roteadores *multigigabit* (CISCO, 1997; PARTRIDGE et al.; 1998). Esses foram projetados para contornar os congestionamentos ocorridos em roteadores com a arquitetura baseada em barramento. Todavia, apesar de não existir engarrafamento nas interconexões entre os NIC, o problema do congestionamento ainda persiste nesses roteadores quando o fluxo é direcionado para um única saída.

O mecanismo de prognóstico de congestionamento proposto nesta tese, denominado ProCon, pode ser instalado tanto em roteadores baseados em barramento quanto em baseados em fábrica de comutação. O mecanismo pode atuar sobre os fluxos enviados para as portas de saída antes de serem armazenados nas filas (*buffers*) e ser executado tanto na UCP dos roteadores quanto nas *interfaces* de saída (NIC), desde que estas tenham capacidade de processamento. O ProCon poderia também agir sobre os fluxos que chegam em um roteador, isto é, nas portas de entrada, identificando os canais que estão em situação de possível

congestionamento. A Figura 3.2 mostra o diagrama funcional de um roteador com capacidade de processamento nas *interfaces* de rede e o posicionamento do mecanismo proposto.

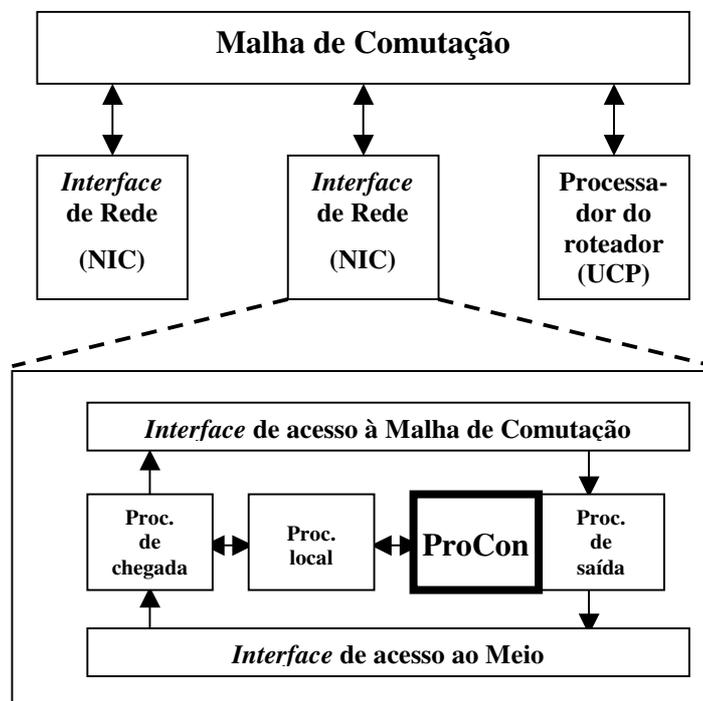


Figura 3.2. Diagrama funcional de um roteador

Na Figura 3.2, cada NIC pode prover capacidade de processamento e espaço de armazenamento necessários as operações que atuam sobre os pacotes que fluem pelo roteador. Os componentes funcionais de um NIC podem processar tanto o tráfego de entrada quanto o de saída. O mecanismo de prognóstico age sobre o tráfego enviado para o componente de saída antes dele ser armazenado no *buffer* desse componente. A *interface* de acesso ao meio executa as funções da camada física e da sub-camada MAC (*Media Access Control*). A *interface* de acesso à Malha de Comutação é responsável pela preparação do pacote para sua “jornada” através da malha. As *interfaces* de rede estão interconectadas via comutador que permite troca de dados e de mensagens de controle.

3.2 Arquitetura do Mecanismo de Prognóstico de Congestionamento

O mecanismo de prognóstico proposto nesta tese utiliza o processo de contagem do número de bits ou de pacotes que chegam no componente ou na porta de saída de um dado roteador, no intervalo de tempo $[0, t]$. Dentro de uma janela de tempo T (janela de observação), onde $T = 2^n$, $n = 1, 2, \dots$, e $T > t$, a série formada pelo processo de contagem é dinâmica porque à

medida que a próxima observação ou coleta (número de pacotes no intervalo $[0,t]$) se torna disponível, a observação mais antiga é desprezada e a mais nova incorporada à série. Assim, todos os componentes funcionais do mecanismo estão continuamente trabalhando com séries atualizadas que representam o comportamento do tráfego em tempo real.

O tamanho da janela de observação para o mecanismo corresponde ao número de observações da série em um intervalo de tempo T , onde cada observação representa o número de bits ou de pacotes contados no intervalo de tempo $[0,t]$. Convém salientar que, como a transformada de *wavelet* discreta trabalha somente com séries de tamanho 2^n , onde $n = 1,2,\dots$, como foi visto no Capítulo 2, o tamanho da janela de observação para o mecanismo está também restrito a esta lei de formação.

O tamanho da janela de observação também depende do tempo de processamento do mecanismo e da velocidade da rede. Atualmente, existem redes de alta velocidade, tanto locais (LAN) quanto metropolitanas (MAN) e de longa distância (WAN) que operam com taxas de transmissão de pelo menos 100 Mbps (milhões de bits por segundo) e são usadas, usualmente, na interconexão de computadores com grande capacidade de processamento ou na interconexão de sistemas computacionais e de redes locais. Para essas redes, a dinâmica de renovação da série, isto é, a incorporação de novas observações e o descarte de observações mais antigas, é limitada pelo desempenho de processamento do mecanismo. Por exemplo, dependendo da velocidade da rede e da capacidade de processamento do nodo que contém o mecanismo, é possível trabalhar com séries de tamanho igual a 512, onde a cada 256 novas observações incorporam-se estas e descartam-se as 256 mais antigas. Neste caso, um dos procedimentos do mecanismo proposto (verificação da tendência) compara as primeiras 256 observações com as 256 últimas para verificar a propensão do tráfego. Após a comparação, o mecanismo descarta as 256 últimas e admite 256 novas observações. Portanto, a cada intervalo de tempo T todos os procedimentos do mecanismo são executados e uma nova série com $T/2$ novas observações é submetida novamente à função inicial.

A coleta de tráfego para o mecanismo deve ser realizada continuamente e demanda um tempo T para que a janela de observação seja preenchida totalmente. Dependendo da velocidade do barramento e da UCP do roteador, este tempo T é da ordem de milissegundos e, por isso, o mecanismo proposto deve executar suas funções em um tempo menor que T . Porém, o mecanismo não pode ser executado com a máxima velocidade da UCP, pois deve levar em conta o tempo necessário para preenchimento da referida janela. Portanto, o

mecanismo deve executar seus procedimentos no momento certo e de forma constante, mesmo que o processamento da janela anterior tenha terminado e, para isso, deve estar projetado para operar em tempo real. A técnica de programação, chamada “*status loop*”, que consulta periodicamente o relógio (*timer tick*) da UCP do roteador para executar procedimentos em tempo real, foi implementada no mecanismo. A Figura 3.3 mostra o laço de execução da programação em tempo real. A coleta de tráfego, não representada na figura, poderia ser executada concorrentemente.

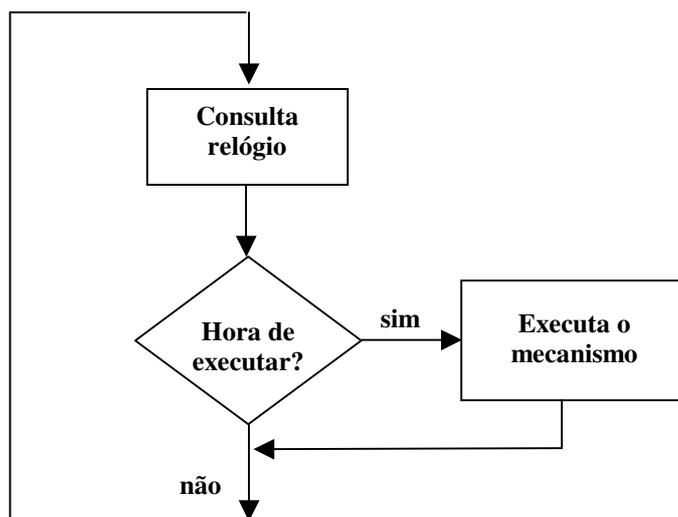


Figura 3.3. Diagrama do laço de execução do mecanismo ProCon

A arquitetura do mecanismo de prognóstico proposto está ilustrada na Figura 3.4 e compreende os procedimentos de:

- Caracterização do tráfego,
- Transformação do tráfego,
- Filtragem dos coeficientes,
- Identificação da tendência e
- Prognóstico de congestionamento.

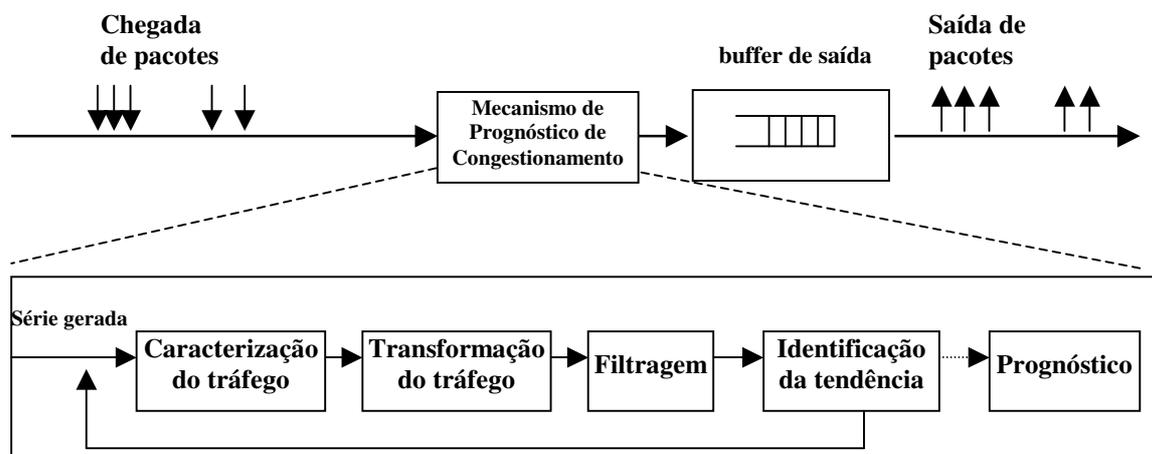


Figura 3.4. Mecanismo de prognóstico de congestionamento

A seguir são descritos os procedimentos ou fases do mecanismo de prognóstico de congestionamento proposto.

Primeiro procedimento: caracterização do tráfego

Tem o objetivo de identificar o grau de auto-similaridade do tráfego, através da utilização da função de autocorrelação, da análise da estatística R/S e do parâmetro de Hurst. A cada intervalo de tempo T a série obtida pelo processo de contagem é submetida a este procedimento. Este procedimento inicial é importante para o mecanismo porque existem situações, por exemplo a transmissão de vídeo ponto-a-ponto, onde o tráfego é dominado pelo protocolo UDP e a transmissão se faz de forma contínua, sem picos de rajadas de fluxo. Como foi visto no Capítulo 2, os fluxos UDP não diminuem a taxa de transmissão de dados em situações de congestionamento, a não ser que as aplicações o façam (BRADEN et al, 1998). Neste caso, se o tráfego gerado não tiver as características de auto-similaridade, os demais procedimentos, como transformação e filtragem, não são necessários e somente o último é executado.

Durante essa fase o mecanismo calcula as seguintes estatísticas sobre a série original: (i) o valor mínimo, o valor máximo e a amplitude, (ii) a média aritmética, (iii) a variância e o desvio padrão e (iv) a estatística de R/S e o parâmetro de Hurst. Estas mesmas estatísticas são também calculadas sobre a série transformada e reconstruída, após o procedimento de identificação de tendência.

Segundo procedimento: transformação do tráfego

Consiste em aplicar à série obtida, que corresponde ao tráfego real em um intervalo de tempo T , uma função de *wavelet* discreta, isto é, efetuar a transformação direta da série. A motivação principal na utilização de *wavelet* é a sua capacidade em reduzir a correlação temporal dos seus coeficientes. Os coeficientes obtidos pela transformada apresentam uma correlação menor que a correlação dos processos temporais correspondentes, sendo, portanto, dependentes de curta duração. As propriedades estatísticas dos coeficientes de *wavelet* podem ser usadas para caracterizar o processo estocástico original. Desta forma, uma série temporal com dependência de curta ou de longa duração pode, após a sua transformação, ser modelada por modelos matemáticos adequados para séries fracamente correlacionadas.

Para realizar este procedimento, o mecanismo faz no início da sua execução uma estimação da base a ser empregada durante o seu funcionamento, usando para isso a primeira janela de observação e as 20 bases de *wavelet* relacionadas no apêndice A. A estimação da base consiste em: (i) calcular a transformada direta de *wavelet*, (ii) efetuar a filtragem dos coeficientes (descrita no terceiro procedimento), (iii) calcular a transformada inversa e (iv) comparar o valor observado (original) com o correspondente valor calculado (reconstruído), computando-se a diferença quadrática média que é a função usada como critério de escolha da base. A base escolhida, dentre as 20, é a que apresentar menor diferença quadrática média. Eventualmente, o ProCon pode efetuar novas estimações da base em função das modificações que porventura ocorram nas características do tráfego.

Terceiro procedimento: filtragem dos coeficientes

A partir dos coeficientes de *wavelet* o passo seguinte consiste em aplicar filtros para zerar ou encolher os coeficientes menores do que um determinado limiar. A escolha da política e do limiar, descrita na seção seguinte, é importante para o mecanismo, pois o número de coeficientes zerados ou encolhidos influirá ou não na suavização da série reconstruída. O limiar deve ser calibrado de maneira que a série reconstruída não apresente valores próximos da série original, mas também não apresente valores muito distantes. Ressalte-se que o mecanismo de prognóstico é configurável, podendo trabalhar com diferentes políticas e tipos de limiares.

Realizada a filtragem, o procedimento efetua a transformada de *wavelet* inversa de *wavelet* usando a mesma base utilizada na transformação direta.

Quarto procedimento: identificação da tendência

Consiste em identificar a tendência crescente ou decrescente na série transformada, através dos testes de Cox-Stuart ou de Spearman (MORETTIN; TOLOI, 1987). Em caso de tendência crescente, é possível identificar um provável congestionamento comparando os elementos da série (número de pacotes) situados no “topo” da curva, isto é, os de maior valor, com um parâmetro de indicação de congestionamento (IC). O parâmetro IC pode ser a média amostral do tráfego ou então informado pelo usuário do mecanismo ProCon. Revelada a tendência, o mecanismo pode efetuar prognóstico do comportamento futuro do tráfego.

Quinto procedimento: prognóstico do tráfego

Consiste de critérios, descritos na Seção 3.4, para prognosticar possíveis congestionamentos, com base nos cálculos efetuados no procedimento anterior. Nesta fase o mecanismo poderia emitir uma notificação explícita para o controle de congestionamento do roteador e também prever o comportamento do tráfego para os próximos $h = 1, 2, \dots$, intervalos de tempo. Contudo, a previsão não é necessária, pois a indicação de tendência e o prognóstico de congestionamento, por si só, são suficientes para os objetivos desta tese. Por conseguinte, apesar da previsão do tráfego ser possível devido ao comportamento suave da série obtida pelos procedimentos anteriores, esta sistemática não foi incluída na implementação do mecanismo proposto.

3.3 Parâmetros do Mecanismo de Prognóstico de Congestionamento

Durante sua execução o mecanismo utiliza os seguintes parâmetros definidos na chamada do programa: (i) a base de *wavelet* usada na transformação; (ii) a política de redução de ruído (ou de coeficientes); (iii) o método de escolha do limiar dos coeficientes (iv) o tamanho da janela de observação e (v) o indicador de congestionamento (IC) ou limiar de congestionamento.

3.3.1 Base de *Wavelet*

A base *wavelet* é um dos parâmetros de entrada do mecanismo que pode ser estimado pelo usuário do ProCon. Contudo, se este parâmetro não for informado, o próprio mecanismo no início da sua execução estima a base, através da comparação do tráfego contido na primeira janela de observação com a sua respectiva série transformada e reconstruída. Dessa forma, a estimação recai sobre a base de menor valor de desvio, isto é, cuja soma dos desvios quadráticos de ajustamento (método dos mínimos quadrados) seja menor. As bases utilizadas

pelo programa são AdelsonSimoncelliHingorani (ADELSON; SIMONCELLI; HINGORANI, 1987), AntoniniBarlaudMatheiuDaubechies_4 (ANTONINI et al., 1992), BattleLemarie (MALLAT, 1989), BurtAdelson (DAUBECHIES, 1992), Coiflet_2, Coiflet_4, Coiflet_6 (BEYLKIN; COIFMAN; ROKHLIN, 1991), Daubechies_4, Daubechies_6, Daubechies_8, Daubechies_10, Daubechies_12, Daubechies_20 (DAUBECHIES, 1992), Haar (DAUBECHIES, 1992), PseudoCoiflet_4_4 (REISSELL, 1993), Spline_2_2, Spline_2_4, Spline_3_3, Spline_3_5, Spline_3_7 (DAUBECHIES, 1992). Estas bases e seus coeficientes de filtros de passa-baixo e de passa-alto estão relacionadas no apêndice A.

3.3.2 Política de Redução dos Coeficientes

A aplicação de filtros sobre os coeficientes é uma das aplicações de *wavelet*. Consiste de uma técnica estatística (GOMES; VELHO; GOLDSTEIN, 1999) que se baseia no encolhimento dos coeficientes de *wavelets*, conhecida como *de-noising*, com o objetivo de reduzir ou mesmo remover o ruído presente num sinal ou série de tempo, através da diminuição da magnitude desses coeficientes.

Este procedimento, aplicado a uma série de tempo $\{x_t\}_{t=0}^{n-1}$ atua na forma de uma eventual redução ou encolhimento de cada um dos coeficientes $\{d_j^k\}$, $0 \leq j \leq n-1$, $0 \leq k \leq 2^j - 1$, de acordo com determinados critérios. Os coeficientes reduzidos são obtidos através da aplicação de limiares aos coeficientes originais. O procedimento de redução dos coeficientes de *wavelet* de um sinal consiste de:

- (i) aplicação da transformada de *wavelet* discreta aos dados x_0, x_1, \dots, x_{n-1} , obtendo-se n (potência de 2) coeficientes d_j^k , todos com ruído;
- (ii) utilização de limiares (*thresholds*) para diminuir os coeficientes, tornando nulos aqueles abaixo de um valor especificado, obtendo-se nesta etapa os coeficientes sem ruído, \tilde{d}_j^k ;
- (iii) aplicação da transformada inversa de *wavelet* sobre os coeficientes da etapa anterior para obter as estimativas do sinal.

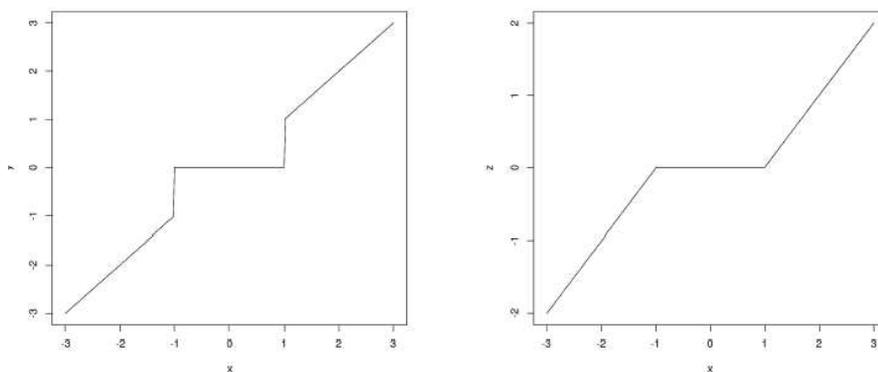
Dois políticas difundidas na literatura sobre redução de coeficientes são denominadas, respectivamente, de *dura* (*hard threshold*) e *suave* (*soft threshold*). Uma vez definido um dado limiar λ de redução, as políticas dura e suave podem ser expressas, respectivamente, por

$$\tilde{d}_j^k = \begin{cases} 0, & \text{se } |d_j^k| \leq \lambda, \\ d_j^k, & \text{se } |d_j^k| > \lambda. \end{cases}$$

e

$$\tilde{d}_j^k = \begin{cases} 0, & \text{se } |d_j^k| \leq \lambda, \\ \text{sign}(d_j^k)(|d_j^k| - \lambda), & \text{se } |d_j^k| > \lambda. \end{cases}$$

Estas políticas estão ilustradas nas Figuras 3.5a e 3.5b. A escolha da política está relacionada a um *trade-off* entre a variância e o viés associados ao processo de redução de ruído em questão. Para o mecanismo de prognóstico de congestionamento a política padrão é a *hard* (ver Seção 4.2 do Capítulo 4).



Figuras 3.5a e 3.5b. Política dura e suave com limiar $\lambda = 1$ para a $f(x) = x$

3.3.3 Método de Escolha do Limiar de Redução dos Coeficientes

A estimação da política de redução e do método de escolha do limiar são de extrema relevância para o mecanismo. Existem dois métodos, denominados, respectivamente, de “quantil” e “universal”, para os cálculos dos limiares que governam as políticas de redução.

O método “quantil” consiste em determinar um parâmetro λ que é um quantil do conjunto de todos coeficientes de *wavelet* (VIDAKOVIC; MÜLLER, 1998). Por exemplo, deseja-se substituir os 30% menores coeficientes (em valores absolutos) por zero. Então o limiar λ corresponde ao maior valor dos coeficientes que estão dentro da percentagem de 30%. Este método exige mais processamento devido à necessidade que se tem de ordenar toda a série por ordem crescente, para identificar os valores que estão abaixo ou acima do limiar escolhido. Por esta razão este método é pouco utilizado. Donoho et al. (1995) propuseram um

estimador, chamado “universal”, para o parâmetro λ que consiste no estabelecimento de um único λ relacionado com o desvio padrão dos coeficientes. Para estimar o parâmetro de limiaridade é usada a fórmula

$$\lambda = \sigma \sqrt{2 \log(n)},$$

onde n é o número de coeficientes e σ o desvio padrão. Este parâmetro vem como padrão nas principais ferramentas existentes que utilizam transformadas de *wavelet*, entre elas, o S-PLUS (vide <http://www.mathsoft.com>) e o MATLAB (vide <http://www.mathworks.com>). Nesta tese este parâmetro é chamado de “universal-2” e é o método de escolha padrão do limiar para o mecanismo de prognóstico de congestionamento.

Em Donoho et al. (1995) também é usada uma variante do limiar “universal-2”, cuja fórmula é

$$\lambda = \sigma \sqrt{\frac{2 \log(n)}{n}}.$$

As propriedades deste parâmetro de limiar são análogas às do “universal-2”. Nesta tese, este parâmetro é chamado de “universal-1”.

3.3.4 Janela de Observação

A janela de observação é constituída pelas observações que estão sendo processadas em determinado instante pelo mecanismo. A série de observações contida em uma janela é dinâmica porque à medida que a próxima observação ou coleta (número de pacotes no intervalo $[0,t]$) se torna disponível, a observação mais antiga é desprezada e a mais nova incorporada à série.

O tamanho da janela de observação deve ser potência de 2, isto é, a série contida em uma janela deve ser de tamanho 2^n , onde $n = 1, 2, \dots$. O tamanho padrão da janela de observação, assumido pelo mecanismo, é 128.

3.3.5 Indicador de Congestionamento

O indicador de congestionamento (IC) corresponde ao patamar (número de pacotes) a partir do qual considera-se o tráfego congestionado. O parâmetro IC pode ser a média amostral do tráfego ou então um valor informado pelo usuário do mecanismo ProCon. O valor padrão de

indicação de congestionamento, assumido pelo mecanismo, é a média aritmética dos valores da 1ª janela.

3.3.6 Resumo dos Parâmetros de Entrada do Mecanismo ProCon

Resumindo, os parâmetros são:

- (i) **Base de *wavelet*.** Apesar do mecanismo estimar a base a ser empregada durante o seu funcionamento, este parâmetro pode ser determinado pelo usuário ProCon;
- (ii) **Janela de execução.** Tamanho da janela padrão é 128 (observações);
- (iii) **Política de filtragem.** Política padrão é a *hard*;
- (iv) **Método de escolha do limiar dos coeficientes de *wavelet*.** Este limiar pode ser escolhido dentre os três métodos descritos na Seção 3.2. O método padrão é o “universal-2”;
- (v) **Indicador de congestionamento (IC).** O valor padrão é a média dos valores observados na 1ª janela de execução.

3.4 Especificação do Indicador de Tendência

Como foi visto na Seção 2.5 do Capítulo 2, a identificação de tendência na série de tráfego pode ser realizada através dos testes de Cox-Stuart e de Spearman. O primeiro teste compara os pares de observações $(x_1, x_{1+c}), (x_2, x_{2+c}), \dots, (x_{n-c}, x_n)$, onde n é o número de observações e $c = n/2$. Este teste rejeita, ou não, a existência de tendência baseado em uma distribuição normal com parâmetros $\mu = n/2$ e $\sigma^2 = n/4$, para um dado nível de significância α (CONOVER, 1971). O segundo teste, baseado no coeficiente de correlação de Spearman, consiste na identificação do posto “natural” de cada x_t e do posto após a classificação dos x_t em ordem ascendente, dentre as n observações. A existência de tendência ou não, neste teste, é dada pelo valor do coeficiente calculado pela fórmula de Spearman (Seção 2.5 do Capítulo 2). Este último teste exige mais processamento em relação ao teste anterior pela necessidade de ordenação dos x_t . Nesta tese utilizou-se apenas o teste de Cox-Stuart.

Ambos os testes verificam somente a tendência da série, mas a identificação de um provável congestionamento é feita através da comparação dos elementos que estão situados no trecho de tendência crescente com um valor (número de pacotes por unidade de tempo) determinado como sendo o limite, a partir do qual é considerado um caso de

congestionamento. Este valor limite, citado na Seção 3.3 como indicador de congestionamento (IC), pode ser determinado de duas formas: (i) através de uma distribuição normal constituída pelos valores da série que estão ao redor de um trecho com tendência crescente, para um dado nível de significância α . Neste caso, o IC corresponde à média aritmética dos valores ao redor do trecho com tendência crescente; ou (ii) através da experiência do usuário do mecanismo ProCon. Caso não seja especificado o IC por uma dessas formas, o ProCon assume, automaticamente, a média aritmética da 1ª janela de observação como valor limite.

O mecanismo ProCon detecta um possível congestionamento da seguinte forma:

- (i) Divide a janela de observação reconstruída (após os três primeiros procedimentos do mecanismo) em duas partes iguais;
- (ii) Compara as observações da primeira metade com os da segunda metade da janela, registrando a quantidade de valores da segunda metade que supera os valores da primeira metade. Esta comparação é feita em pares de observações, isto é, a observação x_1 é comparada com a x_{1+c} , a x_2 com x_{2+c} , e assim por diante, onde $c = n/2$ e n é o número de observações da janela. Esta comparação também é realizada entre as observações da primeira metade da janela que está sendo observada e as observações da segunda metade da janela anterior;
- (iii) O mecanismo considera um possível caso de congestionamento quando 50% dos valores da segunda metade são maiores que os da primeira metade. Este percentual é usado quando o IC é igual à média aritmética da primeira janela. Para outros valores de IC, um provável caso de congestionamento é dado por

$$\frac{m}{n/2} \geq \frac{0.5\mu}{IC},$$

onde n é o número de observações da janela, m é o número de pares observados cujo segundo termo é maior que o primeiro e μ é a média aritmética das observações da 1ª janela.

Se o mecanismo contivesse o procedimento de previsão de tráfego, a indicação de congestionamento no canal ou em portas do equipamento monitorado poderia ser inferida através da comparação do número de pacotes previsto, n_p , com valores limites que balizam o

fluxo de tráfego, isto é, os valores mínimo, m , e máximo, M , de pacotes por unidade de tempo. Quando $n_p < m$ o sistema funciona sem indicação de congestionamento. Se $m \leq n_p < M$ o mecanismo proposto informa ao controle de congestionamento do equipamento a possibilidade de ocorrência de engarrafamento do tráfego. A caracterização de um fato concreto de congestionamento ocorre quando $n_p \geq M$. Neste caso, o mecanismo alerta ao controle sobre o acontecimento.

3.5 Descrição do Tráfego Simulado

Realizar mensurações na rede de variáveis como retardo, vazão e descarte de pacotes para cada nó ou cada enlace de uma rede WAN, por exemplo, é uma tarefa, muitas vezes, impossível de ser executada. Para contornar este problema, pode-se realizar medições fim-a-fim, isto é, mensurações em pontos finais da rede, através de *softwares* coletores de tráfego, tais como o MRTG (<http://www.mrtg.org>), Analyzer (<http://analyzer.polipo.it>), MG-SOFT MIB Browser (<http://www.mg-soft.si>), Tivoli (<http://www.ibm.com/br/products>) e Log Analyzer (<http://www.netif.com/products>). Além dessa alternativa, é possível obter-se (baixar) arquivos com tráfego real, disponíveis na Internet, ou então utilizar programas que geram tráfego que simulam a realidade.

Dentre as alternativas, optou-se pela utilização de tráfego simulado porque permite ser manuseado com mais facilidade, obtendo-se uma gama maior de testes. Introduzir modificações, por exemplo acréscimos em parte dos valores gerados pela simulação que espelhem situações de congestionamento, é uma vantagem do tráfego simulado.

Para fins de teste do mecanismo de prognóstico de congestionamento, foi desenvolvido um programa que gera tráfego simulado a partir de um modelo de tráfego genérico, conhecido como IRP (*Interrupted Renew Process*), proposto por Huang (2000) ao grupo de trabalho do IEEE 802.16. Este modelo é uma generalização do modelo 4IPP (*Sources Four - Interrupted Poisson Process*) (BAUGH, 2000) usado para simular tráfego auto-similar de aplicações em redes sem fio de banda larga, mas também serve para simular tráfego de voz, vídeo e dados de redes WAN (BAUGH et al., 2001).

O modelo IRP está baseado em processos ON/OFF, descritos no Capítulo 2, onde o estado ON representa um surto de tráfego, explosivo ou não, e o estado OFF representa a inatividade ou a ausência de tráfego. A Figura 3.6 ilustra este modelo.

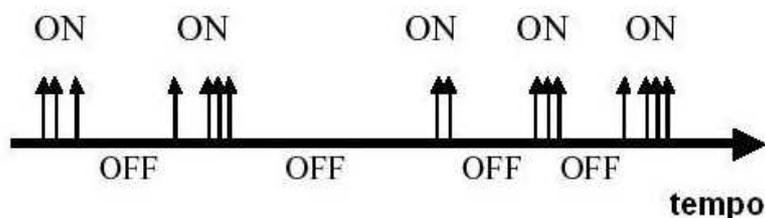


Figura 3.6. Processo ON/OFF

O mecanismo (ilustrado na Figura 3.7) que faz esta simulação dentro do programa simulador de tráfego é implementado da seguinte forma:

- (i) O tempo no estado ON, T_{ON} , é calculado por uma distribuição de Pareto, com parâmetros α_1 e β_1 . Este tempo pode ser maior, menor ou igual à unidade de tempo especificada no processo de contagem do tráfego (segundos ou milissegundos).
- (ii) O número de observações (pacotes) durante o período ON é computado a partir dos intervalos de tempo exponenciais entre chegadas sucessivas de pacotes. Cada intervalo de tempo exponencial é subtraído do tempo no estado ON até $T_{ON} \leq 0$.
- (iii) O tempo no estado OFF também é calculado por uma distribuição de Pareto, com parâmetros α_2 e β_2 .

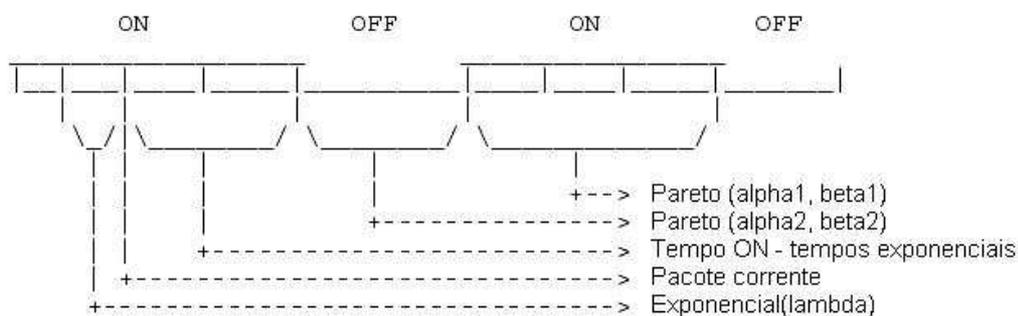


Figura 3.7. Descrição do mecanismo simulador

O programa simulador de tráfego pode ser configurado com seis parâmetros, a seguir descritos:

- (i) **Duração.** É o tamanho do arquivo que se deseja gerar, isto é, o número de observações contidas no tráfego gerado.
- (ii) **Tipo de congestionamento.** Corresponde aos acréscimos que se deseja introduzir nos valores da série gerada com o objetivo de simular congestionamento. As opções para este parâmetro são: sem congestionamento, congestionamento linear, congestionamento logarítmico e aleatório. No congestionamento linear é acrescentado um percentual constante, informado ao programa, sobre o valor originalmente gerado. No congestionamento logaritmo utiliza-se a fórmula recursiva $x_i = x_i(\ln(\log(x_i)))$ para calcular o percentual de acréscimo. Por fim, no congestionamento aleatório o percentual é aleatório no intervalo (1, 2).
- (iii) **Tamanho da janela de congestionamento.** O valor padrão deste parâmetro é 64.
- (iv) **Posição da janela de congestionamento.** Pode ser fixa ou aleatória. Se for fixa, a posição de início da janela deve ser informada ao programa. Caso seja aleatória, o programa escolhe, aleatoriamente, qualquer posição dentro dos valores gerados de tráfego.
- (v) **Tamanho do pacote.** Pode ser fixo ou aleatório. Caso seja fixo, o valor deve ser informado. Para tamanho aleatório o programa escolhe aleatoriamente valores entre 64 e 1518 bytes, inclusive, que correspondem aos tamanhos mínimo e máximo de um pacote TCP.

3.6 Descrição dos Programas do Mecanismo ProCon

O ProCon, desenvolvido na linguagem C/C++, é constituído de rotinas (funções) que executam as seguintes tarefas:

- (i) Cálculo das estatísticas, tais como valores mínimo e máximo, amplitude, média aritmética, variância e desvio padrão, e identificação de auto-similaridade no tráfego, através do parâmetro de Hurst.
- (ii) Cálculo das transformadas de *wavelet* direta e inversa.
- (iii) Estimativa da base de *wavelet* mais adequada para o cálculo do item anterior.

- (iv) Redução ou encolhimento dos coeficientes de *wavelet* a partir de políticas de filtragem e de valores de limiars estimados.
- (v) Identificação das tendências da série e prognóstico de comportamentos futuros do tráfego.

O Quadro 3.1 relaciona as funções do sistema ProCon que constitui a implementação do mecanismo de prognóstico de congestionamento.

Quadro 3.1. Funções do Sistema ProCon

Função	Descrição	Parâmetros de entrada
<i>Estatistic()</i>	Cálculo da estatística R/S, parâmetro de Hurst, valores mínimo, máximo, amplitude, média aritmética, variância e desvio padrão	Série de entrada Janela de observação
<i>Doit()</i> + <i>wvlt.lib</i>	Cálculo da transformada direta e inversa de <i>wavelet</i>	Série de entrada Janela de observação Base de <i>wavelet</i> Tipo de transformada Tipo de arredondamento ¹
<i>Estimate_base()</i>	Estimação da base de <i>wavelet</i> mais adequada para o cálculo do item anterior	Série de entrada Janela de observação Base escolhida
<i>Compute_thresh()</i> + <i>Apply_thresh()</i> + <i>Quicksort.h</i>	Redução (filtragem) dos coeficientes de <i>wavelet</i>	Coefficientes Política de redução Tipo de limiar Percentual de redução
<i>Tend()</i>	Verificação da tendência da série transformada	Série de entrada Janela de observação Indicador de congestionamento Tipo de teste de tendência

Também foram desenvolvidos programas que auxiliam os testes do mecanismo e as análises realizadas sobre os resultados. Estes programas são: (i) gerador de tráfego simulado auto-similar, descrito na Seção 3.5, (ii) cálculo das autocorrelações de uma série até o *lag*

¹ A transformação de wavelet pode gerar valores não inteiros. O ProCon arredonda os valores da série reconstruída para o inteiro mais próximo.

100, (iii) cálculo dos estatísticos de uma série temporal, (iv) cálculo da transformada de *wavelet* usando o esquema Lifting (SWELDENS, 1997) - este esquema não foi utilizado na construção do ProCon, (v) gerador de números sequenciais para obtenção de gráficos, (vi) cálculo de previsão de séries temporais com os métodos de alisamento exponencial adaptativo e de filtragem adaptativa (MORETTIN; TOLOI, 1987). Este último programa foi elaborado no início do desenvolvimento do ProCon, quando se pretendia incluir previsão de valores futuros no mecanismo.

Todos os programas desenvolvidos para esta tese foram compilados pela versão 5.5 do compilador C++ *Builder* da Borland (IMPRISE, 2000). Os gráficos que ilustram os resultados do mecanismo (ver Capítulo 4) foram obtidos através do *software* GNUPLOT, versão 3.8i, de domínio público (<http://www.gnuplot.org>).

O programa ProCon tem cinco funções, a seguir descritas:

- (i) ***Estatistic***. Função de cálculo dos estatísticos da série ou da janela de observação.
- (ii) ***Doit***. Função de cálculo da transformada de *wavelet*. Esta função utiliza uma biblioteca de rotinas de domínio público, escritas para a linguagem C e conhecidas como *wvlt.lib* (LEWIS, 1995), da Universidade British Columbia (UBC). Essa biblioteca contém rotinas que executam operações básicas com *wavelets*, como as transformações direta e inversa.
- (iii) ***Estimate_base***. Função que estima a base de *wavelet*, escolhendo uma com menor valor de desvio quadrático médio.
- (iv) ***Compute_thresh***. Função de cálculo do valor do limiar a partir dos parâmetros especificados, descritos na Seção 3.3. Os coeficientes abaixo deste limiar são zerados ou reduzidos. Caso o limiar escolhido seja do tipo “quantil”, esta função, antes de realizar o cálculo do limiar, ordena a série através da função *Quicksort*.
- (v) ***Tend***. esta função está descrita na Seção 3.4.

Além dessas funções o ProCon também realiza:

- (i) Simulação da aplicação em tempo real, onde as funções são executadas em intervalos fixos de tempo.
- (ii) Gravação dos coeficientes obtidos pelos procedimentos de transformação e de filtragem.

- (iii) Registro do tempo que o programa leva para efetuar um ciclo completo de execução.
- (iv) Gravação do patamar de congestionamento para ilustrar os gráficos de resultados.

Um fluxograma das funções do ProCon está ilustrado na Figura 3.8.

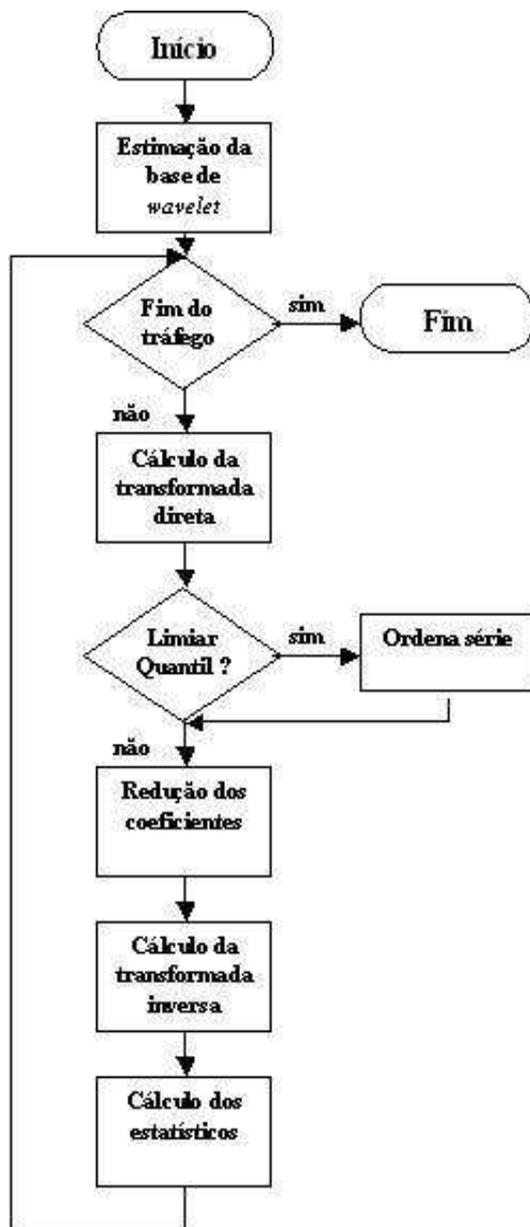


Figura 3.8. Fluxograma das funções do ProCon

Outros programas que não fazem parte do núcleo do ProCon, mas que foram desenvolvidos com o objetivo de auxiliar, comparar, otimizar e complementar funções do próprio mecanismo, estão relacionados no Quadro 3.2.

Quadro 3.2. Programas Auxiliares

Programa	Descrição	Parâmetros
<i>GenTraffic</i>	Simulador de tráfego auto-similar	Modelo de tráfego Duração (# observações) Largura de banda Congestionamento (tipo) Janela de congestionamento Posição da janela Tamanho do pacote Série de saída
<i>Autocor</i>	Cálculo de autocorrelações	Série de entrada Autocorrelações
<i>CalcEstat</i>	Cálculo dos estatísticos	Série de entrada Estatísticos
<i>Lifting</i>	Cálculo da transformada de <i>wavelet</i> usando o esquema <i>Lifting</i> e a base Haar	Série de entrada Série de saída Tipo de transformação
<i>ALExpAdapt</i>	Previsão com o método de alisamento exponencial adaptativo de Trigg & Leach	Série de entrada Série de saída
<i>FilAdapt</i>	Previsão com o método de filtragem adaptativa	Série de entrada Série de saída

A seguir, complementam-se as descrições dos programas auxiliares:

- (i) o programa *GenTraffic* está descrito na Seção 3.5.
- (ii) o programa *Autocor* auxilia na análise dos resultados obtidos, calculando as autocorrelações de uma série até o lag 100.
- (iii) O *CalcEstat* calcula os estatísticos de uma série temporal, tais como mínimo, máximo, amplitude, média aritmética, variância, desvio padrão, coeficiente de variação, R/S e o parâmetro de Hurst.
- (iv) o programa *Lifting* usa o esquema de mesmo nome, descrito no Capítulo 2. Esta técnica favorece a geração de um algoritmo de menor complexidade em relação aos algoritmos baseados na estrutura clássica de domínio da frequência, com um custo

de computação proporcional ao número de observações da série (SWELDENS, 1997), isto é, a transformação linear de um vetor de tamanho N requer $O(N^2)$ operações.

- (v) o programa *AIExpAdapt* utiliza o método AEA (Alisamento Exponencial Adaptativo), derivado do método AES (Alisamento Exponencial Simples) (MORETTIN; TOLOI, 1985), para previsão de valores futuros.
- (vi) o programa *FilAdapt* usa o método FA (Filtragem Adaptativa) para previsão de valores futuros, que pode ser aplicado em séries não estacionárias, permitindo adaptações a possíveis mudanças no padrão da série.

3.7 Cenário de Execução do Mecanismo ProCon

A execução do mecanismo de prognóstico é via linha de comando, digitando-se a palavra “procon” e em seguida os argumentos (parâmetros) do programa. Para isso é necessário abrir uma janela DOS no ambiente Windows. Os argumentos do programa, apresentados na Figura 3.9, podem ser vistos digitando-se “procon -?” na linha de comando.

```

C:\>procon [<arg>] [<nome da serie de entrada>] [<nome serie de saida>]
           [<nome do arquivo de estatisticas>]

Os argumentos sao:
-b <w>    usa base de wavelet selecionada pelo pela opcao <w>. Se o argumento
          nao for especificado, o programa seleciona a base.
-c        grava um arquivo contendo os coeficientes de wavelet.
-p <p>    usa uma politica selecionada pela opcao <p>, onde p = hard ou soft.
-t <t>    usa um limiar selecionado pela opcao <t>, onde t = universal ou quan-
          til. Se quantil pede-se o percentual de observacoes (default 50%).
-j <j>    usa o tamanho de janela especificada por <j> (default 128).
-m <m>    usa o patamar de congestionamento especificado por <m>.
-?       opcao de ajuda para listar os argumentos do programa.

Bases includidas no programa:
AdelsonSimoncelliHingorani      AntoniniBarlaudMatheiuDaubechies_4_4
BattleLemarie                    BurtAdelson                      Coiflet_2                        Coiflet_4
Coiflet_6                        Daubechies_4                     Daubechies_6                   Daubechies_8
Daubechies_10                   Daubechies_12                   Daubechies_20                  Haar
Pseudocoiflet_4_4              Spline_2_2                      Spline_2_4                     Spline_3_3
Spline_3_5                      Spline_3_7

OBS: O programa aceita qualquer palavra abreviada, por exemplo, daub*4

C:\CURSOC~1>

```

Figura 3.9. Tela de execução do programa ProCon

O ProCon foi executado em três computadores com processadores distintos, um Pentium MMX de 233 MHz e 96 MB de RAM, um Celeron de 995 MHz e 256 MB de RAM e um

Pentium IV de 2,2 GHz e 256 MB de RAM, com objetivo de mensurar os tempos que o programa demandaria para executar os prognósticos de congestionamento (ver Quadro 4.2 do Capítulo 4).

Para validar o mecanismo foram geradas trinta séries com 1024 e 2048 observações, representando tráfegos simulados com dependência de longa duração. Os arquivos gerados, contendo as séries, apresentam um *layout*, como o ilustrado no Quadro 3.3, onde cada linha corresponde a uma observação em uma unidade de tempo. Na primeira coluna do quadro citado tem-se o tempo registrado da observação, em uma unidade de tempo que pode ser segundo ou milissegundo, e na segunda a quantidade de pacotes por unidade de tempo (ut). O número de observações das séries geradas é o mesmo número de unidades de tempo.

Quadro 3.3. Amostra do arquivo gerado

Tempo	Tráfego (pacotes/ut)
07:47,3	220
07:47,4	150
07:47,5	140

As bases escolhidas para validar o mecanismo ProCon foram selecionadas pelo próprio mecanismo, baseado na menor e na maior soma quadrática de desvios, ou seja, na soma das diferenças entre o valor observado do tráfego e o valor modelado pela transformada de wavelet. Para cada uma destas bases optou-se por janelas de observação de tamanho 128 e 512, a fim de mostrar a influência do tamanho da janela na modelagem do tráfego.

Com relação às políticas de redução de ruído, utilizou-se tanto a denominada “dura” (*hard*) quanto a “suave” (*soft*), com predominância pela primeira, pois a segunda poderia apresentar vícios grandes, enquanto que a política dura tem vício menor mas variância maior (MORETTIN,1997). Os métodos selecionados para escolha dos limiares de redução dos coeficientes foram os citados na Seção 3.3. A maioria das validações do mecanismo foi realizada com o método “universal-2”, pelos seguintes motivos: (i) o método “quantil” exige mais processamento, demandando mais tempo e (ii) o método “universal-1” não gera o nível de suavização da série transformada requerido pelo mecanismo. O Quadro 3.4 mostra os parâmetros escolhidos para validar o mecanismo ProCon. Este quadro apresenta: (i) na primeira coluna, o número de observações da janela de observação, (ii) na segunda, as bases

de *wavelet* escolhidas, (iii) nas duas colunas seguintes as políticas de filtragem (redução) dos coeficientes e (iv) nas três últimas os métodos utilizados para cálculo dos limiares.

Quadro 3.4. Bases de *wavelet*, políticas de filtragem e métodos de limiar

Número de observações da janela	Base de <i>wavelet</i>	Política de redução		Método de escolha do limiar		
		hard	soft	universal-1	universal-2	quantil
128	Base com menor soma de desvios	X			X	
	Base com maior soma de desvios		X	X		
512	Base com menor soma de desvios		X		X	
	Base com maior soma de desvios	X				X

3.8 Comentários

Este capítulo expôs no seu início a descrição do posicionamento do ProCon dentro de nodos intermediários, como em roteadores, por exemplo. A seção seguinte descreveu a arquitetura, funções e procedimentos executados pelo mecanismo. As especificações dos parâmetros usados pelo mecanismo de prognóstico (bases de *wavelet*, políticas de redução de ruído, tipos de limiares de coeficientes, percentuais de redução e indicadores de tendências de congestionamento) foram discutidas na Seção 3.3. Na Seção 3.4 é feita uma descrição de como a tendência do tráfego é inferida. A Seção 3.5 apresentou a simulação do tráfego utilizado nos testes do mecanismo de prognóstico. Os programas que compõem o ProCon e suas funções foram descritos na Seção 3.6. As duas últimas seções descreveram, respectivamente, os cenários de execução e os comentários finais.

Capítulo 4

Resultados e Discussão

Este capítulo apresenta os resultados obtidos e sua análise. O capítulo inicia com a escolha dos parâmetros do mecanismo. A Seção 4.2 descreve os resultados obtidos utilizando os parâmetros considerados padrões pelo mecanismo proposto. A variação dos parâmetros para valores diferentes do padrão não apresentou resultados satisfatórios, como pode ser visto na Seção 4.3. Os parâmetros padrões foram estabelecidos através de um projeto fatorial tipo 2^{k-1} e são os que mais favorecem para a exatidão dos prognósticos de tendência de congestionamento. A Seção 4.3 faz uma análise do mecanismo após a variação dos parâmetros: (i) base de *wavelet*, (ii) tamanho da janela, (iii) política de filtragem, (iv) método de escolha do limiar dos coeficientes e (v) limiar de congestionamento. A Seção 4.4 analisa os tempos de processamento para o mecanismo realizar prognósticos de congestionamento. A validação do ProCon é apresentada na Seção 4.5. A última Seção, 4.6, apresenta os comentários finais.

4.1 A Escolha dos Parâmetros do Mecanismo

Os parâmetros selecionados para o mecanismo influem na modelagem do tráfego. Para otimizar a escolha dos parâmetros do ProCon, foi adotado um projeto experimental (JAIN, 1991) de maneira que possa permitir:

- (i) A realização de um conjunto de experimentos para medição;
- (ii) O desenvolvimento de um modelo que melhor descreva os dados obtidos;
- (iii) Estimação da contribuição dos fatores tamanho de janela, base, política e limiar de redução de coeficientes em relação à corretude do prognóstico, e verificar se os mesmos são significativos.

A métrica utilizada na avaliação foi o percentual de acertos do prognóstico do mecanismo. O projeto experimental usado consiste de uma técnica fatorial do tipo 2^k , onde k é a quantidade de fatores. Para o desenvolvimento desta técnica foram escolhidos quatro fatores, onde cada fator possui dois níveis. Os fatores, os níveis escolhidos e os sinais que representam os efeitos dos fatores estão no Quadro 4.1.

Quadro 4.1. Fatores escolhidos

Fator	Parâmetro	Níveis	Sinais
A	Janela	64	-1
		128	1
B	Base	Menor SQD ²	-1
		Maior SQD	1
C	Política	Soft	-1
		Hard	1
D	Limiar	Universal1	-1
		Universal2	1

Cada experimento foi realizado oito vezes, ou seja, oito simulações com tráfegos diferentes. Assim, obteve-se um projeto fatorial $2^4 * 8$, o que significa que foram realizadas 128 observações (valores medidos). Com 4 fatores e dois níveis obteve-se 16 combinações a serem usadas para a medição dos percentuais de acertos.

Como a matriz de cálculo da técnica citada apresentou muitos dados (128 no total) para a análise, verificou-se que esta técnica podia ser substituída por uma outra, denominada fatorial fracionada 2^{k-p} , para reduzir a quantidade de experimentos. A técnica fatorial fracionada faz a fixação de um dos fatores, ou seja, ao invés de se realizar 16 experimentos, utiliza-se o fatorial 2^{4-1} que permite um planejamento com 8 combinações. Os fatores usados com esta técnica foram A, B, C, e D e as interações AB, AC, e BC. As demais interações entre os fatores não foram usadas, pois já estavam representadas.

A percentagem da variação da resposta explicada por cada fator é importante na decisão da escolha dos parâmetros do mecanismo, pois um fator pode ter um impacto significativo ou não. A variação devido aos erros experimentais também deve ser isolada. A variação total da resposta é conhecida como SST (*Sum of Squares Total*) e é calculada através da seguinte expressão:

$$SST = \sum_{i,j} (y_{ij} - \bar{y})^2 ,$$

onde y_{ij} representa os valores medidos correspondentes para cada um dos 16 experimentos, i representa o experimento e j o número da repetição (simulações), e \bar{y} é a média amostral para cada experimento.

A variação total da resposta (SST) encontrada foi de 5.815472 e é explicada por:

- O tamanho da janela (fator A) é responsável por 10,95% da variação.

² Menor ou maior SQD (Soma Quadrática de Desvios).

- A base de *wavelet* (fator B) é responsável apenas por 0,55% da variação.
- A política de filtragem (fator C) é responsável por 29,04% da variação.
- O limiar de redução (fator D) explica apenas 5,50% da variação.
- A interação dos fatores (AB) é responsável por 21,10% da variação.
- A interação dos fatores (AC) explica 2,16% da variação.
- A interação (BC) é responsável por apenas 1,11% da variação.
- O restante de 29,56% não explicado pode ser atribuído a grande variabilidade das séries de tráfego.

Com base nessas percentagens, os parâmetros escolhidos e considerados padrões para o mecanismo ProCon foram:

- (i) A política de filtragem *hard*, representado pelo nível 2 do fator C, tem a maior percentagem (29,04%);
- (ii) Tamanho da janela 128, representado pelo nível 2 do fator A (10,95%).
- (iii) O fator B (base de *wavelet*) não tem influência, pois explica 0,55% da variação, mas a interação com o fator A é significativa (21,10%). Neste caso, optou-se por deixar o próprio mecanismo estimar a base, baseado na menor soma quadrática de desvios.
- (iv) Por fim, o limiar de redução universal-2, representado pelo nível 2 do fator D.

4.2 Resultados Usando os Parâmetros Padrões do Mecanismo

Os resultados apresentados a seguir mostram o comportamento do mecanismo proposto usando os parâmetros escolhidos na seção anterior. A retirada do ruído das séries de tráfego simulado através do encolhimento dos coeficientes de *wavelet*, isto é, da diminuição da sua magnitude, é realizada conforme mencionado no Capítulo 3.

Como foi mencionado no capítulo citado, este encolhimento só é factível devido à autocorrelação dos coeficientes ser próxima de zero. O patamar de congestionamento, um outro parâmetro padrão considerado pelo mecanismo, é estabelecido como a média dos valores da 1ª janela de observação.

As bases de *wavelet* usadas para as transformações foram estimadas pelo próprio mecanismo, de acordo com critério descrito na Seção 3.3 do Capítulo 3. Nas janelas de observação com tamanho 128 e 512 são usadas as bases *Coiflet_4* e *Daubechies_10* para a

série com 1024 observações e as bases Daubechies_8 e Spline_2_2 para a série com 2048 observações. O Quadro 4.2 resume as bases usadas para teste do mecanismo.

Quadro 4.2. Bases escolhidas pelo mecanismo

Tamanho da janela de observação	Tráfego com 1024 observações	Tráfego com 2048 observações
	Bases de wavelet	
128	Coiflet_4	Daubechies_8
512	Daubechies_10	Spline_2_2

As autocorrelações dos tráfegos simulados e transformados e dos coeficientes foram comparadas através do parâmetro de Hurst e dos gráficos de autocorrelações. A Tabela 4.1 mostra os parâmetros de Hurst, calculados pela Eq. (15), tanto dos coeficientes quanto das séries (tráfego) simuladas e transformadas. As duas primeiras colunas desta tabela apresentam estes parâmetros para o tráfego com 1024 observações e as duas últimas para o tráfego com 2048. Observa-se que os parâmetros de Hurst das séries simuladas estão acima de 0,84 e das transformadas acima de 0,90, enquanto seus respectivos coeficientes apresentam valores menores que 0,56, indicando que a transformada de *wavelet* reduz a dependência temporal dos coeficientes. Portanto, os coeficientes obtidos pelas transformadas apresentam dependência de curta duração, o que favorece modificações nos seus valores sem alterar suas autocorrelações.

Tabela 4.1. Parâmetro de Hurst dos coeficientes e das séries simuladas e transformadas

Séries	Tráfego com 1024 observações		Tráfego com 2048 observações	
	Janela de 128	Janela de 512	Janela de 128	Janela de 512
Coeficientes	0.54	0.51	0.53	0.55
Simuladas	0.87	0.87	0.85	0.85
Transformadas	0.93	0.93	0.91	0.91

A Tabela 4.2 apresenta somente os estatísticos dos coeficientes obtidos pela transformação das séries simuladas com 1024 e 2048 observações. Os dados desta tabela mostram que: (i) os desvios padrões dos coeficientes são maiores que os correspondentes desvios das séries simuladas e transformadas (vide Tabela 4.3), sendo até três vezes maior que os desvios padrões da série com 2048 observações; (ii) a variabilidade dos coeficientes é

grande, como pode ser observado através do valor da amplitude, comparada com a das séries simuladas e transformadas (vide Tabelas 4.3 e 4.4); (iii) as médias aritméticas dos coeficientes são menores do que 5.20, sobressaindo uma próxima de zero (0.17), enquanto as médias das séries simuladas e transformadas estão acima de 36 (vide Tabela 4.3 e 4.4).

Tabela 4.2. Estatísticos dos coeficientes de *wavelets*

Estatísticos	Tráfego com 1024 observações		Tráfego com 2048 observações	
	Janela de 128 Coiflet_4	Janela de 512 Daubechies_10	Janela de 128 Daubechies_8	Janela de 512 Spline_2_2
Média	3.56	0.17	5.19	3.48
Variância	2375.49	2388.12	5387.16	6118.79
Desvio padrão	48.74	48.87	73.40	78.22
Coef. Variação	13.70	294.22	14.14	22.50
Mínimo	-197.78	-296.28	-252.76	-435.24
Máximo	698.98	1085.90	1159.66	1785.58
Amplitude	896.76	1382.18	1412.42	2220.82

Tabela 4.3. Estatísticos da série simulada com 1024 observações

Estatísticos	Série Simulada	Série transformada (Coiflet_4)	Série transformada (Daubechies_10)
Média	36.77	37.02	37.15
Variância	1036.21	440.38	384.51
Desvio padrão	32.19	20.99	19.61
Coef. de variação	0.88	0.57	0.53
Mínimo	0.00	0.00	0.00
Máximo	130.00	87.00	74.00
Amplitude	130.00	87.00	74.00
Parâmetro de Hurst	0.87	0.93	0.93

As Tabelas 4.3 e 4.4 apresentam os estatísticos das séries simuladas e transformadas (reconstruídas). Os dados destas tabelas mostram que: (i) as médias aritméticas das séries simuladas e transformadas estão muito próximas umas das outras; (ii) os desvios padrões das séries reconstruídas são menores que a correspondente medida das séries simuladas, indicando uma menor variabilidade das observações transformadas; (iii) as séries transformadas pelas bases de Daubechies_10 (Tabela 4.3) e Spline_2_2 (Tabela 4.4) apresentaram menor dispersão; (iv) todos os parâmetros de Hurst estão acima de 0,86, indicando auto-similaridade em todas as séries. Observa-se porém, que as transformações produziram séries com parâmetro de Hurst acima do parâmetro da série original.

Tabela 4.4. Estatísticos da série simulada com 2048 observações

Estatísticos	Série simulada	Série transformada (Daubechies_8)	Série transformada (Spline_2_2)
Média	62.88	62.76	62.87
Variância	1460.84	544.87	473.07
Desvio padrão	38.22	23.34	21.75
Coef. de variação	0.61	0.37	0.35
Mínimo	0.00	0.00	15.00
Máximo	136.00	117.00	115.00
Amplitude	136.00	117.00	100.00
Parâmetro de Hurst	0.85	0.91	0.91

A Figura 4.1 mostra as autocorrelações das séries transformada e simulada e dos coeficientes gerados a partir da base Coiflet_4. Observa-se que as autocorrelações das séries simulada e transformada decaem lentamente se comparadas com a autocorrelação da série dos coeficientes a qual está próxima de zero, confirmando-se que a transformada de *wavelet* obtém coeficientes com dependência de curta duração. As autocorrelações das demais transformadas apresentam o mesmo comportamento e por isso os gráficos são omitidos.

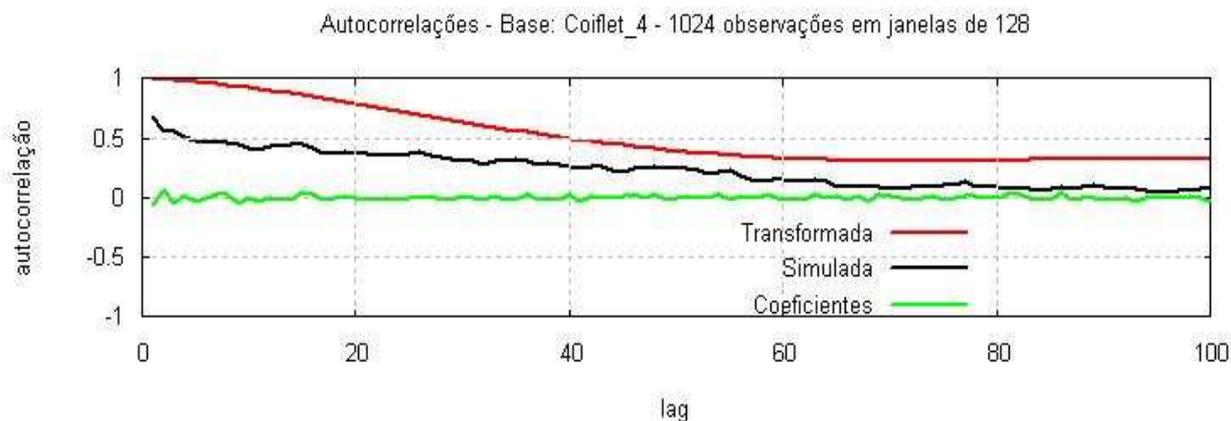


Figura 4.1. Autocorrelações

Com base nos dados das Tabelas 4.1, 4.2 e 4.3, obtidos através dos programas auxiliares descritos na Seção 3.6 do Capítulo 3 e na Figura 4.1, apresentam-se as seguintes análises:

- (i) Os coeficientes gerados pela transformada discreta de *wavelet* apresentam autocorrelação temporal menor que a do tráfego simulado, sendo, portanto,

dependente de curta duração. Estudos teóricos descritos no Capítulo 2 também inferem este resultado;

- (ii) As séries transformadas (reconstruídas) são dependentes de longa duração e apresentam variabilidade menor do que a das séries originais simuladas. Os dados obtidos empiricamente permitem esta conclusão;
- (iii) As transformações produziram séries que apresentaram parâmetros de Hurst maiores que os da série original (simulada). Este resultado foi obtido empiricamente, através de testes. Uma possível explicação está no encolhimento dos coeficientes gerados na transformação direta;
- (iv) Alterações nos valores dos coeficientes, realizadas através de filtros, reduzem o ruído presente no tráfego simulado. Isto pode ser constatado pela variabilidade das séries transformadas e constitui uma das principais aplicações da *wavelet* na Estatística, descrita no Capítulo 2.

A execução da transformada inversa, a partir dos coeficientes gerados pela transformada discreta de *wavelet*, constitui um dos procedimentos do mecanismo ProCon para prognosticar congestionamento. A série obtida pela transformada inversa, após o procedimento de filtragem, apresenta um comportamento suave, como pode ser visto no tráfego da Figura 4.2.

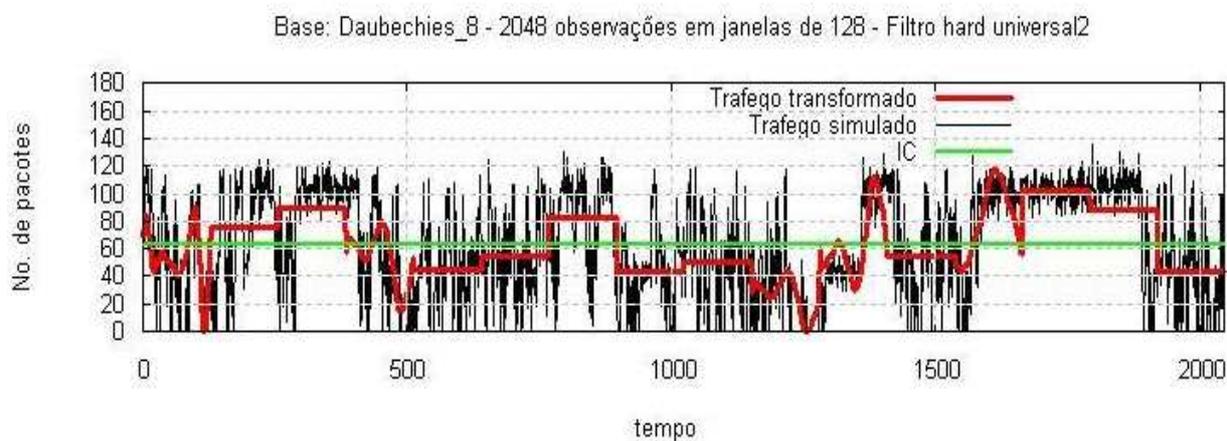


Figura 4.2. Série original e série transformada pela base de Daubechies_8

O gráfico da Figura 4.2 mostra:

- (i) o tráfego original simulado com 2048 observações (traço preto);
- (ii) a série transformada pela base de Daubechies_8 em janelas de observação de tamanhos 128 (traço vermelho); e

- (iii) o indicador de congestionamento, IC, representado pela média da 1ª janela de observação (traço verde).

A Figura 4.3 contém as mesmas séries da Figura 4.2, porém com mais detalhes, isto é, o gráfico mostra as primeiras 512 observações. Verifica-se na Figuras 4.2, e com mais detalhes na Figura 4.3 (vide setas), que a série transformada acompanha as inflexões da série original, indicando as tendências.

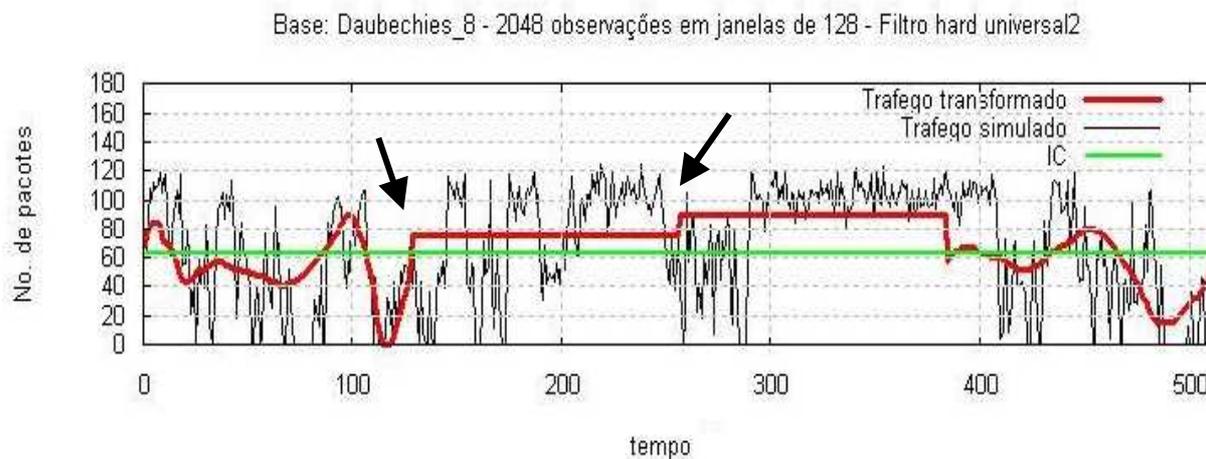


Figura 4.3. Séries original e transformada com as primeiras 512 observações

O mecanismo ProCon infere um possível congestionamento verificando a tendência da série gerada pela transformada inversa em cada janela de observação. Por exemplo, na série simulada com 2048 observações e janelas de tamanho 128, o mecanismo executou 16 vezes o teste de tendência, uma para cada janela de observação. A Figura 4.4 apresenta o resultado inferido pelo mecanismo após a aplicação do teste de tendência nas quatro primeiras janelas de observação da série citada como exemplo. Este resultado mostra possíveis casos de congestionamento na 2ª e na 3ª janela em função da tendência crescente da série transformada, nas observações 129 a 192 e 257 a 320. Confrontando esses resultados com o gráfico da Figura 4.3, observa-se que nesses intervalos existem realmente tendências crescentes, como estão indicados pelas setas. Além desses intervalos, observa-se também a existência de tendências crescentes nas observações próximas de 100 e no início da série, que não foram detectadas pelo mecanismo. Nota-se pelo gráfico, que os comportamentos nestes pontos, isto é, no início da série e na observação 100, são oscilantes, estando acima do patamar de congestionamento, mas nos instantes seguintes estão com valor zero. Estes tipos de comportamentos são interpretados pelo mecanismo como rajadas curtas e súbitas que

podem ser absorvidas pelo *buffer* do controle de congestionamento do roteador, sem contudo caracterizar congestionamento. Esta tendência poderia ser detectada se o tamanho da janela escolhida fosse 64, o que não é o caso deste exemplo cuja janela é de tamanho 128.

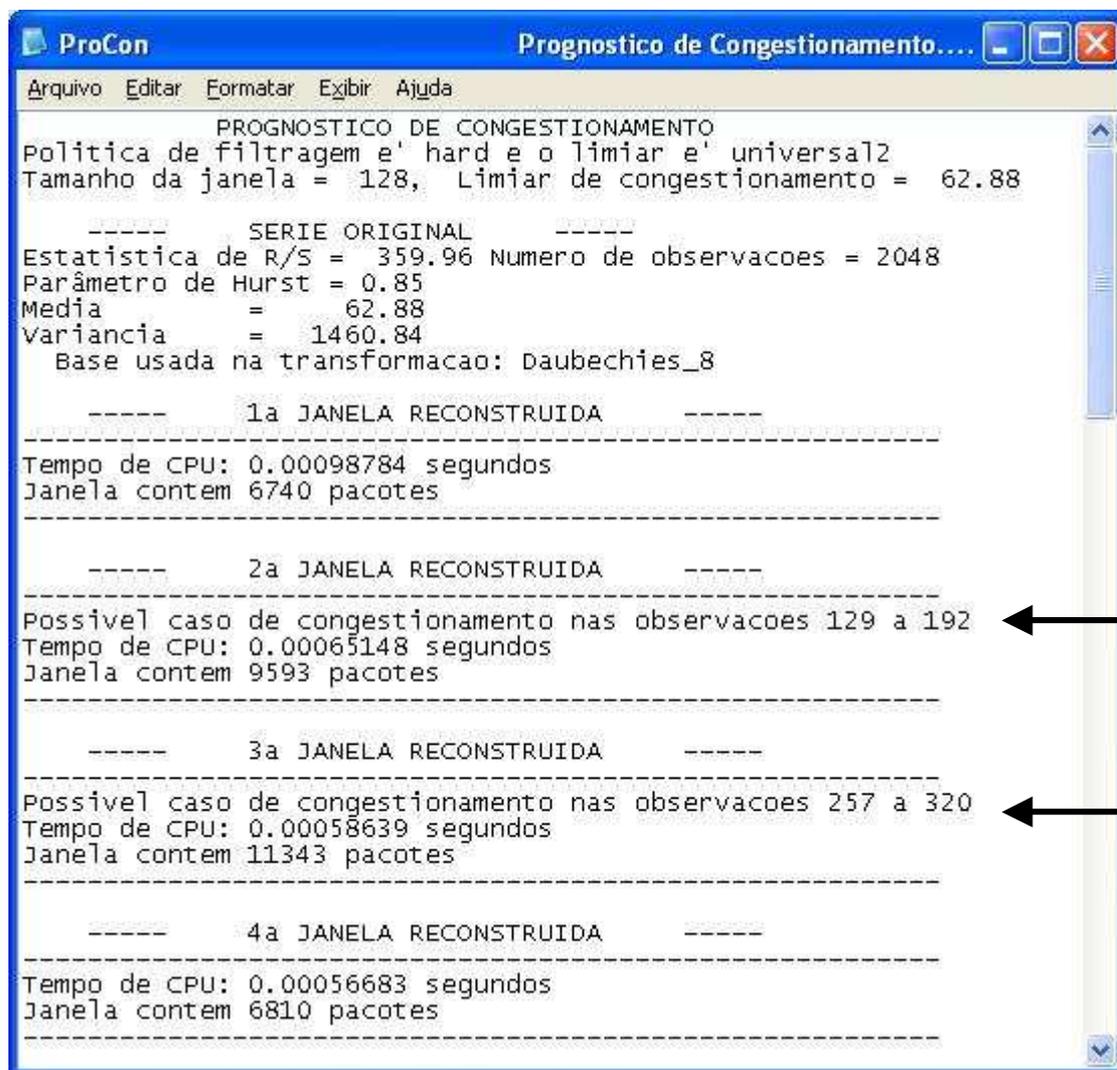


Figura 4.4. Saída do mecanismo ProCon – da 1ª à 4ª janela

As figuras seguintes apresentam as saídas do mecanismo ProCon para as demais janelas de observação, isto é, da 5ª à 16ª janela, e os respectivos gráficos com as séries simuladas e transformadas. A Figura 4.5 mostra a saída do ProCon para a 5ª, 6ª, 7ª e 8ª janelas e a Figura 4.6 apresenta o gráfico das séries para estas janelas, isto é, da observação de número 512 a 1024. Da mesma forma, as Figura 4.7 e 4.8 mostram, respectivamente, a saída do mecanismo e o gráfico correspondente, assim como as Figuras 4.9 e 4.10.



Figura 4.5. Saída do mecanismo ProCon – da 5ª à 8ª janela

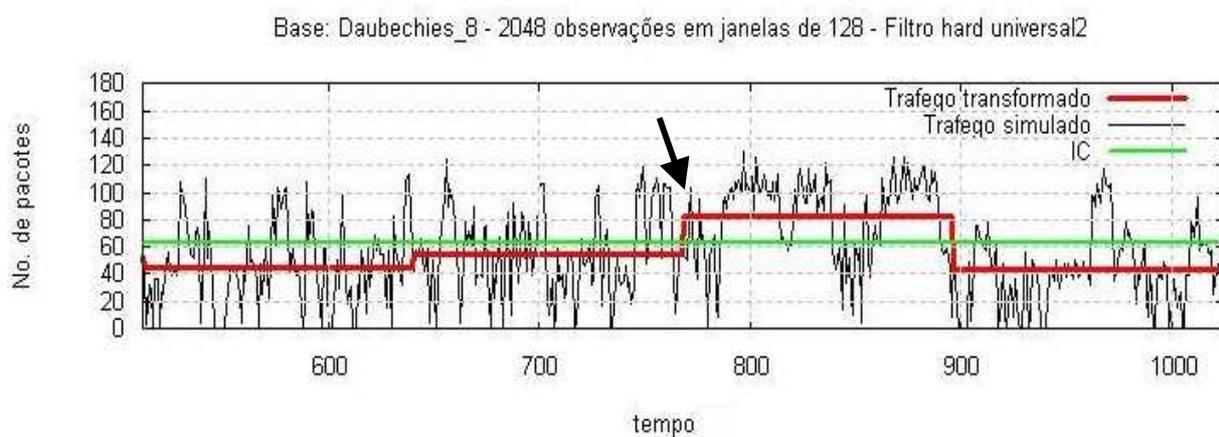


Figura 4.6. Séries original e transformada – da observação de número 513 a 1024

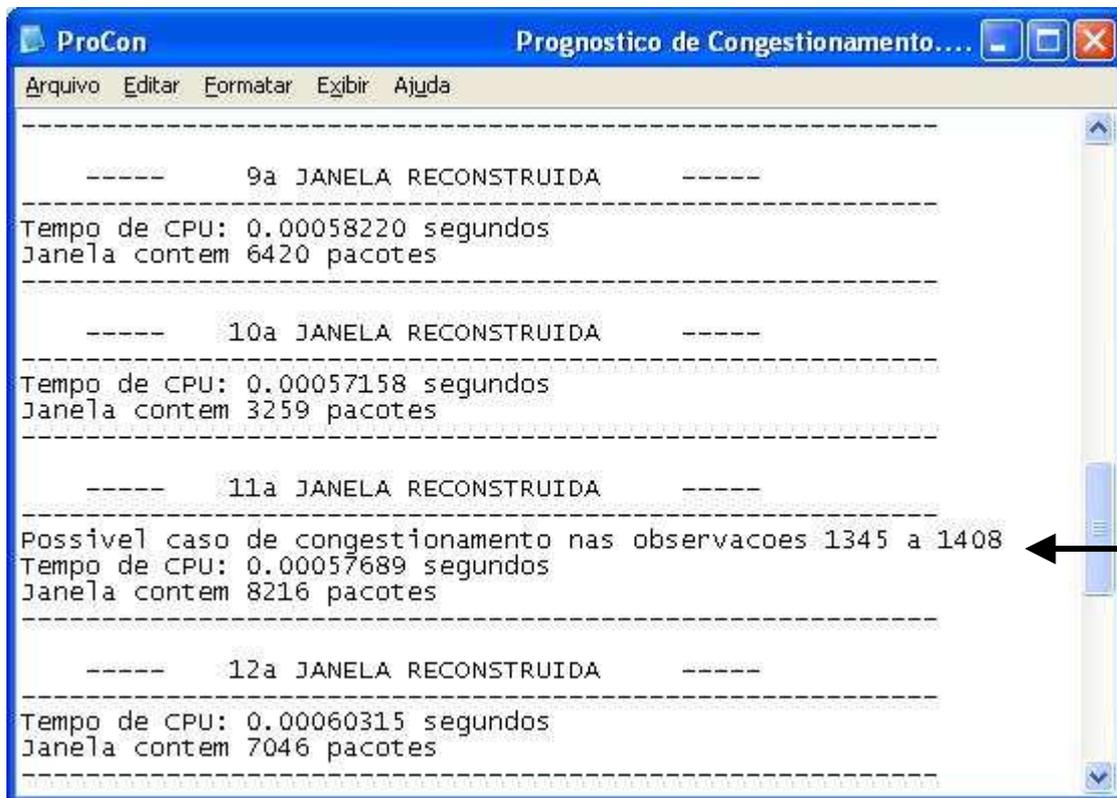


Figura 4.7. Saída do mecanismo ProCon – da 9ª à 12ª janela

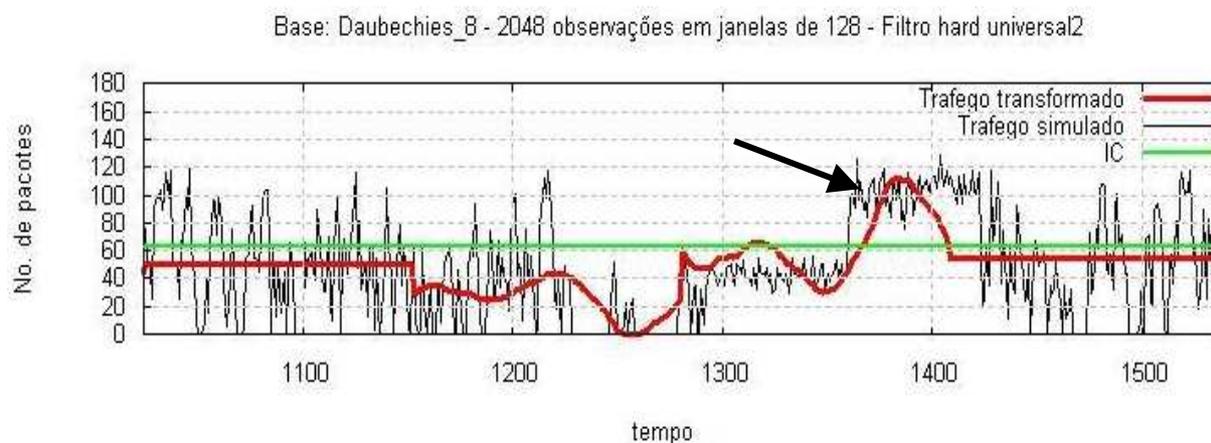


Figura 4.8. Séries original e transformada – da observação de número 1025 a 1536



Figura 4.9. Saída do mecanismo ProCon – da 13ª à 16ª janela

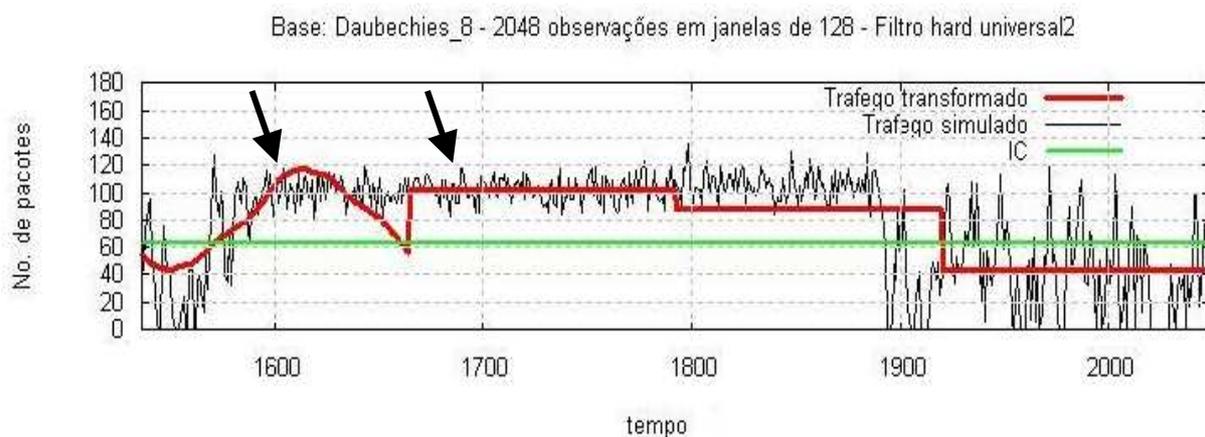


Figura 4.10. Séries original e transformada – da observação de número 1537 a 2048

Observa-se na Figura 4.3, entre os tempos zero e 128, aproximadamente, que o tráfego oscila de 0 a 120 pacotes. A partir daí, entre os tempos 129 a 420 verifica-se que o tráfego simulado permanece a maior parte do tempo com valores acima do patamar de congestionamento (IC). Na Figura 4.6 nota-se que o tráfego simulado oscila de 0 a 120 em grande parte do tempo, com exceção do intervalo de 780 a 900, aproximadamente. Este mesmo comportamento é observado nos tempos 1025 a 1536 (Figura 4.8). A exceção está entre os tempos 1370 a 1430, onde o número de pacotes permanece acima do IC, e entre os tempos 1220 a 1369, onde o número de pacotes está consistentemente abaixo do IC. O último quarto do tráfego, representado pelas observações 1537 a 2048, apresenta comportamento diferente dos três anteriores. Nota-se que o tráfego simulado persiste, na maior parte do tempo, acima do IC entre os tempos 1570 a 1900. Em todas essas situações de persistência do tráfego acima do IC o mecanismo detectou, logo no início da tendência, a possibilidade de congestionamento.

4.3 Resultados Variando os Parâmetros do Mecanismo

Foi visto na seção anterior que os testes do mecanismo com parâmetros estabelecidos como padrões obtêm resultados corretos e identificam a tendência de congestionamento. Também foram feitos testes variando esses parâmetros, isto é, modificando o tamanho da janela, a política de filtragem, o método de escolha do limiar de redução dos coeficientes e a base de *wavelet*. Os resultados obtidos com essas modificações são a seguir descritos.

A Figura 4.11 mostra os resultados gerados pelo mecanismo após a alteração de um único parâmetro, no caso o tamanho da janela de observação de 128 para 512. A simples modificação do tamanho da janela pode resultar em alterações em outros parâmetros. Por exemplo, com a janela de tamanho 512 o mecanismo passou a utilizar a base de *wavelet* Spline_2_2, diferente da obtida quando do uso dos parâmetros padrões (Daubechies_8), como pode ser comprovado na Figura 4.11. Esta nova base foi estimada sobre os dados da 1ª janela (de tamanho 512). Confrontando-se os dados gerados pelo mecanismo nas duas situações, isto é, quando se usa a janela de tamanho 128 – situação 1 (Figuras 4.4, 4.5, 4.6 e 4.7) e quando utiliza a janela de tamanho 512 – situação 2 (Figura 4.11), observa-se que:

- (i) as observações 129 a 288 e 1665 a 1696 foram sinalizadas pelo mecanismo como possível congestionamento em ambas situações;

- (ii) um possível congestionamento nas observações 769 a 880 (situação 1) foi sinalizado antecipadamente pelo mecanismo quando na situação 2 (observações 641 a 768); e
- (iii) o mecanismo não sinalizou congestionamentos nas observações 449 a 480 e 1569 a 1632 quando na situação 2 (janela de tamanho 512), o que ocorreu com a situação 1.

```

ProCon                               Prognostico de Congestionamento ...
Arquivo  Editar  Formatar  Exibir  Ajuda

          PROGNOSTICO DE CONGESTIONAMENTO
Politica de filtragem e' hard e o limiar e' universal
Tamanho da janela = 512, Limiar de congestionamento = 62.88

-----          SERIE ORIGINAL          -----
Estatistica de R/S = 359.96 Numero de observacoes = 2048
Parâmetro de Hurst = 0.85
Media          =          62.88
Variância          =          1460.84
Base utilizada na transformacao: spline_2_2

-----          1a JANELA RECONSTRUIDA          -----
Possivel congestionamento nas observacoes de 129 a 256
Possivel congestionamento nas observacoes de 257 a 384

Tempo de CPU: 0.010 segundos
Relogio: iniciou as 1:29:35:64 e terminou as 1:29:35:65

-----          2a JANELA RECONSTRUIDA          -----
Possivel congestionamento nas observacoes de 641 a 768

Tempo de CPU: 0.000 segundos
Relogio: iniciou as 1:29:35:65 e terminou as 1:29:35:65

-----          3a JANELA RECONSTRUIDA          -----
Tempo de CPU: 0.000 segundos
Relogio: iniciou as 1:29:35:65 e terminou as 1:29:35:65

-----          4a JANELA RECONSTRUIDA          -----
Possivel congestionamento nas observacoes de 1665 a 1792

Tempo de CPU: 0.000 segundos
Relogio: iniciou as 1:29:35:65 e terminou as 1:29:35:65

```

Figura 4.11. Saída do mecanismo ProCon com janela de observação de tamanho 512

A explicação para essas discrepâncias de sinalização do mecanismo está nas diferenças das bases utilizadas, Daubechies_8 na situação 1 e Spline_2_2 na situação 2, e nos tamanhos

das janelas. Uma terceira situação, a execução do mecanismo com a janela de tamanho 512 (situação 2), mas com a mesma base da situação 1 (Daubechies_8), produziu os resultados que estão na Figura 4.12(b). A Figura 4.12(a) serve de comparação e corresponde aos resultados gerados pelo mecanismo na situação 2.

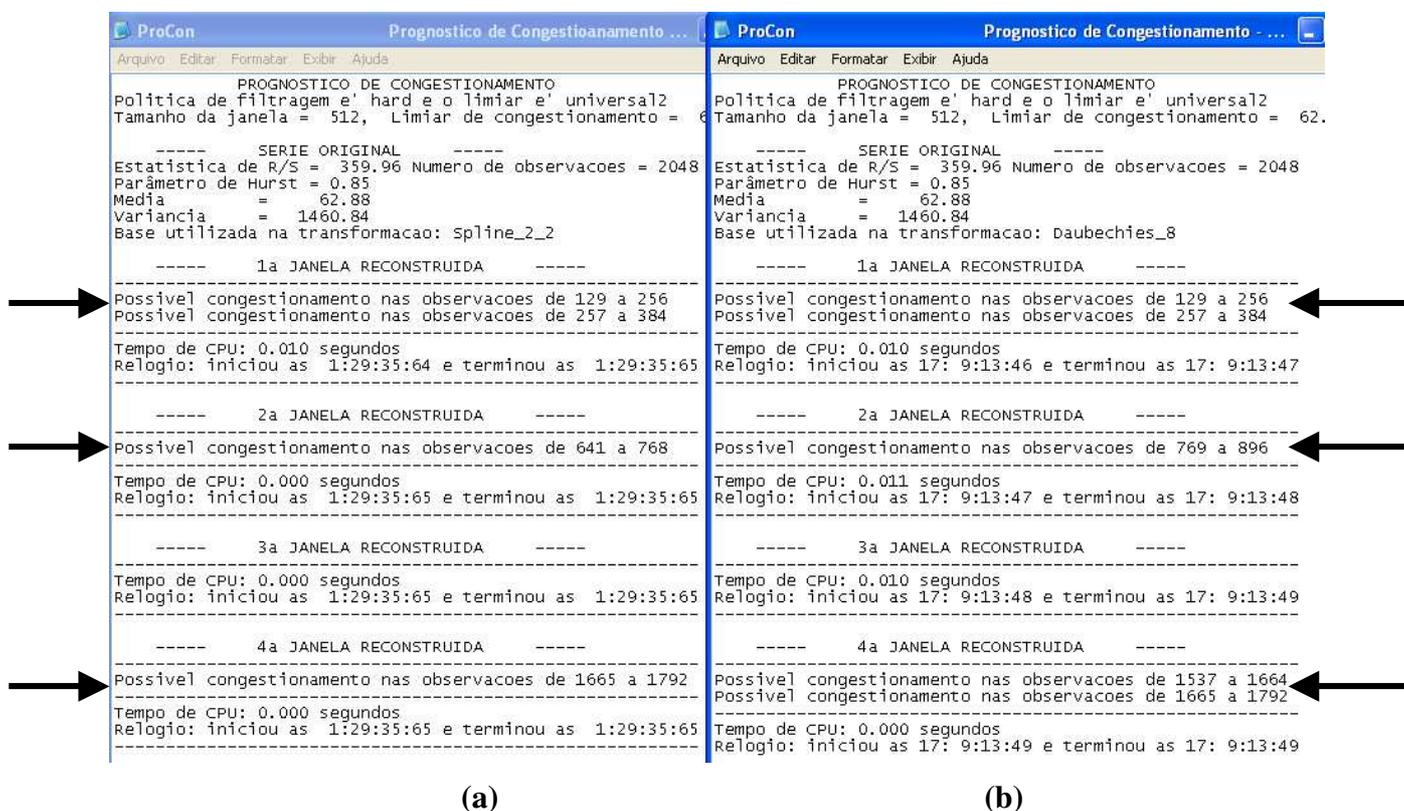


Figura 4.12. Saída do mecanismo usando as bases (a) Spline_2_2 e (b) Daubechies_8

Esta terceira situação (Figura 4.12-(b)) produziu resultados diferentes dos resultados da situação 2 (Figura 4.12-(a)) em decorrência das bases não serem as mesmas. Porém, confrontando-se os dados gerados pelo mecanismo na situação 3 com os gerados na situação 1, ambos com as mesmas bases mas com janelas diferentes, nota-se que:

- (i) as observações 129 a 288, 769 a 800 e 1569 a 1696 foram sinalizadas pelo mecanismo como possível congestionamento em ambas situações; e
- (ii) as observações 449 a 480 e 1377 a 1408 foram sinalizadas pelo mecanismo na situação 1, mas não na situação 3.

Os dados gerados pelo mecanismo na situação 3 estão mais próximos dos dados da situação 1. Somente dois intervalos de observações (449 a 480 e 1377 a 1408) não foram detectados na situação 3. Esta diferença deve-se ao tamanho da janela. Enquanto na situação 1

o mecanismo executou 16 vezes o teste de tendência (janela de tamanho 128), na situação 3 foram apenas quatro. Na situação 3 o mecanismo tem uma “visão” macro do comportamento do tráfego, enquanto na situação 1 os detalhes sobressaem.

A execução do mecanismo usando os métodos de seleção de limiar “quantil” e “universal-1” apresentaram resultados insatisfatórios, isto é, as séries transformadas não foram suficientemente suavizadas, como pode ser visto na Figura 4.13.

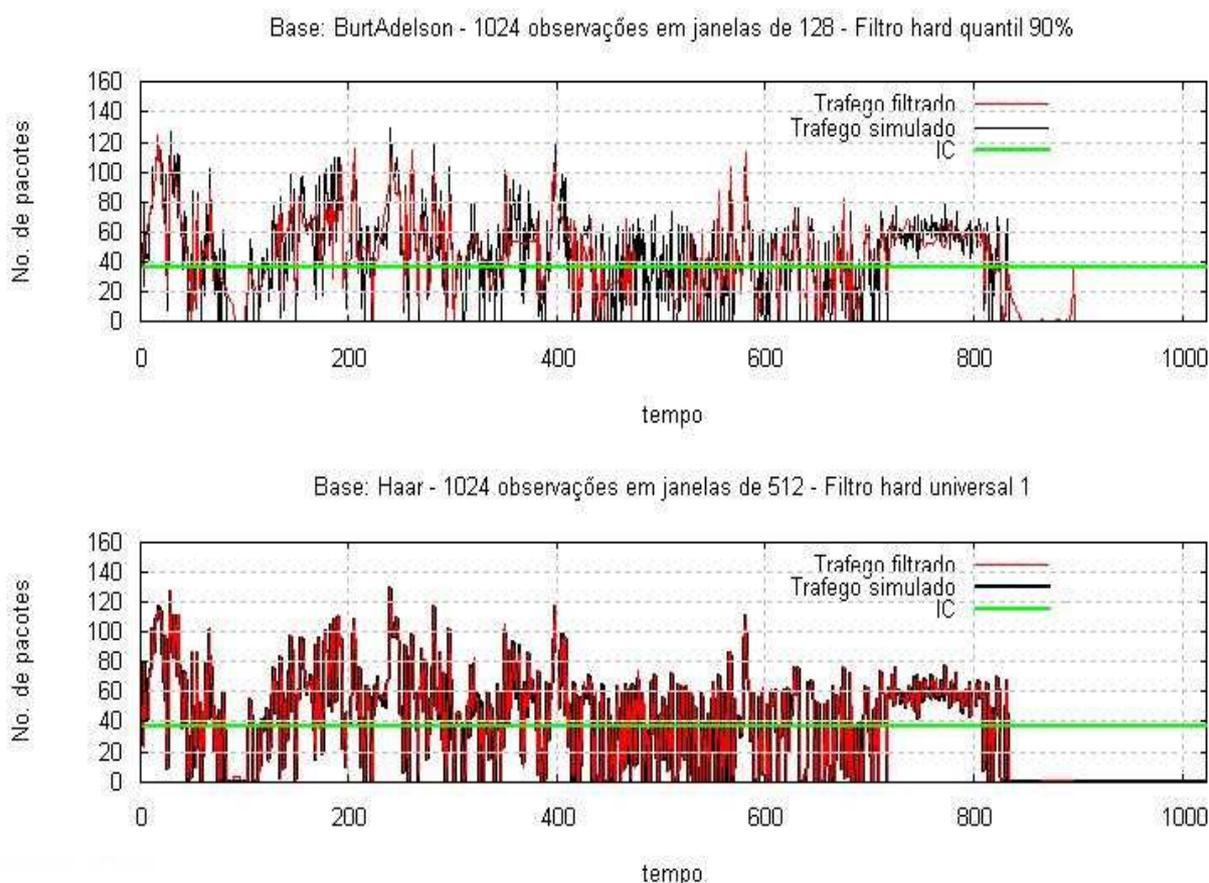


Figura 4.13. Séries de 1024 observações com métodos de limiar “quantil” e “universal-1”

A execução do mecanismo com a política de filtragem “soft” também não apresentaram resultados satisfatórios como se pode ver na Figura 4.14. A política “soft”, diferentemente da “hard”, não zera os coeficientes durante o procedimento de filtragem e, por isso, o tráfego filtrado e transformado apresentou variabilidade maior que o tráfego simulado.

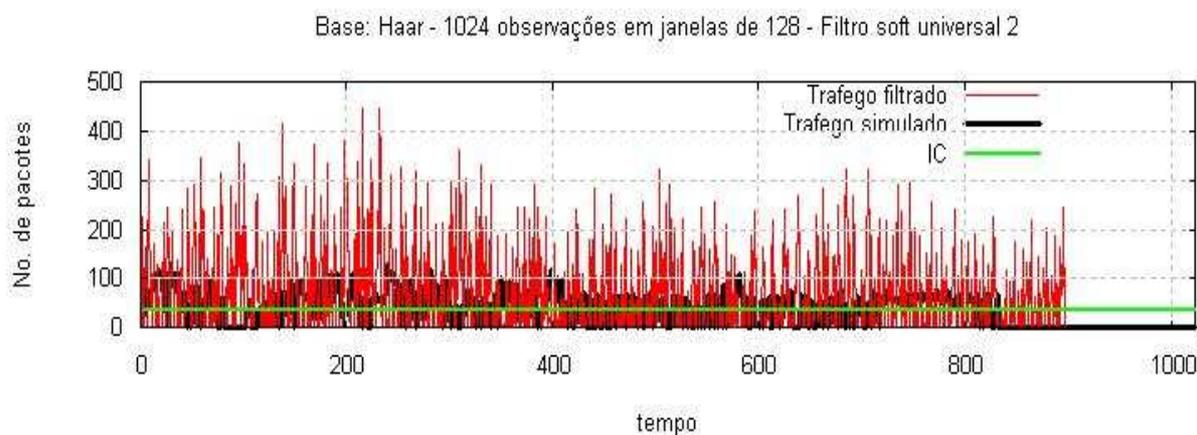


Figura 4.14. Série de 1024 observações com política de filtragem “soft”

4.4 Tempo de Execução do Mecanismo

O ProCon foi projetado para trabalhar em tempo real e utiliza o relógio da UCP do equipamento, no qual está instalado, para executar seus procedimentos no momento certo e de forma constante, mesmo que o processamento da janela anterior tenha terminado. O tempo da UCP de equipamentos roteadores não é medido em horas, minutos e segundos, mas em função dos ciclos do processador, isto é, depende da frequência (medida em *hertz*). Por exemplo, se no início da execução do mecanismo o tempo medido é zero e ao final o tempo é 20, significa dizer que o equipamento utilizou 20 ciclos para executar os comandos do programa. A divisão do intervalo de tempo, entre duas medições sucessivas do relógio da UCP, por 1000 representa o tempo decorrido em segundos.

O mecanismo ProCon, durante a sua execução, consulta o relógio da UCP antes do cálculo da transformada discreta *wavelet* e após o teste de tendência. A diferença entre a 2^a consulta ao relógio e a 1^a corresponde ao tempo demandado pelo mecanismo para realizar um prognóstico, isto é, para executar os procedimentos de transformação da série, de filtragem dos coeficientes, de transformação inversa e de teste da tendência. Para estimar o tempo médio de execução do mecanismo, foram simuladas 30 (trinta) séries de tráfego com 2048 observações cada.

O Quadro 4.3 apresenta os tempos de execução do mecanismo em três diferentes computadores. Os testes realizados com o ProCon mostram que, para janelas de tamanho 128, os tempos médios de execução medidos, usando os processadores Celeron e Pentium IV, estão na casa de microssegundos (608 e 350 microssegundos respectivamente) e com o

processador Pentium MMX o tempo médio é de 5,436 milissegundos. Para janelas de tamanho 512, os tempos medidos, usando o processador Pentium IV, estão abaixo de 2 milissegundos, com o processador Celeron próximos de 5 milissegundos e no Pentium MMX acima de 20 milissegundos. É importante ressaltar que o mecanismo demora menos de 1 milissegundo para processar uma janela de tamanho 128 usando processadores mais rápidos de clock 933 MHz e 2,2 GHz (vide as saídas do mecanismo representadas pelas Figuras 4.4, 4.5, 4.7 e 4.9), e na ordem de 20 milissegundos, para uma janela de tamanho 512, em um computador mais lento (233 MHz).

Considerando-se: (i) que um prognóstico de congestionamento é para um tempo futuro de curto prazo da ordem de unidades de milissegundos, (ii) que a coleta do tráfego é, usualmente, realizada a cada 1 milissegundo e (iii) que os tempos obtidos pelo ProCon para realizar um prognóstico em uma janela de tamanho 128, usando os processadores MMX está acima de 1 milissegundos, poder-se-ia inferir que o mecanismo proposto não constitui uma aplicação viável em roteadores lentos. Todavia, considerando-se ainda: (i) que testes realizados com o mecanismo, utilizando os três processadores citados e janelas de tamanho 64, mostraram que os tempos demandados para um prognóstico são inferiores a 1 milissegundo (da ordem de 400 microssegundos), (ii) que a programação do mecanismo não está otimizada, (iii) que nos processador de 933 MHz e 2.2 GHz, o ProCon gastou menos de 1 milissegundo para cada janela de tamanho 128, (iv) que o mecanismo proposto deve ser “embarcado ou embutido” em um equipamento, como um roteador, (v) que a linguagem de alto nível usada na programação do ProCon, C++ não é usada em sistemas embarcados, poder-se-ia concluir que o mecanismo é viável em relação aos tempos apresentados.

Quadro 4.3. Tempos de execução

Equipamento	Tamanho da janela	Tempo médio (segundos)
Processador Pentium MMX, 233 MHz	128 observações	0,005436
	512 observações	0,021252
Processador Celeron, 995 MHz	128 observações	0,000608
	512 observações	0,005563
Processador Pentium IV, 2.2 GHz	128 observações	0,000350
	512 observações	0,001760

4.5 Validação do Mecanismo ProCon

A comparação do prognóstico efetuado pelo ProCon com outras técnicas de previsão de séries temporais serve para avaliar a exatidão dos prognósticos do mecanismo. Para aplicar os procedimentos de comparação de prognósticos, foram simuladas 30 (trinta) séries de tráfego com 2048 observações cada. Para cada uma dessas séries, o ProCon utilizou, baseado no projeto experimental citado na Seção 4.1, os seguintes parâmetros: (i) janela de tamanho 64, (ii) base de *wavelet* estimada pelo próprio mecanismo, (iii) política *hard* de redução de coeficientes, e (iv) limiar universal-2. No total, foram prognosticadas 960 séries de tamanho 64, isto é, cada série de tráfego com 2048 observações contém 32 séries de 64 observações.

O modelo FARIMA foi escolhido para modelar e prever o comportamento dessas séries e comparar com os prognósticos do ProCon. Optou-se por este modelo, primeiro devido à natureza auto-similar e a não-estacionaridade das séries simuladas. Processos FARIMA são, como citado no Capítulo 2, assintoticamente auto-similares. Segundo, em função do parâmetro de diferenciação fracionária d desse modelo poder ser estimado a partir do parâmetro Hurst calculado pelo próprio mecanismo. O processo de estimação e previsão usando FARIMA foi feito através de um pacote baseado na ferramenta Ox (DOORNIK; OOMS, 2001).

A comparação do mecanismo com o modelo FARIMA consistiu na contagem do número de casos de congestionamento detectados ou não, onde se sabia previamente as janelas que continham esses casos. Também foi computado o número de casos de congestionamento detectados por ambos em que não havia congestionamento. O Quadro 4.4 mostra os números de casos de congestionamento detectados pelo ProCon e o Quadro 4.5 mostra os números do modelo FARIMA.

Quadro 4.4. Número de acertos do ProCon

ProCon	Detectado		Não detectado		Total
Havia congestionamento	240	73%	90	27%	330
Não havia congestionamento	57	9%	573	91%	630
Total	297		663		960

Quadro 4.5. Número de acertos do modelo FARIMA

FARIMA	Detectado		Não detectado		Total
Havia congestionamento	262	79%	68	21%	330
Não havia congestionamento	75	12%	555	88%	630
Total	337		623		960

Observa-se nos quadros citados, que o modelo FARIMA apresenta um percentual de acertos maior que o mecanismo ProCon quando realmente há congestionamento. Contudo, nos casos em que não há congestionamento, o modelo FARIMA detectou mais falsos congestionamentos do que o ProCon.

Apesar do mecanismo ProCon ter acertado menos que o modelo FARIMA, o seu processamento foi extremamente rápido quando comparado com o FARIMA. Enquanto o ProCon processou uma série de 2048 observações em menos de 30 milissegundos, a ferramenta Ox demandou mais de 15 minutos para processar a mesma série em um processador de 933 MHz.

4.6 Comentários Finais

Este capítulo apresentou os resultados obtidos e suas análises, os resultados variando os parâmetros do mecanismo e os tempos de execução.

Quais os questionamentos que poderiam ser levantados sobre o mecanismo proposto? Algumas indagações, relacionadas a seguir, tentam cobrir esses questionamentos.

- a) Por que não aplicar um método de previsão de séries temporais ao invés da transformada de *wavelet*?
 - A aplicação pura e simples de um método de previsão ou de um teste de tendência em tráfego real produz indicadores errôneos sobre seu comportamento devido à natureza auto-similar do tráfego atual. Rajadas súbitas de curtíssima duração poderiam ser confundidas com congestionamento.
- b) Qual a confiabilidade do prognóstico de congestionamento, isto é, quando é que o mecanismo de prognóstico funciona bem?

- O mecanismo mostrou ser eficiente pois detectou possíveis congestionamentos nos tráfegos simulados usando os parâmetros: (i) “*hard*”, (ii) “*universal-2*” e (iii) janela de tamanho 128, que favorecem a precisão do prognóstico. Com esses parâmetros, o mecanismo soube identificar os congestionamentos, separando-os de rajadas súbitas de curta duração.

c) Qual a viabilidade de aplicação do mecanismo proposto em função dos tempos de execução?

- Usando os parâmetros citados anteriormente, o mecanismo detectou um possível congestionamento em média de 1,35 milissegundos. Usualmente, existem *softwares* que coletam tráfego em até 1 milissegundo. A otimização da programação, o uso de processadores mais rápidos e a implementação do mecanismo em sistemas embarcados são fatores que contribuirão para a diminuição do tempo de execução, viabilizando sua utilização.

d) Como avaliar a qualidade do mecanismo proposto em função da comparação de resultados com o modelo FARIMA?

- Os métodos de previsão de séries temporais são mais apropriados para séries estacionárias, o que não é o caso do tráfego das redes atuais. As *wavelets* são adequadas para análise de séries não-estacionárias, contendo transitoriedades e estruturas tipo fractais. O modelo FARIMA é o único sistema linear de previsão de séries temporais adequado para séries não-estacionárias e auto-similares. Além disso, não existe na literatura proposta semelhante usando outros modelos para que se possa efetuar outras comparações de resultados.

e) Quais as ações que devem ser tomadas após a iminência de um congestionamento?

- O mecanismo não substitui o controle de congestionamento. Os atuais mecanismos de controle de congestionamento baseados, por exemplo, em gerenciamento ativo de filas, continuam com suas funções dentro de um roteador. O mecanismo somente detecta possíveis casos de congestionamento. A partir da percepção do congestionamento, o ProCon poderia alertar o mecanismo de controle de congestionamento.

Capítulo 5

Conclusões

Foi apresentado um mecanismo que aplica a teoria de *wavelet* para prognosticar congestionamento de tráfego de redes. Este mecanismo consiste: (i) da aplicação da transformada de *wavelet* discreta em um conjunto de dados representando um tráfego de redes, (ii) da filtragem dos coeficientes para redução do ruído presente, (iii) da transformação inversa de *wavelet*, (iv) do teste de tendência e (v) do prognóstico do comportamento futuro do tráfego.

A proposta deste mecanismo é uma nova abordagem para prognósticos de congestionamentos de tráfego de redes. O mecanismo proposto infere somente prognóstico de tendência, mas não prevê valores futuros e por isso, não pode ser comparado com métodos e/ou mecanismos que fazem previsão de séries temporais. Convém salientar que o ProCon pode ser utilizado por mecanismos de controle de congestionamentos situados nos nodos intermediários das redes, tornando-os pró-ativos.

As principais conclusões deste trabalho são:

- (i) a aplicação pura e simples de métodos de previsão de séries temporais para prever congestionamentos não revela o real comportamento do tráfego devido à natureza auto-similar do tráfego atual;
- (ii) a transformada de *wavelet* discreta, a redução dos coeficientes e a transformada inversa geram uma série suavizada;
- (iii) testes de tendência do comportamento da série, que antes não podiam ser aplicados em decorrência da natureza auto-similar do tráfego de redes, podem ser usados em função da suavização da série reconstruída pela transformada;
- (iv) os resultados apresentados pelo mecanismo validam os itens (i) e (ii);
- (v) a escolha correta dos parâmetros do mecanismo proposto, tais como tamanho da janela de observação, política de filtragem, método de escolha do limiar de redução dos coeficientes e patamar de congestionamento, favorece a corretude do prognóstico;

- (vi) os tempos de execução obtidos, da ordem de unidades de milissegundos, e a possibilidade de otimização do mecanismo viabilizam sua utilização.

A seleção correta ou não dos parâmetros do mecanismo ocasiona maior ou menor precisão do prognóstico de congestionamento. O projeto fatorial (Seção 4.1) elaborado com os resultados do mecanismo mostraram que a política “soft” e os métodos “universal-1” e “quantil” são parâmetros inadequados para prognosticar tendências, enquanto que a política “hard” e o método “universal-2” favorecem suposições de tendências mais exatas.

A seleção correta da base de *wavelet* também influi na precisão do prognóstico de congestionamento. Testes com o mecanismo evidenciaram que a base que confere maior precisão do prognóstico é a da transformada de menor valor de desvio quadrático, em relação à série original.

O tamanho da janela de observação é importante para a exatidão do prognóstico. O projeto fatorial citado mostrou que janelas com tamanho de 512 ou 1024, por exemplo, diminuem a precisão, enquanto janelas de menor tamanho, por exemplo 64 ou 128, aumentam a precisão.

As principais contribuições desta tese são:

- (i) o estudo da transformada de *wavelet* com propósitos de realizar prognósticos de congestionamento;
- (ii) a proposta do próprio mecanismo, da sua arquitetura e funções;
- (iii) a aplicabilidade prática do mecanismo ProCon; e
- (iv) a implementação do mecanismo.

5.1 Trabalhos Futuros

Alguns trabalhos estão em andamento, que são:

- (i) implementação do modelo de tráfego 4IPP para gerar tráfego simulado;
- (ii) implementação do modelo de tráfego Weibull-Pareto.

Apesar de ambos serem trabalhos marginais para os objetivos do mecanismo, estas implementações fornecerão dados que simulam o tráfego WWW, permitindo mais verificações da corretude do mesmo.

Como trabalhos futuros sugere-se:

- (i) usar o esquema *Lifting* nas transformadas de *wavelet* para reduzir o custo computacional do mecanismo e também diminuir o tamanho do código;
- (ii) estimar a dimensão fractal do tráfego de redes, isto é, identificar se o tráfego é monofractal (auto-similar) ou multifractal. Isto permitiria verificar a corretude do mecanismo para o tráfego multifractal;
- (iii) fazer uma verificação do mecanismo usando dados reais; e
- (iv) por fim, aplicar o mecanismo em sistemas embarcados usando tráfego real. Esta é a maior contribuição desta tese para um trabalho futuro

Referências Bibliográficas

(De acordo com a NBR 6023 da ABNT, de agosto de 2002)

ABRY, P.; FLANDRIN, P.; TAQQU, M. S.; VEITCH, D., “Long-Range Dependence: Theory and Applications”. Capítulo do livro *Self-Similarity and Long Range Dependence through the Wavelet Lens*, Birkhauser, 2002.

ABRY, P.; GONÇALVES, P.; FLANDRIN, P., “Wavelets, Spectrum Estimation and 1/f Processes”. In: *Wavelets and Statistics, Lectures Note in Statistics*, v.103, p.15-30, Springer-Verlag, NY, 1995.

ABRY, P.; VEITCH, D., “Wavelet Analysis of Long-Range Dependent Traffic”. *IEEE Transactions Inf. Theory*, v. 44, n. 1, p. 2-15, 1998.

ADAS, A.; MUKHERJEE, A., “On Resource Management and QoS Guarantees for Long Range Dependent Traffic”. In: *IEEE INFOCOM’95, Proceedings ...*, p. 779-787, 1995.

ADDIE, R.; ZUKERMAN, M.; NEAME, T., “Fractal Traffic: Measurements, Modeling and Performance Evaluation”. In: *IEEE INFOCOM’95, Proceedings ...*, p. 977-984, 1995.

ADELSON; SIMONCELLI; HINGORANI, "Orthogonal Pyramid Transforms for Image Coding". In: *SPIE, Visual Communications and Image Processing II, Proceedings...*, v. 845, p. 50-58, 1987.

ALLMAN, M.; PAXSON V.; STEVENS W., “TCP Congestion Control”. Request for Comments 2581, IETF, Apr. 1999.

ANDERSEN, A. T.; NIELSEN, B. F., “A Markovian Approach for Modeling Packet Traffic with Long-Range Dependence”, *IEEE Journal on Selected Areas in Communications*, v.16, n.5, p.719-732, June 1998.

ANTONINI, M.; BARLAUD, M.; MATHIEU, P.; DAUBECHIES,I., "Image Coding Using Wavelet Transforms". *IEEE Trans. on Image Processing*, Vol. 1 No. 2, April 1992.

AWDUCHE, D. O., "MPLS and Traffic Engineering in IP Networks". *IEEE Communications Magazine*, p. 42-47, Dec. 1999.

AWDUCHE, D. O.; CHIU, A.; ELQALID, A.; WIDJAJA, I.; XIAO, X., “A Framework for Internet Traffic Engineering”. Internet Draft, <draft-ietf-te-framework-01.txt>, May 2000.

AWDUCHE, D. O.; MALCOLM, J.; AGOGBUA, J.; O'DELL, M.; MCMANUS, J. "Requirements for Traffic Engineering over MPLS". Request for Comments 2702, IETF, Sept. 1999.

BAGAL, P.; KALYANARAMAN, S.; PACKER, B., "Comparative Study of RED, ECN and TCP Rate Control". Technical Report, Rensselaer Polytechnic Institute, Mar, 1999. Disponível em: <http://www.packeteer.com/technology/Pdf/packeteer-final.pdf>.

BALAKRISHNAN, H.; SESHAN, S., "The Congestion Manager". Request for Comments 3124, IETF, June 2001.

BAUGH, C. R., "4IPP Traffic Model for IEEE 802.16.3". IEEE 802.16.3c-00/51, Meeting #10, Tampa, Oct. 2000.

BAUGH, C. R.; HUANG, J.; SCHWARTZ, R.; TRINKWON, D., "Traffic Model for IEEE 802.16.3 TG3 MAC/PHY Simulations". IEEE 802.16.3c-00/30r1, Task Group 3, Mar. 2001.

BENMOHAMED, L.; MEERKOV, S. M., "Feedback Control of Congestion in Store-and-Forward Networks: the Case of a Single Congested Node". *IEEE/ACM Transactions on Networking*, v.1, n.6, p.693-707, Dec. 1993.

BERAN, J., "Statistics for Long-Memory Processes". Chapman & Hall, 1994.

BERAN, J.; SHERMAN, R.; TAQQU, M. S.; WILLINGER, W., "Long-Range Dependence in Variable-Bit-Rate Video Traffic". *IEEE Transactions on Communications*, v. 43, p.1566-1579, 1995.

BERNET, Y.; FORD, P.; YAVATKAR, R.; BAKER, F.; ZHANG, L.; SPEER, M.; BRADEN, R.; DAVIE, B.; WROCLAWSKI, J.; FELSTAIN, E., "A Framework for Integrated Services Operation over DiffServ Networks". Request for Comments 2998, IETF, Nov. 2000.

BEYLKIN, G.; COIFMAN, R.; ROKHLIN, V., "Fast Wavelet Transforms and Numerical Algorithms I". *Communication on Pure and Applied Mathematics*, v. 44, p. 141-183, 1991.

BOX, G. P.; JENKINS, G. M., "Time Series Analysis, Forecasting and Control". Revised Edition, San Francisco, Holden-Day, 1976, p. 1-19.

BRADEN, B.; CLARK, D.; CROWCROFT, J.; DAVIE, B.; DEERING, S.; ESTRIN, D.; FLOYD, S.; JACOBSON, V.; MINSHALL, G.; PARTRIDGE, C.; PETERSON, L.; RAMAKRISHNAN, K.; SHENKER, S.; WROCLAWSKI, J.; ZHANG, L.,

“Recommendations on Queue Management and Congestion Avoidance in the Internet”. Request for Comments 2309, IETF, Apr. 1998.

BRYANT, S. F.; BRASH, D. L., “The DECNIS 500/600 Multiprotocol Bridge/Router and Gateway”. *Digital Technical Journal*, v.5, n1, 1993.

CAMPOS, M. A.; SILVA, J. L. C.; CUNHA, P. R. F., “Modelagem Estocástica em Redes de Alta Velocidade”. In: XXI Congresso da Sociedade Brasileira de Computação (XIX JAI), **Anais...** V. 2, p. , Fortaleza, CE, ago. 2001.

CHRISTIANSEN, M.; JEFFAY, K.; OTT, D.; SMITH, F.,”Tuning RED for Web Traffic”. In: SIGCOMM’00, **Proceedings ...**, Aug. 2000.

CISCO, “Cisco 12000 Gigabit Switch Router,” *White Paper*, Cisco Systems, 1997.

CONOVER, W. J. “Practical Nonparametric Statistics”. John Wiley and Sons, N.Y. 1971.

CRAWLEY, E.; NAIR, R.; RAJAGOPALAN, B.; SANDICK, H., “A Framework for QoS-based Routing in the Internet”. Request for Comments 2386, IETF, Aug. 1998.

CROVELLA, M.; BESTAVROS, A., “Sel-Similarity in World Wide Web Traffic: Evidence and Possible Causes”. *IEEE/ACM Transaction on Networking*, v.5, p.835-846, 1997.

DAUBECHIES, I. “Ten Lectures on Wavelets”. SIAM, Philadelphia, 1992.

DELBEKE L. “Wavelet based Estimators for the Scaling Index of a Self-Similar Process with Stationary Increments”. Ph.D. Thesis, KU Leuven, Belgium, 1998.

DONOHO, D. L., “Interpolating Wavelet Transforms”. Preprint, Department of Statistics, Stanford University, 1992.

DONOHO, D. L.; JOHNSTONE, I. M., “Adapting to unknown smoothing via wavelet shrinkage”. *Journal American of Statistics Association*, v.90, p.1200-1224, 1995.

DONOHO, D. L.; JOHNSTONE, I. M.; KERKYACHARIAN, G.;PICARD, D. “Universal near Minimaxy of Wavelet Shrinkage”. Technical Report, Stanford University, 1995.

DOORNIK, J. A.; OOMS, M., “Introduction to Ox”, Timberlake Consultants Press, 2001.

FAUCHER, F. L.; WU, L.; DAVIE, B.; DAVARI, S.; VAANANEN, P.; KRISHNAN, R.; CHEVAL, P.; HEINANEN, J., “MPLS Support of Differentiated Services”. Request for Comments 3270, IETF, May 2002.

FELDMAN, A.; GILBERT, A. C.; HUANG, P.; WILLINGER, W. "Dynamics of IP Traffic; a Study of the Role of Variability and the Impact of Control". In: ACM SIGCOMM'99, **Proceedings ...**, p.301-313, 1999.

FELDMAN, A.; GILBERT, A. C.; WILLINGER, W., "Data Networks as Cascades: Investigating the Multifractal Nature of Internet WAN Traffic". In: ACM SIGCOMM'98, **Proceeding...**, p. 25-38, 1998.

FENG, W., "Improving Internet Congestion Control and Queue Management Algorithms". Ph.D. Thesis, The University of Michigan, Ann Harbor, Michigan, 1999.

FENG, W.; SHIN, K. G.; KANDLUR, D.; SAHA, D., "Blue: An Alternative Approach To Active Queue Management". In: NOSSDAV'01, **Proceedings ...**, July 2001a.

FENG, W.; KANDLUR, D.; SAHA, D.; SHIN, K. G., "Stochastic Fair Blue: A queue management algorithm for enforcing fairness". In: IEEE INFOCOM, **Proceedings ...**, pp. 1520-1529, Apr. 2001b.

FLANDRIN, P., "Wavelet Analysis and Synthesis of Fractional Brownian Motion". *IEEE Transactions on Information Theory*, V. 38, N. 2, pp. 910-917, 1992.

FLANDRIN, P. "Fractional Brownian Motion and Wavelets". *Wavelets, Fractals and Fourier Transforms*, p.109-122, Clarendon Press, Oxford, 1993.

FLOYD, S., "Recommendation on Using the Gentle Variant of RED". Mar. 2000a. Disponível em: <<http://www.aciri.org/floyd/red/gentle.html>. Acesso em: 30 abril 2002.

FLOYD, S., "Congestion Control Principles". Request for Comments 2914, IETF, Mar. 2000b.

FLOYD, S., "A Report on Some Recent Developments in TCP Congestion Control". *IEEE Communications Magazine*, Apr. 2001.

FLOYD, S.; FALL, K., "Promoting the Use of End-to-End Congestion Control in the Internet". *IEEE/ACM Transactions on Networking*, Aug. 1999.

FLOYD, S.; GUMMADI, R.; SHENKER, S., "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management". Berkeley, CA, 2001. Disponível em: <http://www.icir.org/floyd/papers/adaptiveRed.pdf>

FLOYD, S.; JACOBSON, V., "Random Early Detection Gateways for Congestion Avoidance". *IEEE/ACM Transactions on Networking*, v. 1, n. 4, p. 397-413, Aug. 1993.

FLOYD, S.; KEVIN, F., “Promoting the Use of End-to-End Congestion Control in the Internet”. *IEEE/ACM Transactions on Networking*, v. 7, n. 4, p. 458-472, Aug. 1999.

GALLAGER, R.; BERTSEKAS, D., “Data Networks”. Prentice-Hall, 2nd edition, 1992.

GARRET, M.; WILLINGER, W., “Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In: ACM SIGCOMM’94, **Proceedings ...**, p. 269-280, 1994

GILBERT, A. C.; WILLINGER, W.; FELDMANN, A., “Scaling Analysis of Conservative Cascades with Applications to Network Traffic”. *IEEE Trans. Information Theory*, v.45, n.3, p.971-991, 1999.

GOMES, J.; VELHO, L.; GOLDSTEIN, E. “Wavelets: Teoria, Software e Aplicações”. In: 21^o Colóquio Brasileiro de Matemática. **Anais...**, IMPA, Rio de Janeiro, 1997.

GOMES, J.; VELHO, L.; GOLDSTEIN, E. “From Fourier Analysis to Wavelets”. In: SIGGRAPH’99. **Anais...**, 1999, 196p.

GORDON, J., “Pareto Process as a Model of Self-Similar Packet Traffic. In: IEEE GLOBECOM '95, **Proceedings...**, v. 3, p. 2232-2236, Singapore, Nov. 1995.

GUO L.; CROVELLA M. E.; MATTA I., “How does TCP Generate Pseudo Self-Similarity?”. Technical Report, BUCS-TR-2000-017, Computer Science Dept., Boston University, 2000.

GUPTA, P.; LIN, S.; MCKEOWN, N., “Routing Lookups in Hardware at Memory Access Speeds”. In: IEEE Infocom’98, **Proceedings...**, Mar. 1998.

HEFFES, H.; LUCANTONI, D. M., “A Markov Modulated Characterization of Packetized Voice and Data Traffic”. *IEEE Journal on Selected Areas in Communications*, v.4, p.856-869, Sept. 1986.

HOLLOT, C.; MISRA, V.; TOWSLEY, D.; GONG, W-B., “A Control Theoretic Analysis of RED”. In: IEEE Infocom, **Proceedings ...**, Apr. 2001. Disponível em: <http://www-net.cs.umass.edu/papers/papers.html>.

HUANG, C.; DEVETSIKIOTIS, M.; LAMBADARIS, I.; KAYE, A., “Modeling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach”. In: ACM SIGCOM’95, **Proceedings...**, p. 114-125, Cambridge, 1995.

HUANG, J., “Generalizing 4IPP Traffic Model for IEEE 802.16.3”. IEEE 802.16.3c-00/58, Meeting #11, Ottawa, Dec. 2000.

HUANG, P.; FELDMANN, A.; WILLINGER, W., "A Non-intrusive, Wavelet based Approach to Detecting Network Performance Problems". In: ACM SIGCOMM Internet Measurement Workshop, **Proceedings ...**, Nov. 2001.

HURST, H. E. "Methods of Long-Term Storage in Reservoirs". In: Institution Civil Engineers. **Proceedings...**, Part I, p.519-577, 1955.

IMPRISE, "C++Builder documentation". Imprise Corporation, 2000. Disponível em: <http://www.borland.com/techpubs/bcppbuilder>.

JACOBSON, V.; KARELS, M., "Congestion Avoidance and Control". In: ACM SIGCOMM, Symposium on Communication Architectures and Protocols, 1988. **Proceedings...** p. 314-329, Aug. 1988.

JAIN, R., "Congestion Control in Computer Networks: Issues and Trends". *IEEE Network Magazine*, p. 24-30, 1990.

JAIN, R., "The Art of Computer Systems Performance Analysis". Jhon Wiley & Sons, Inc., 1991.

JAIN, R., "Myths about Congestion Management in High Speed Networks". *Internetworking: Res. and Exp.*, v.3, 1992.

JAGERMAN, D. L.; MELAMED, B.; WILLINGER, W., "Stochastic Modeling of Traffic Process". In *Frontiers in Queueing: Models, Methods and Problems*, CRC Press, 1996. Disponível em: <<http://rutcor.rutgers.edu/melamed>>.

JOHNSON, N. L.; KOTZ, S.; BALAKRISHNAM, N., "Continuous Univariate Distributions". v.1, John Wiley & Sons, 1994.

KAPLAN, L. M.; KUO, C. J. "Fractal Estimation from Noisy Data via Discrete Fractional Gaussian Noise (DFGN) and the Haar Basis". *IEEE Transactions on Information Theory*, v.41, n.3, p.3554-3562, 1993.

KELLY, F. P., "Mathematical Modeling of the Internet". In: 4th International Congress on Industrial and Applied Mathematics, **Proceedings ...**, July 1999. Disponível em: <http://www.statslab.cam.ac.uk/~frank/mmi.html>.

KLEINROCK, L., "Queueing Systems". v.1, John Wiley & Sons, 1975.

KUNNIYUR, S.; SRIKANT, R., “End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks”. In: IEEE Infocom, **Proceedings ...**, Mar. 2000. Disponível em: <http://www.ieee-infocom.org/2000/papers/401.ps>.

LEFELHOCZ, C.; LYLES, B.; SHENKER, S.; ZHANG, L., “Congestion Control for Best-Effort Service: Why We Need a New Paradigm”. *IEEE Network*, v. 10, p. 40-48, Jan. 1996.

LELAND, W.; TAQQU, M. S.; WILLINGER, W.; WILSON, D., “On the Self-Similar Nature of Ethernet Traffic”. In: ACM SIGCOMM’93, **Proceedings ...**, p.183-193, 1993.

LELAND, W. E.; TAQQU, M. S.; WILLINGER, W.; WILSON, D. V., ”On the Self-Similar Nature of Ethernet Traffic (Extended Version)”. *IEEE/ACM Transactions on Networking*, v.2, n.1, p.1-15, Feb. 1994.

LEWIS, B., “The Imager Wavelet Library”. 1995. Disponível em: <http://www.cs.ubc.ca/nest/imager/contributions/bobl/wvlt/top.html>.

LI, T.; REKHTER, Y., “A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)”. Request for Comments 2430, IETF, Oct. 1998.

LIKHANOV, N.; TSYBAKOV, B.; GEORGANAS, N., “Analysis of an ATM Buffer with Self-Similar (‘fractal’) Input Traffic”. In: IEEE INFOCOM’95, **Proceedings ...**, p. 985-992, 1995.

LIN, D.; MORRIS, R. “Dynamics of Random Early Detection”. In: SIGCOMM’97. **Proceedings...**, p. 127-137, Cannes, France, Sept. 1997.

LOW, S. H.; LAPSLEY, D. E., “A Duality Model of TCP Flow Controls”. In: ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, **Proceedings ...**, p. 18-20, Sept. 2000. Disponível em: <http://netlab.caltech.edu>.

MA, S.; JI, C. “Modeling Heterogeneous Network Traffic in Wavelet Domain”. *IEEE/ACM Transactions on Networking*, v. 9, n. 5, 2001.

MAGLARIS, B.; ANASTASSION, D.; SEM, P.; KARLSSON. G.; ROBBINS, J., “Performance Models of Statistical Multiplexing in Packet Video Communications”. *IEEE Transactions on Communication*, v. 36, p. 834-844, July, 1988.

MALLAT, S. G., “A Theory for Multiresolution Signal Decomposition: the Wavelet Representation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 11, n. 7, p. 674-693, 1989.

MANDELBROT, B. B. “Long-run linearity, locally gaussian process, h-spectra and infinite variances”. *International Economic Review*, v.10, p.82-113, 1969.

MANDELBROT, B. B. “The Fractal Geometry of Nature”. W. H. Freeman, New York, 1982.

MANDELBROT, B. B.; VAN NESS, H. W., “Fractional Brownian Motions, Fractional Noises and Applications”. *SIAM Review*, v. 10, p. 422-436, Oct. 1968.

MASCOLO, S., “Congestion Control in High-Speed Communication Networks using The Smith Principle”. *Automatica*, Dec. 1999.

MASRY, E., “The Wavelet Transform of Stochastic Process with Stationary Increments and Its Application to Fractional Brownian Motion”. *IEEE Transaction on Information Theory*, v.39, n.1, p.260-264, 1993.

MAY, M.; BONALD, T.; BOLOT, J., “Analytic Evaluation of RED Performance”. In: IEEE INFOCOM, **Proceedings...**, Mar. 2000.

MELAMED, B., “An Overview of TES Process and Modeling Methodology”. *Performance Evaluation of Computer and Communication Systems*, Springer-Verlag, p.359-393, 1993.

MISRA, V.; GONG, W-B.; TOWSLEY, D., “Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED”. In: ACM SIGCOMM, **Proceedings ...**, 2000.

MO, J.; WALRAND, J., “Fair End-toEnd Window-based Congestion Control”. *IEEE/ACM Transactions on Networking*, v.8, n.5, p.556-567, Oct. 2000.

MORETTIN, P.A. “Wavelets in Statistics”. *Reviews of The Institute of Mathematics and Statistics of The University of Sao Paulo*, v.3 n.2, p.211-272, 1997.

MORETTIN, P. A.; TOLOI, C. M. C., “Previsão de Séries temporais”. Editora Atual, 1985.

MORETTIN, P. A.; TOLOI, C. M. C., “Séries temporais”. Editora Atual, 136 p., 1987.

NAGARAJAN, R.; KUROSE, J. F.; TOWSLEY, D., “Approximation Techniques for Computing Packet Loss in Finite-Buffered Voice Multiplexers”. *IEEE Journal on Selected Areas in Communications*, v.9, p.369-377, Apr. 1991.

NORROS, I., “A Storage Model with Self-Similar Input”. *Queueing Systems*, v. 16, p. 387-396, 1994.

NORROS, I., "On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks". *IEEE Journal on Selected Areas in Communications*, v.13, p.953-962, Aug. 1995.

OZBAY, H.; KALYANARAMAN, S.; IFTAR, A., "On Rate-based Congestion Control in High-Speed Networks: Design of an h_{∞} based Flow Controller for Single Bottleneck". In: American Control Conference, **Proceedings ...**, 1998.

PARK, K.; KIM, G.; CROVELLA, M., "On the Relationship between File Sizes, Transport Protocols, and Self-Similar Network Traffic". In: IEEE International Conference on Network Protocols, **Proceedings...**, p.171-180, 1996.

PARK, K.; WILLINGER, W., "Self-Similar Network Traffic and Performance Evaluation". John Wiley & Sons, Inc., 2000.

PARTRIDGE, C.; CARVEY, P. P.; KOHLER, E.; WINTERBLE, S., "A 50Gb/s IP Router". *IEEE/ACM Trans. on Networking*, v. 6, n. 3, p. 237-248, Jun 1998.

PAXSON, V., "Fast Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic". *ACM SIGCOMM Computer Communication Review*, v. 27, n. 5, Oct. 1997.

PAXSON, V.; ALLMAN, M.; DAWSON, S.; FENNER, W.; GRINER, J.; HEAVENS, I.; LAHEY, K.; SEMKE, J.; VOLZ B., "Known TCP Implementation Problems". Internet Request for Comments 2525, IETF, Mar. 1999.

PAXSON V.; FLOYD S., "Wide-Area Traffic: The Failure of the Poisson Modeling". *IEEE/ACM Transaction on Networking*, v. 3 n. 3 p. 226-244, June 1995.

PELTIER, R.; LÉVY-VÉHEL, J., "Multifractional Brownian Motion: Definition and Preliminary Results". Technical report, INRIA 2645, 1997.

PESQUET-POPESCU, B. "Statistical Properties of The Wavelet Decomposition of Some Non-Gaussian Self-Similar Processes". Invited paper, *Signal Processing* v.75, n.3, 1999.

QIAO, Y.; SKICEWICZ, J.; DINDA, P., "Multiscale Predictability of Network Traffic". Technical Report TR NWU-CS-02-13, Northwestern University, Evanston, IL, 2003.

RAMAKRISHNAN, K. K.; FLOYD, S., "A Proposal to Add Explicit Congestion Notification (ECN) to IP". Request for Comments 2481, Jan. 1999.

RAMAKRISHNAN, K. K.; FLOYD, S.; BLACK, D., "The Addition of Explicit Congestion Notification (ECN) to IP". Internet draft, arquivo draft-ietf-tsvwg-ecn-03.txt, Mar. 2001.

REISSELL, "Multiresolution Geometric Algorithms Using Wavelets I: Representation for Parametric Curves and Surfaces", UBC TR 93-17, p. 33, 1993.

RIEDI, R.; CROUSE, M.; RIBEIRO, V. J.; BARANIUK, R. G. "A Multifractal Wavelet Model with Application to Network Traffic". *IEEE Transaction on Information Theory*, Special issue on Multiscale Statistical Signal Analysis and Its Applications, v.45, n.3, p.992-1018, 1999.

ROSOLEN, V.; BONAVENTURE, Q.; LEDUC, G., "A RED Discard Strategy for ATM Networks and its Performance Evaluation with TCP/IP Traffic". *ACM Computer Communication Review*, v.29, n.3, July 1999.

ROSS, S. M., "Introduction to Probability Models". Academic Press Inc., 1985.

ROUGHAN, M.; VEITCH, D.; ABRY, P., "On-Line Estimation of the Parameters of Long-Range Dependence". In: Globecom'98, **Proceedings ...**, v. 6, p. 3716-3721, Nov. 1998.

SALIN, H. J.; AHMED, U., "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks". Request for Comments 2884, July 2000.

SILVA, J. L. C.; CAMPOS, M. A.; CUNHA, P. R. F., "Modelagem Estocástica em Redes de Comunicação". In: XXI Simpósio Brasileiro de Telecomunicações, **Anais...** p.13, Fortaleza, CE, set. 2001b.

SILVA, J. L. C.; CAMPOS, M. A., "Modelagem Matemática e Tráfego Auto-Similar". In: 1º Encontro de Matemática Aplicada e Computacional, **Anais...** p. 14, Recife, PE, ago. 2001c.

SILVA, J. L. C.; CAMPOS, M. A.; SOUZA, J. N.; CUNHA, P. R. F., "Análise Estocástica de Descarte de Pacotes em Redes TCP". In: 20º Simpósio Brasileiro de Redes de Computadores, **Anais...** V. 1, p.358-373, Búzios, RJ, maio 2002a.

SILVA, J. L. C.; CAMPOS, M. A.; SOUZA, J. N.; CUNHA, P. R. F., "Packets Discard in TCP Networks". In: International Symposium on Performance Evaluation of Computer and Telecommunication Systems, **Proceedings...** V. 6, p. 494-501, San Diego, California, July 2002b.

SILVA, J. L. C.; CAMPOS, M. A.; CUNHA, P. R. F., “Aplicando Cadeias de Markov na Descrição Comportamental de Algoritmos do Protocolo TCP”. In: 2º Encontro Regional de Matemática Aplicada e Computacional, **Anais...** p. 23-24, Natal, RGN, ago. 2002c.

SILVA, J. L. C.; CAMPOS, M. A.; CUNHA, P. R. F., “Modelos Estocásticos de Previsão para Análise de Fluxo de Tráfego em Redes Multimídia”. In: XIV Congresso Nacional de Matemática Aplicada e Computacional, **Anais...** Nova Friburgo, RJ, set. 2002d.

SILVA, J. L. C.; CAMPOS, M. A.; CUNHA, P. R. F., “Aplicação da Transformada de Wavelet Discreta no Prognóstico de Congestionamento de Tráfego de Redes”. In: XV Congresso Nacional de Matemática Aplicada e Computacional, **Anais...**, Rio Preto, SP, set. 2003.

SILVA, J. L. C.; CAMPOS, M. A.; CUNHA, P. R. F., “Enterprise Network Optimization through Traffic Congestion Detection using Wavelet”. Submetido ao periódico IEEE – Journal on Selected Areas Communication, abril 2004a.

SILVA, J. L. C.; CUNHA, P. R. F. “Modelos de Implementação de Gerência de Redes baseados na Arquitetura de Redes Ativas e no Conceito de Delegação de Autoridade”. In: 5º Workshop TMN (18º SBRC). **Anais...** Belo Horizonte, MG, maio 2000.

SILVA, J. L. C.; CUNHA, P. R. F., “Implementation Models of Networks Management based in the Active Networks Architecture and in the Authority Delegation Concept”. In: IEEE International Conference on Telecommunications, **Proceedings...** V. 3, p. 615-620, Bucharest, Romenia, June 2001a.

SRIKANT, R., “Control of Communication Networks”. In: Tariq Samad, editor, *Perspectives in Control Engineering; Technologies, Applications, New Directions*, p.462-488, 2000.

STALLINGS, W., “Viewpoints: Self-similarity upsets data traffic assumptions”. IEEE Spectrum. p.28-29, Jan. 1997.

SWELDENS, W., “The lifting scheme: a construction of second generation wavelets”. *SIAM Journal on Mathematical Analysis*, v. 29, n. 2, p. 511–546, 1997.

TAQQU, M.S.; LEVY, J.B., “Using Renewal Process to Generate Long-Range Dependence and High Variability”. In: M. S. Taqqu, editor, *Progress in Probability and Statistics*, v.11, Birkhauser, Boston, 1996.

TAQQU, M.S.; TEVEROVSKY, V., “On Estimating the Intensity of Long-Range Dependence in Finite and Infinite Variance Time Series. In: R.J. Adler, R.E. Feldman, and M.S. Taqqu, editors, *A Practical Guide to Heavy Tails*, p. 177-217. Birkhauser, 1998.

TAQQU, M. S.; TEVEROVSKY, V.; WILLINGER, W., “Estimators for Long-Range Dependence: an Empirical Study”. *Fractals*, v.3, p.785-798, 1995.

TAQQU, M. S.; WILLINGER, W.; SHERMAN, R., “Proof of a Fundamental Result in Self-similar Traffic Modeling”. *Computer Communication Review*, n.27, p.5-23, 1997.

THOMSON, K.; MILLER, G. J.; WILDER, R., “Wide-Area Internet Traffic Patterns and Characteristics”. *IEEE Network*, Nov. 1997.

TSYBAKOV, B.; GEORGANAS, N. D., “Self-Similar Traffic and Upper Bounds to Buffer Overflow in an ATM Queue”. *Performance Evaluation*, v.36, n.1, p.57-80, 1998.

TUAN, T.; PARK, K., “Multiple Time Scale Congestion Control for Self-Similar Network Traffic”. *Performance Evaluation*, v.36, p.359-386, 1999.

TUAN, T.; PARK, K., “Multiple Time Scale Redundancy Control for QoS-Sensitive Transport of Real-Time Traffic”. In: IEEE INFOCOM, **Proceedings...**, p.1683-1692, 2000.

VIDAKOVIC, B.; MÜLLER, P. “Wavelets for Kids: A Tutorial Introduction”. Technical Report, Duke University, 1998.

WANG, Z. “Internet QoS: Architectures and Mechanisms for Quality of Service”. Morgan Kaufmann Publishers, 2001.

WHEELWRIGHT, S. C.; MAKRIDAKIS, S., “Forecasting Methods for Management”. 4th edition, New York, John Wiley & Sons Inc, 1985.

WIDRNER, J.; DENDA, D.; MAUVE, M., “A Survey on TCP-Friendly Congestion Control”. *IEEE Network*, v. 15, n. 3, p. 28-37, 2001. Disponível em: <http://citeseer.nj.nec.com/widrner01survey.html>.

WILLINGER, W.; PAXSON, V., “Where Mathematics Meets the Internet”. *Notices of the AMS*, v. 45, p.961-970, Sept. 1998a.

WILLINGER, W.; PAXSON, V., “Heavy Tail Modeling and Teletraffic Data”. Discussion by S. I. Resnick, *Ann. Statistics*, v.25, p.1856-1866, 1998b.

WILLINGER, W.; TAQQU, M. S.; SHERMAN, R.; WILSON, D. V., "Self-Similarity Through High-Variability : Statistical Analysis of Ethernet LAN Traffic at the Source Level". *IEEE/ACM Transactions on Networking*, v.5, n.1, p. 71-76, Feb. 1997.

YANG, C. Q.; REDDY, A. V. S., "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks". *IEEE Network Magazine*, July 1995.

ZHANG, X.; SHIN, K. G., "Second-Order Rate-Control Based Transport Protocols". In: IEEE ICNP2001, **Proceedings...**, 2001.

ZHANG, X.; SHIN, K. G.; SAHA, D.; KANDLUR, D., "Scalable flow control for multicast ABR service in ATM networks". *IEEE/ACM Transactions on Networking*, 2002.

Anexos

Anexo 1

Minicurso e livro publicado na XIX Jornada de Atualização e Informática (JAI) do XXI Congresso da Sociedade Brasileira de Computação.

Modelagem Estocástica de Tráfego em redes de Alta Velocidade

Marcília A. Campos, Jorge L. de C. e Silva, Paulo R. F. Cunha

Centro de Informática – Universidade Federal de Pernambuco (UFPE)

Caixa Postal 7851 – 50732-970 – Recife – PE – Brasil

{mac,jlcs,prfc}@cin.ufpe.Br

Abstract – *The coming of the high-speed networks and of the multimedia applications, with characteristics different from the traditional applications, it propitiated the use of new distribution models of network traffic. These new models have the capacity to cover statistics that better represent the characteristics observed in the nultimedia traffic. Recent studies show that packet trafic in modern networks is strongly auto-correlated, presents a long-range dependency and could be modelled by self-similar processes. The study of the sel-similarity and other processes that model traffics of long-range and short-range dependency have been objective of big attention in the last years.*

Resumo – *O advento das redes de alta velocidade e das aplicações multimídia, com características diferentes das aplicações tradicionais, ensejou o uso de novos modelos de distribuição de tráfego de rede. Estes novos modelos têm a capacidade de cobrir estatísticas que representam melhor as características observadas no tráfego multimídia. Estudos recentes mostram que o tráfego de pacotes em modernas redes é fortemente autocorrelacionado, apresentam dependência de longa duração e podem ser modelados por processos auto-similares. O estudo da auto-similaridade e de outros processos que modelam tráfegos de longa e de curta duração têm sido alvo de grande atenção nos últimos anos.*

Anexo 2

Artigo publicado no SPECS 2002 - International Symposium on Performance Evaluation of Computer and Telecommunication Systems.

Packets Discard in TCP Networks

Jorge L. de C. e Silva Marcilia A.Campos José N. de Souza Paulo R. Freire Cunha
jlcs@cin.ufpe.br mac@cin.ufpe.br neuman@ufc.br prfc@cin.ufpe.br

Federal University of Pernambuco
Av. Prof. Luiz Freire s/n - Cidade Universitária
Caixa Postal 7851 CEP 50732-970
Recife – PE – Brazil
Fax: +55 021 81 3271-8438

Abstract:

Self-similarity is present in traffic of computer networks. The "non Poissonian" behavior which influences the performance of networks and systems was estimated under the assumption of a traffic that observes the Poisson model. The TCP protocol has mechanisms that reduce the traffic congestion and, consequently, increases the network performance. The typical type of packets losses occurrence in TCP networks is an indicator that it can be used in control mechanisms for detecting congestion. The loss of packets can present a behavior which different from the pattern presented by the network traffic flow. This work investigates the behavior of packets which are discarded in TCP networks.

Keywords: self-similarity, performance evaluation, congestion control, traffic shapping, traffic characteristics.

Anexo 3

Artigo publicado no SBRC 2002 - XX Simpósio Brasileiro de Redes de Computadores.

Análise Estocástica de Descarte de Pacotes em Redes TCP

Jorge L. de C. e Silva, Marcilia A. Campos, Paulo R. F. Cunha, José N. de Souza
jlcs@cin.ufpe.br mac@cin.ufpe.br prfc@cin.ufpe.br neuman@ufc.br

Universidade Federal de Pernambuco
Av. Prof. Luiz Freire, s/n – Cidade Universitária
Caixa postal 7851 – CEP 50732-970
Recife - PE

Resumo:

Auto-similaridade está presente no tráfego das redes de computadores. Este comportamento “não Poissoniano” influi no desempenho das redes e sistemas que foram projetados na suposição de um tráfego ajustado ao modelo de Poisson. O protocolo TCP tem mecanismos que reduzem o congestionamento do tráfego e, conseqüentemente, aumentam o desempenho das redes. As ocorrências de perdas de pacotes em redes TCP são indicativos para os mecanismos de controle detectarem congestionamento. Essas perdas de pacotes podem apresentar comportamento diferente do padrão de fluxo do tráfego da rede. Este trabalho investiga o comportamento dos pacotes descartados em redes TCP.

Palavras-chave: auto-similaridade, avaliação de desempenho, controle de congestionamento.

Abstract:

Self-similarity is present in traffic of computers networks. The "non Poissonian" behavior that influences the performance of networks and systems was estimated under the assumption of a traffic that observes the Poisson model. The TCP protocol has mechanisms that reduce the traffic congestion and, consequently, increases the network performance. The typical type of occurrence of packets losses in TCP networks is an indicator that it can be used in control mechanisms for detecting congestion. The loss of packets can present a behavior which different from the pattern presented by the network traffic flow. This work investigates the behavior of packets which are discarded in TCP networks.

Keywords: self-similarity, performance evaluation, congestion control.

Anexo 4

Artigo submetido ao IEEE – Journal on Selected Areas in Communication, em abril 2004.

Enterprise Network Optimization through Traffic Congestion Detection using Wavelet

Jorge L. C. Silva¹ José N. de Souza² Marcilia A. Campos¹ Paulo R. F. Cunha¹

Computer Science Center - Federal University of Pernambuco (UFPE)

P. Box 7851 - 50740-540 - Recife - PE - Brazil

¹{jlcs, mac, prfc}@cin.ufpe.br

²neuman@ufc.br

Abstract - *Enterprise network has showed a traffic behavior different from the commonly assumed in the traffic stochastic modeling. This new pattern of observed behavior motivates the use of self-similar stochastic processes for traffic modeling. The current congestion control algorithms are reactive and based only on the packets discard. There are preventive schemes of congestion control that treats the problem at the output queues of the network nodes using actives queues management mechanisms. However, they do not avoid the use of the discard policies. This work proposes and implements a mechanism for detecting possible congestion cases. The main obtained result was that the mechanism detected possible congestion cases for all the simulated traffic series.*

Keywords: *wavelet, traffic congestion, prediction, self-similarity.*

Anexo 5

Código fonte do mecanismo ProCon.

```

//***** wxfrm.cpp *****
//* Author: Jorge Luiz de Castro e Silva *
//* Universidade Federal de Pernambuco *
//* http://www.cin.ufpe.br/~jlcs *
//* Email: jlcs@cin.ufpe.br *
//*-----*
//* Programa: Prognostico de Congestionamento de Redes de COmputadores *
//*****
//*
//* Politica de Limiar *
//* Hard = 0 *
//* Soft = 1 *
//*
//* Tipo de Limiar *
//* Universal-1 = 0 *
//* Quantile = 1 (necessario informar percentual) *
//* Universal-2 = 2 *
//*****
//*
//* Como executar o programa Progl.cpp *
//* progl (optional <args>) <serie name> <serie transf.> <estatisticos> *
//* os argumentos sao: -b <wavelet base> -p <politica> -t <limiar> *
//* -j <tam janela> -m <trafego medio> -? (help) *
//*
//*****

//----- Inclusão de arquivos -----
#include <stdio.h> // para printf() e feof()
#include <stdlib.h> // para exit() e atof()
#include <math.h> // para pow()
#include <iostream.h>
#include <dos.h>
#include <time.h>
#include "local.h"
#include "quicksort.h"
#include <sys\timeb.h> // para ftime() e struct timeb

//----- Funções -----
void escolhe_base();
static void usage _PROTO((void));
inline void doit();
void estatistic();
double calcula_thresh(int tipo, int perc);
void aplica_thresh(double thresh, int politica, int tipo, int perc);
int sinal(double s);
double calcula_variancia();
double calcula_rs(double mom1, double mom2, long int m, double *vet);

//----- Variaveis globlais -----
double *Z, *X1, *Z1, *ZT; // Vetor de serie temporal
double *x, *y;
static FILE *fpIn = NULL;
static FILE *arqsaida = NULL;
static FILE *arqout = NULL;
static FILE *arqcoef = NULL;

```

```

static FILE *arqcfil = NULL;
static FILE *arqmed = NULL;
char *file_fpIn = "trafIn.txt";
char *file_fpOut = "trafOut.txt";
char *file_estat = "estat.txt";
static char *progname = NULL;
static bool isFwd = TRUE;
static int n = -1;
static bool isArred = FALSE;
static waveletfilter *wfltr = &wfltrHaar;
static waveletfilter *wfltr_menor;
static waveletfilter wfltrl;
static unsigned int windsize = 512;
double thresh = 0.0, limiar_cong = 0.0;
int tipo = 0, politica = 2, perc = 50;

/*****
/* Programa principal
*****/

void main(int argc, char **argv)
{
    struct timeb start, stop;          // estrutura dos tempos start e stop
    double elapsed;                   // Elapsed time in seconds

    // Variaveis locais para calculo da transformada
    unsigned int ini, fim;
    int ch, nPoss, i, r, perc_base;
    extern char *optarg;
    extern int optind;
    bool ok = TRUE, base = FALSE, flag_c = FALSE, grava_coef = FALSE;
    double xNew, mom1, mom2;
    double Hurst = (double) 0.0, rs_value = (double) 0.0;
    char *nome_pol = "sem politica";
    char *nome_limiar = " ";
    long int acum_pos, inicial, final;
    struct time tempol, tempo2;

    progname = argv[0];
    while ((ch = getopt(argc, argv, "cb:p:t:j:m:?")) != -1) {
        switch (ch) {

            case 'b':
                wfltr = wfltr_select(optarg, &nPoss);
                if (wfltr == NULL) {
                    if (nPoss == 0) {
                        (void) fprintf(stderr, "erro: Nenhuma base esta\
                            casando com a base fornecida \"%s\"\n", optarg);
                    } else {
                        (void) fprintf(stderr, "erro: %d base encontrada\
                            \"%s\"\n", nPoss, optarg);
                    }
                }
                ok = FALSE;
            }
            base = TRUE;
            break;

            case 'c':
                grava_coef = TRUE;
                file_fpOut = "nulol";

```

```

        file_estat = "nulo2";
        break;

    case 'p':
        if (STR_EQ(optarg, "hard"))
            politica = 0;
        else if (STR_EQ(optarg, "soft"))
            politica = 1;
        else if (STR_EQ(optarg, "sem"))
            politica = 2;
        else {
            (void) fprintf(stderr, "erro: política
                                inexistente \"%s\"\n", optarg);
            ok = FALSE;
        }
        nome_pol = optarg;
        break;

    case 't':
        if (STR_EQ(optarg, "universal1"))
            tipo = 0;
        else if (STR_EQ(optarg, "quantil"))
            tipo = 1;
        else if (STR_EQ(optarg, "universal2"))
            tipo = 2;
        else {
            (void) fprintf(stderr, "erro: tipo de limiar
                                inexistente \"%s\"\n", optarg);
            ok = FALSE;
        }
        nome_limiar = optarg;
        if ((tipo == 1) && (argv[optind][0] != '-')) {
            perc = atoi(argv[optind]);
            optind++;
        }
        break;

    case 'j':
        windsize = atoi(optarg);        // Recebe o tamanho da janela
        break;

    case 'm':
        limiar_cong = atof(optarg);
        break;

    case '?':
        usage();
        exit(0);
    }
}

if (!ok) {
    usage();
    exit(1);
}

if (! argv[optind] == '\0')
    file_fpIn = argv[optind];
fpIn = fopen(file_fpIn, "r");
if (fpIn == NULL) {
    (void) fprintf(stderr, "*** ERRO ao abrir serie de entrada

```

```

        \"%s\"\n", argv[optind]);
    exit(1);
}
optind++;
if (! argv[optind] == '\0')
    file_fpOut = argv[optind];
arqsaida = fopen(file_fpOut, "w");
if (arqsaida == NULL) {
    cout << "*** ERRO ao abrir serie transformada" << file_fpOut <<
        endl;
    fclose(fpIn);
    exit(1);
}
optind++;
if (! argv[optind] == '\0')
    file_estat = argv[optind];
arqout = fopen(file_estat, "w");
if (arqout == NULL) {
    cout << "*** ERRO ao abrir o arquivo de estatisticas" <<
        file_estat << endl;
    fclose(fpIn);
    fclose(arqsaida);
    exit(1);
}
if (grava_coef){
    arqcoef = fopen("coef.txt", "w");
    arqcfil = fopen("cfil.txt", "w");
}

// ---- Leitura da serie original ----
X1 = NULL;
n = 0;
while (fscanf(fpIn, "%lf", &xNew) == 1) {
    VOID_MR_ALLOC_LINTOK(X1, n+1, double);
    X1[n++] = xNew;
}
if (n < windsize) {
    cout << "tamanho do arquivo eh menor que o tamanho da janela\n";
    exit(1);
}

// ---- Grava as estatisticas da serie original ----
fprintf(arqout, "                PROGNOSTICO DE CONGESTIONAMENTO\n");
fprintf(arqout, "Politica de filtragem e\' %s e o limiar e\' %s",
        nome_pol, nome_limiar);
if (tipo == 1)
    fprintf(arqout, " com %2d%\n", perc);
else
    fprintf(arqout, "\n");
for (i=0; i<n; i++){
    mom1 = mom1 + (X1[i] / n);           // calcula media
    mom2 = mom2 + (pow(X1[i], 2.0) / n); // calcula variancia
}
if (limiar_cong == (double) 0.0)
    limiar_cong = mom1;

// ---- Grava arquivo com o valor do congestionamento ----
arqmed = fopen("media.dat", "w");
for (i=1; i<=n; i++) {
    fprintf(arqmed, "%d %7.2lf", i, limiar_cong);
}

```

```

        if (i != n)
            fprintf(arqmed, "\n");
    }
fclose(arqmed);

fprintf(arqout, "Tamanho da janela = %4d, Limiar de congestionamento
              = %6.2lf\n", windsize, limiar_cong);
// Calcula valor R/S para serie original de tamanho n
rs_value = calcula_rs(mom1, mom2, n, X1);
if (rs_value > (double) 0.0)
    Hurst = log10(rs_value) / (log10(n) - log10(2.0));
fprintf(arqout, "\n      ----- SERIE ORIGINAL      ----- \n");
fprintf(arqout, "Estatistica de R/S = %7.2lf Numero de observacoes =
              %d\n", rs_value, n);
if (Hurst > (double) 0.0)
    fprintf(arqout, "Parâmetro de Hurst = %4.2lf\n", Hurst);
else
    fprintf(arqout, "Parâmetro de Hurst = nao informado\n");
fprintf(arqout, "Media          = %9.2lf\n", mom1);
fprintf(arqout, "Variancia       = %9.2lf\n", mom2 - pow(mom1, 2.0));

if (! base) {
    escolhe_base();
    wfltr = wfltr_menor;
}
wfltr1 = *wfltr;
fprintf(arqout, "Base utilizada na transformacao: %s\n", wfltr1.name);

/*****
/***** Rotina principal *****/
/*****
x = dald_new(windsize);
Z = dald_new(windsize);
ZT = dald_new(windsize/4);
if (tipo == 1)
    Z1 = dald_new(windsize);
ini = 0;
fim = windsize;
while (n >= fim) {
    gettime(&tempo1);
    ftime(&start);
    for (i=0; i<windsize; i+=1) {
        x[i] = X1[i+ini];
    }
    isFwd = TRUE;
    isArred = FALSE;
    doit();
    /*****
    /* Grava coeficientes de wavelet - Introduzido em 06/02/2004
    /*****
    if (grava_coef) {
        for (i=0; i<windsize; i+=1) {
            if ((i == windsize-1) && (n == fim))
                fprintf(arqcoef, "%7.2lf", Z[i]);
            else
                fprintf(arqcoef, "%7.2lf\n", Z[i]);
        }
    }
}

```

```

//*****

if (!(politica == 2)) {
    if (tipo == 1) {
        for (i=0; i<windsize; i++){
            Z1[i] = fabs(Z[i]);
        }
        r = windsize-1;
        QuickSort(0, r);
        perc_base = iround((perc * windsize) / 100);
        if (perc_base < 1)
            perc_base = 1;
    }
    thresh = calcula_thresh(tipo, perc_base);
    aplica_thresh(thresh, politica, tipo, perc_base);
}
//*****
/* Grava coeficientes filtrados - Introduzido em 06/02/2004
//*****
if (grava_coef) {
    for (i=0; i<windsize; i+=1) {
        if ((i == windsize-1) && (n == fim))
            fprintf(arqcfil, "%7.2lf", Z[i]);

        else
            fprintf(arqcfil, "%7.2lf\n", Z[i]);
    }
}
//*****

isFwd = FALSE;
isArred = TRUE;
for (i=0; i<windsize; i++){
    x[i] = Z[i];
}
doit();
for (i=0; i<windsize; i+=1) {
    if ((i == windsize-1) && (n == fim))
        fprintf(arqsaida, "%7.2lf", Z[i]);
    else
        fprintf(arqsaida, "%7.2lf\n", Z[i]);
}

/* Grava informacoes sobre o tempo de processamento e as
indicacoes de congestionamento
fprintf(arqout, "\n      -----      %2da JANELA RECONSTRUIDA      ----
\n", fim / windsize);
fprintf(arqout, "-----\n");

/* Teste de Cox-Stuart ou teste do sinal
if (ini > 0) {
    acum_pos = 0;
    for (i=0; i<windsize/4; i++) {
        if ((ZT[i] < Z[i]) && ((double)Z[i] > limiar_cong))
            acum_pos++;
    }
    if (((float)acum_pos / (float) (windsize/4)) > (((float)0.5 *
mom1) / limiar_cong)){
        fprintf(arqout, "Possivel congestionamento nas observações

```

```

        de %d a %d\n", ini+1, ini+windsize/4);
    flag_c = TRUE;
}
}
acum_pos = 0;
inicial = 0;
final = (windsize / 2) - 1;
while (final <= windsize) {
    for (i=inicial; i<=(final-(windsize/4)); i++){
        if ((Z[i] < Z[(windsize/4) + i]) &&
            ((double)Z[(windsize/4) + i] > limiar_cong))
            acum_pos++;
    }
    if (((float)acum_pos / (float) (windsize/4)) > (((float)0.5 *
        mom1) / limiar_cong)) {
        fprintf(arqout, "Possivel congestionamento nas observações
            de %d a %d\n",
                ini+inicial+1+(windsize/4), ini+final+1);
        flag_c = TRUE;
    }
    acum_pos = 0;
    inicial = inicial + (windsize/4);
    final = final + (windsize/4);
}
for (i=0; i<windsize/4; i++)
    ZT[i] = Z[i+((windsize/4)*3)];
ftime(&stop);
gettime(&tempo2);
elapsed = ((double) stop.time + ((double) stop.millitm * 0.001)) -
    ((double) start.time + ((double) start.millitm * 0.001));
if (flag_c)
    fprintf(arqout, "-----\n");
flag_c = FALSE;
fprintf(arqout, "Tempo de CPU: %5.3lf segundos\n", elapsed);
fprintf(arqout, "Relogio: iniciou as %2d:%2d:%2d:%2d e terminou as
    %2d:%2d:%2d:%2d\n", (int)tempo1.ti_hour, (int)tempo1.ti_min,
        (int)tempo1.ti_sec, (int)tempo1.ti_hund, (int)tempo2.ti_hour,
        (int)tempo2.ti_min, (int)tempo2.ti_sec, (int)tempo2.ti_hund);
fprintf(arqout, "-----\n");
//estatistic();
ini = fim;
fim += windsize;
}
dald_delete(Z);
dald_delete(x);
dald_delete(ZT);
if (tipo == 1)
    dald_delete(Z1);
FREE_LINTOK(X1);
fclose(fpIn);
fclose(arqsaida);
fclose(arqout);
if (grava_coef){
    fclose(arqcoef);
    fclose(arqcfil);
}
exit(0);
}

```

```

/*****
/* Função que executa a transformada de wavelet
/*****
inline void doit()
{
    int i, lgN;
    double xNew;

    for (lgN = 0; (1 << lgN) < windsize; lgN++){
        continue;
    }
    if ((1 << lgN) != windsize) {
        cout << stderr << " # de observacoes deve ser uma potencia de 2 -
            terminando programa" << endl;
        exit(1);
    }
    y = dald_new(windsize);
    wxfrm_dald(x, windsize, isFwd, wfltr, y);
    // Foi introduzido este teste para fazer arredondamento de valores -
    Julho/2003
    if (isArred){
        for (i = 0; i < windsize; i++) {
            if (!(politica == 2)) {
                Z[i] = iround(y[i]);
                if (Z[i] < 0)
                    Z[i] = 0;
            }else
                Z[i] = fround(y[i], 6);
        }
    }else{
        for (i = 0; i < windsize; i++) {
            Z[i] = y[i];
        }
    }
    dald_delete(y);
    return;
}

/*****
/* Função de estatística
/*****

void estatistic()
{
    // Variaveis locais para calculo estatístico
    double rs_value = (double) 0.0; // Valor R/S calculado
    double Hurst = (double) 0.0; // Valor do parametro de Hurst
    double min, max; // Valores minimo e maximo
    long int minpos, maxpos; // Posicoes de min e max
    double mom1, mom2; // Primeiro e segundo momentos
    double mean; // Media calculada
    double var; // Variancia calculada
    double stddev; // Desvio padrao calculado
    double coef; // Coeficiente de variacao calculado
    int i;

    // Laco para calcular min, max, media e variancia
    min = max = Z[0];
    minpos = maxpos = 0;

```

```

mom1 = mom2 = 0.0;
for (i=0; i<windsize; i++){
    if (Z[i] <= min){
        min = Z[i];
        minpos = i;
    }
    if (Z[i] >= max){
        max = Z[i];
        maxpos = i;
    }
    mom1 = mom1 + (Z[i] / windsize);           // calcula media
    mom2 = mom2 + (pow(Z[i], 2.0) / windsize); // calcula variancia
}

// Calcula valor R/S para serie X de tamanho N
rs_value = calcula_rs(mom1, mom2, windsize, Z);
if (rs_value > (double) 0.0)
    Hurst = log10(rs_value) / (log10(windsize) - log10(2.0));

// Saida do valor R/S
fprintf(arqout, "Estatistica de R/S = %8.2lf Num. de observacoes =
    %d\n", rs_value, windsize);
if (Hurst > (double) 0.0)
    fprintf(arqout, "Parâmetro de Hurst = %4.2lf\n", Hurst);
else
    fprintf(arqout, "Parâmetro de Hurst = nao informado\n");

// Calcula desvio padrao e coeficiente de variacao
mean = mom1;
var = mom2 - pow(mom1, 2.0);                 // calcula variancia
stddev = sqrt(var);
if (mom1 == (double)0.0)
    coef = 0.0;
else
    coef = stddev / mom1;
fprintf(arqout, "Minimo           = %4.0f Posicao do minimo = %d\n",
    min, minpos);
fprintf(arqout, "Maximo           = %4.0f Posicao do maximo = %d\n",
    max, maxpos);
fprintf(arqout, "Amplitude        = %4.0f\n", max - min);
fprintf(arqout, "Media            = %9.2lf\n", mean);
fprintf(arqout, "Variancia         = %9.2lf\n", var);
fprintf(arqout, "Desv padrao      = %9.2lf\n", stddev);
fprintf(arqout, "Coef Variacao    = %9.2lf\n", coef);
return;
}

//*****
/* Função usage *
//*****
static void usage()
{
    int iWfltr, col, colreq;
    int width = 19; /* width of column */
    int mxncol = 4; /* maximum number of columns */
    int pad = 1; /* space required between columns */

    cout << "\nComo chamar o programa:\n";
    cout << "C:\\>procon [{argumentos}] [{nome da serie de entrada}]
        [{nome serie de saida}]\n";
}

```

```

    [{nome do arquivo de estatisticas}]\n" << endl;
cout << " Os argumentos sao:\n";
cout << " -b {w}   usa base de wavelet selecionada pela opção
    {w}. Se o argumento\
    nao for especificado, o programa seleciona a base.\n";
cout << " -c       grava um arquivo contendo os coeficientes de
    wavelet.\n";
cout << " -p {p}   usa uma politica selecionada pela opcao {p}, onde
    p = hard ou soft.\n";
cout << " -t (t)   usa um limiar selecionado pela opcao {t}, onde t =
    universal ou quan-\
    til. Se quantil pede-se o percentual de observacoes (default
    50%).\n";
cout << " -j {j}   usa o tamanho de janela especificada por {j}
    (default 128).\n";
cout << " -m {m}   usa o patamar de congestionamento especificado por
    {m}.\n";
cout << " -?       opcao de ajuda para listar os argumentos do
    programa.\n" << endl;
cout << " Bases incluidas no programa:\n";
col = 0;
(void) fputc(' ', stderr);
(void) fputc(' ', stderr);
for (iWfltr = 0; wfltrList[iWfltr] != NULL; iWfltr++) {
    colreq = (strlen(wfltrList[iWfltr]->name) + pad + width - 1) /
        width;
    if (col + colreq > mxncol) {
        (void) fprintf(stderr, "\n %*s", -colreq * width,
            wfltrList[iWfltr]->name);
        col = colreq;
    } else {
        (void) fprintf(stderr, "%*s", -colreq * width,
            wfltrList[iWfltr]->name);
        col += colreq;
    }
}
(void) fputc('\n', stderr);
(void) fprintf(stderr, "\n OBS: O programa aceita qualquer palavra
    abreviada, por exemplo, daub*4\n");
getchar();
return;
}

/*****
/* Função para calcular o limiar do filtro */
/*****
double calcula_thresh(int tipo, int percent)
{
    double var, thresh = 0.0;

    if (tipo == 0) {
        var = calcula_variancia();
        thresh = sqrt(var) * sqrt(2 * log(windowsize) / windowsize);
    } else if (tipo == 1)
        thresh = Z1[percent-1];
    else if (tipo == 2) {
        var = calcula_variancia();
        thresh = sqrt(var) * sqrt(2 * log(windowsize));
    }
}
return(thresh);

```

```

}

void aplica_thresh(double thresh, int politica, int tipo, int percent)
{
    int i, j=0;

    if (tipo == 1 && thresh > (double) 0.0 ) {
        i = 0;
        while (i < windsize && j < percent) {
            if ((politica == 0) && (-thresh <= Z[i]) && (Z[i] <= thresh)) {
                j++;
                Z[i] = (double) 0.0;
            }
            else {
                if (politica == 1)
                    Z[i] = sinal(Z[i]) * (fabs(Z[i]) - thresh);
            }
            i++;
        }
    }
    if ((tipo == 0 || tipo == 2) && (thresh > (double) 0.0)) {
        for (i=0; i<windsize; i++) {
            if ((politica == 0) && (-thresh <= Z[i]) && (Z[i] <= thresh)) {
                Z[i] = (double) 0.0;
            }
            else {
                if (politica == 1)
                    Z[i] = sinal(Z[i]) * (fabs(Z[i]) - thresh);
            }
        }
    }
    return;
}

int sinal(double s)
{
    if (s < 0)
        return -1;
    else
        return 1;
}

/*****
/* Função para calcular variância da série Z
*****/
double calcula_variancia()
{
    double mean, var;
    long int i;

    mean = var = 0.0;
    for (i=0; i<windsize; i++){
        mean = mean + (Z[i] / windsize);
        var = var + (pow(Z[i], 2.0) / windsize);
    }
    var = var - pow(mean, 2.0);
    return(var);
}

/*****

```

```

/* Funcao para calcular o valor R/S para serie Z de tamanho N
/*****
double calcula_rs(double mom1, double mom2, long int m, double *vet)
{
    double      s;                // Desvio padrao
    double      w;                // Valor W
    double      r;                // Valor R
    double      min_w;            // Valor minimo de W
    double      max_w;            // Valor maximo de W
    double      rs_value;         // Valor de R/S para ser retornado
    double      sum;              // Valor da soma temporaria
    long int    i;                // Contador de laço

    s = sqrt(mom2 - pow(mom1,2));

    // Laço para encontrar os valores minimo e maximo de W
    min_w = max_w = 0.0;
    sum = 0.0;
    for (i=0; i<m; i++) {
        sum = sum + vet[i];
        w = sum - ((i+1) * mom1);          // Fix #1
        if (w > max_w) max_w = w;
        if (w < min_w) min_w = w;
    }

    // Calcula o valor de R como o maximo de W menos o minimo de W
    r = max_w - min_w;

    // Calcula o valor de R/S
    if (r == 0.0 || s == 0.0) {
        rs_value = double(0.0);
    }else {
        rs_value = r / s;
    }
    return(rs_value);
}

/*****
/* ***** Função Escolhe_Base *****
/*****
void escolhe_base()
{
    int j, i, r, perc_base;
    double soma = 0, menor = -1;

    for (j = 0; j<20; j++) {
        wfltr = wfltrList[j];
        wfltr1 = *wfltr;
        x = da1d_new(windowsize);
        Z = da1d_new(windowsize);
        if (tipo == 1)
            Z1 = da1d_new(windowsize);

        for (i=0; i<windowsize; i++) {
            x[i] = X1[i];
        }
        isFwd = TRUE;
        isArred = FALSE;
        doit();
        if (!(politica == 2)) {

```

```

    if (tipo == 1){
        for (i=0; i<windsiz; i++){
            Z1[i] = fabs(Z[i]);
        }
        r = windsiz-1;
        QuickSort(0, r);
        perc_base = iround((perc * windsiz) / 100);
        if (perc_base < 1)
            perc_base = 1;
    }
    thresh = calcula_thresh(tipo, perc_base);
    aplica_thresh(thresh, politica, tipo, perc_base);
}
isFwd = FALSE;
isArred = TRUE;
for (i=0; i<windsiz; i++){
    x[i] = Z[i];
}
doit();
soma = 0;
for (i=0; i<windsiz; i++) {
    soma = soma + pow((X1[i]-Z[i]),2) / windsiz;
}
if (menor == -1) {
    menor = soma;
    wfltr_menor = wfltr;
} else if (soma < menor) {
    menor = soma;
    wfltr_menor = wfltr;
}
}
dald_delete(Z);
dald_delete(x);
if (tipo == 1)
    dald_delete(Z1);
}
}

```

Apêndices

Apêndice A

As definições das bases e filtros usados no cálculo das transformadas de *wavelet* (direta e inversa) estão neste apêndice. A fonte de consulta mais comum para esses filtros é o livro "Ten Lectures on Wavelets" de Daubechies. As *wavelets* ortonormais têm filtros iguais tanto para a transformada direta quanto para a inversa.

Quadro A.1. Filtros da base de AdelsonSimoncelliHingorani

Transformada direta	transformada inversa
$\sqrt{2} * 0.01995$	$\sqrt{2} * 0.01995$
$\sqrt{2} * -0.04271$	$\sqrt{2} * -0.04271$
$\sqrt{2} * -0.05224$	$\sqrt{2} * -0.05224$
$\sqrt{2} * 0.29271$	$\sqrt{2} * 0.29271$
$\sqrt{2} * 0.56458$	$\sqrt{2} * 0.56458$
$\sqrt{2} * 0.29271$	$\sqrt{2} * 0.29271$
$\sqrt{2} * -0.05224$	$\sqrt{2} * -0.05224$
$\sqrt{2} * -0.04271$	$\sqrt{2} * -0.04271$
$\sqrt{2} * 0.01995$	$\sqrt{2} * 0.01995$

Fonte: Adelson, Simoncelli, Hingorani, "Orthogonal Pyramid Transforms for Image Coding", Visual Communications and Image Processing II, Proc. SPIE Vol. 845, pp. 50-58.

Quadro A.2. Filtros da base de AntoniniBarlaudMatheiuDaubechies_4_4

transformada direta	transformada inversa
$\sqrt{2} * 0.026749$	$\sqrt{2} * -0.045636$
$\sqrt{2} * -0.016864$	$\sqrt{2} * -0.028772$
$\sqrt{2} * -0.078223$	$\sqrt{2} * 0.295636$
$\sqrt{2} * 0.266864$	$\sqrt{2} * 0.557543$
$\sqrt{2} * 0.602949$	$\sqrt{2} * 0.295636$
$\sqrt{2} * 0.266864$	$\sqrt{2} * -0.028772$
$\sqrt{2} * -0.078223$	$\sqrt{2} * -0.045636$
$\sqrt{2} * -0.016864$	
$\sqrt{2} * 0.026749$	

Fonte: Antonini, M.; Barlaud, M.; Mathieu, P.; Daubechies, I., "Image Coding Using Wavelet Transforms", IEEE Trans. on Image Processing, Vol. 1 No. 2, April 1992.

Quadro A.3. Filtros da base de BattleLemarie

transformada direta	transformada inversa
$\sqrt{2} * -0.002$	$\sqrt{2} * -0.002$
$\sqrt{2} * -0.003$	$\sqrt{2} * -0.003$
$\sqrt{2} * 0.006$	$\sqrt{2} * 0.006$
$\sqrt{2} * 0.006$	$\sqrt{2} * 0.006$
$\sqrt{2} * -0.013$	$\sqrt{2} * -0.013$
$\sqrt{2} * -0.012$	$\sqrt{2} * -0.012$
$\sqrt{2} * 0.030$	$\sqrt{2} * 0.030$
$\sqrt{2} * 0.023$	$\sqrt{2} * 0.023$
$\sqrt{2} * -0.078$	$\sqrt{2} * -0.078$
$\sqrt{2} * -0.035$	$\sqrt{2} * -0.035$
$\sqrt{2} * 0.307$	$\sqrt{2} * 0.307$
$\sqrt{2} * 0.542$	$\sqrt{2} * 0.542$
$\sqrt{2} * 0.307$	$\sqrt{2} * 0.307$
$\sqrt{2} * -0.035$	$\sqrt{2} * -0.035$
$\sqrt{2} * -0.078$	$\sqrt{2} * -0.078$
$\sqrt{2} * 0.023$	$\sqrt{2} * 0.023$
$\sqrt{2} * 0.030$	$\sqrt{2} * 0.030$
$\sqrt{2} * -0.012$	$\sqrt{2} * -0.012$
$\sqrt{2} * -0.013$	$\sqrt{2} * -0.013$
$\sqrt{2} * 0.006$	$\sqrt{2} * 0.006$
$\sqrt{2} * 0.006$	$\sqrt{2} * 0.006$
$\sqrt{2} * -0.003$	$\sqrt{2} * -0.003$
$\sqrt{2} * -0.002$	$\sqrt{2} * -0.002$

Fonte: Mallat, S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE PAMI, v.11, n.7, 674-693, Table 1.

Quadro A.4. Filtros da base de BurtAdelson

transformada direta	transformada inversa
$\sqrt{2} * -1 / 20$	$\sqrt{2} * -3 / 280$
$\sqrt{2} * 5 / 20$	$\sqrt{2} * -15 / 280$
$\sqrt{2} * 12 / 20$	$\sqrt{2} * 73 / 280$
$\sqrt{2} * 5 / 20$	$\sqrt{2} * 170 / 280$
$\sqrt{2} * -1 / 20$	$\sqrt{2} * 73 / 280$
	$\sqrt{2} * -15 / 280$
	$\sqrt{2} * -3 / 280$

Fonte: Daubechies, I., "Ten Lectures on Wavelets". SIAM, Philadelphia, 1992, Table 8.4.

Quadro A.5. Filtros da base de Coiflet_2

transformada direta	transformada inversa
$\sqrt{2} * (\sqrt{15} - 3) / 32$	$\sqrt{2} * (\sqrt{15} - 3) / 32$
$\sqrt{2} * (1 - \sqrt{15}) / 32$	$\sqrt{2} * (1 - \sqrt{15}) / 32$
$\sqrt{2} * (6 - 2 * \sqrt{15}) / 32$	$\sqrt{2} * (6 - 2 * \sqrt{15}) / 32$
$\sqrt{2} * (2 * \sqrt{15} + 6) / 32$	$\sqrt{2} * (2 * \sqrt{15} + 6) / 32$
$\sqrt{2} * (\sqrt{15} + 13) / 32$	$\sqrt{2} * (\sqrt{15} + 13) / 32$
$\sqrt{2} * (9 - \sqrt{15}) / 32$	$\sqrt{2} * (9 - \sqrt{15}) / 32$

Quadro A.6. Filtros da base de Coiflet_4

transformada direta	transformada inversa
0.0011945726958388	0.0011945726958388
-0.01284557955324	-0.01284557955324
0.024804330519353	0.024804330519353
0.050023519962135	0.050023519962135
-0.15535722285996	-0.15535722285996
-0.071638282295294	-0.071638282295294
0.57046500145033	0.57046500145033
0.75033630585287	0.75033630585287
0.28061165190244	0.28061165190244
-0.0074103835186718	-0.0074103835186718
-0.014611552521451	-0.014611552521451
-0.0013587990591632	-0.0013587990591632

Quadro A.7. Filtros da base de Coiflet_6

transformada direta	transformada inversa
-0.0016918510194918	-0.0016918510194918
-0.00348787621998426	-0.00348787621998426
0.019191160680044	0.019191160680044
0.021671094636352	0.021671094636352
-0.098507213321468	-0.098507213321468
-0.056997424478478	-0.056997424478478
0.45678712217269	0.45678712217269
0.78931940900416	0.78931940900416
0.38055713085151	0.38055713085151
-0.070438748794943	-0.070438748794943
-0.056514193868065	-0.056514193868065
0.036409962612716	0.036409962612716
0.0087601307091635	0.0087601307091635
-0.011194759273835	-0.011194759273835
-0.0019213354141368	-0.0019213354141368
0.0020413809772660	0.0020413809772660
0.00044583039753204	0.00044583039753204
-0.00021625727664696	-0.00021625727664696

Fonte: Beylkin, Coifman, Rokhlin "Fast Wavelet Transforms and Numerical Algorithms I", Comm. Pure Appl. Math, v. 44, Appendix A

Quadro A.8. Filtros da base de Daubechies_4

Transformada direta	transformada inversa
$\sqrt{2} * (1 + \sqrt{3}) / 8$	$\sqrt{2} * (1 + \sqrt{3}) / 8$
$\sqrt{2} * (3 + \sqrt{3}) / 8$	$\sqrt{2} * (3 + \sqrt{3}) / 8$
$\sqrt{2} * (3 - \sqrt{3}) / 8$	$\sqrt{2} * (3 - \sqrt{3}) / 8$
$\sqrt{2} * (1 - \sqrt{3}) / 8$	$\sqrt{2} * (1 - \sqrt{3}) / 8$

Quadro A.9. Filtros da base de Daubechies_6

Transformada direta	transformada inversa
0.332670552950	0.332670552950
0.806891509311	0.806891509311
0.459877502118	0.459877502118
-0.135011020010	-0.135011020010
-0.085441273882	-0.085441273882
0.035226291882	0.035226291882

Quadro A.10. Filtros da base de Daubechies _8

Transformada direta	transformada inversa
0.230377813309	0.230377813309
0.714846570553	0.714846570553
0.6308807667930	0.6308807667930
-0.027983769417	-0.027983769417
-0.187034811719	-0.187034811719
0.030841381836	0.030841381836
0.032883011667	0.032883011667
-0.010597401785	-0.010597401785

Quadro A.11. Filtros da base de Daubechies _10

Transformada direta	transformada inversa
0.1601023979741929	0.1601023979741929
0.6038292697971895	0.6038292697971895
0.7243085284377726	0.7243085284377726
0.1384281459013203	0.1384281459013203
-0.2422948870663823	-0.2422948870663823
-0.0322448695846381	-0.0322448695846381
0.0775714938400459	0.0775714938400459
-0.0062414902127983	-0.0062414902127983
-0.0125807519990820	-0.0125807519990820
0.0033357252854738	0.0033357252854738

Quadro A.12. Filtros da base de Daubechies _12

Transformada direta	transformada inversa
0.1115407433501095	0.1115407433501095
0.4946238903984533	0.4946238903984533
0.7511339080210959	0.7511339080210959
0.3152503517091982	0.3152503517091982
-0.2262646939654400	-0.2262646939654400
-0.1297668675672625	-0.1297668675672625
0.0975016055873225	0.0975016055873225
0.0275228655303053	0.0275228655303053
-0.0315820393184862	-0.0315820393184862
0.0005538422011614	0.0005538422011614
0.0047772575119455	0.0047772575119455
-0.0010773010853085	-0.0010773010853085

Quadro A.13. Filtros da base de Daubechies _20

Transformada direta	transformada inversa
0.026670057901	0.026670057901
0.188176800078	0.188176800078
0.527201188932	0.527201188932
0.688459039454	0.688459039454
0.281172343661	0.281172343661
-0.249846424327	-0.249846424327
-0.195946274377	-0.195946274377
0.127369340336	0.127369340336
0.093057364604	0.093057364604
-0.071394147166	-0.071394147166
-0.029457536822	-0.029457536822
0.033212674059	0.033212674059
0.003606553567	0.003606553567
-0.010733175483	-0.010733175483
0.001395351747	0.001395351747
0.001992405295	0.001992405295
-0.000685856695	-0.000685856695
-0.000116466855	-0.000116466855
0.000093588670	0.000093588670
-0.000013264203	-0.000013264203

Fonte: Daubechies, I., “Ten Lectures on Wavelets”. SIAM, Philadelphia, 1992, Table 6.1.

Quadro A.14. Filtros da base de Haar

Transformada direta	transformada inversa
$\sqrt{2} * 1 / 2$	$\sqrt{2} * 1 / 2$
$\sqrt{2} * 1 / 2$	$\sqrt{2} * 1 / 2$

Fonte: Daubechies, I., “Ten Lectures on Wavelets”. SIAM, Philadelphia, p. 10, 1992.

Quadro A.15. Filtros da base de Pseudocoiflet_4_4

Transformada direta	transformada inversa
$\sqrt{2} * -1 / 512$	$\sqrt{2} * -1 / 32$
0.0	0.0
$\sqrt{2} * 18 / 512$	$\sqrt{2} * 9 / 32$
$\sqrt{2} * -16 / 512$	$\sqrt{2} * 16 / 32$
$\sqrt{2} * -63 / 512$	$\sqrt{2} * 9 / 32$
$\sqrt{2} * 144 / 512$	0.0
$\sqrt{2} * 348 / 512$	$\sqrt{2} * -1 / 32$
$\sqrt{2} * 144 / 512$	
$\sqrt{2} * -63 / 512$	
$\sqrt{2} * -16 / 512$	
$\sqrt{2} * 18 / 512$	
0.0	
$\sqrt{2} * -1 / 512$	

Fonte: Reissell, "Multiresolution Geometric Algorithms Using Wavelets I: Representation for Parametric Curves and Surfaces", UBC TR 93-17, p. 33

Quadro A.16. Filtros da base de Spline_2_2

Transformada direta	transformada inversa
$\sqrt{2} * -0.125$	$\sqrt{2} * 1 / 4$
$\sqrt{2} * 0.25$	$\sqrt{2} * 2 / 4$
$\sqrt{2} * 0.75$	$\sqrt{2} * 1 / 4$
$\sqrt{2} * 0.25$	
$\sqrt{2} * -0.125$	

Quadro A.17. Filtros da base de Spline_2_4

Transformada direta	transformada inversa
$\sqrt{2} * 3 / 128$	$\sqrt{2} * 1 / 4$
$\sqrt{2} * -6 / 128$	$\sqrt{2} * 2 / 4$
$\sqrt{2} * -16 / 128$	$\sqrt{2} * 1 / 4$
$\sqrt{2} * 38 / 128$	
$\sqrt{2} * 90 / 128$	
$\sqrt{2} * 38 / 128$	
$\sqrt{2} * -16 / 128$	
$\sqrt{2} * -6 / 128$	
$\sqrt{2} * 3 / 128$	

Quadro A.18. Filtros da base de Spline_3_3

Transformada direta	transformada inversa
$\sqrt{2} * 3 / 64$	$\sqrt{2} * 1 / 8$
$\sqrt{2} * -9 / 64$	$\sqrt{2} * 3 / 8$
$\sqrt{2} * -7 / 64$	$\sqrt{2} * 3 / 8$
$\sqrt{2} * 45 / 64$	$\sqrt{2} * 1 / 8$
$\sqrt{2} * 45 / 64$	
$\sqrt{2} * -7 / 64$	
$\sqrt{2} * -9 / 64$	
$\sqrt{2} * 3 / 64$	

Quadro A.19. Filtros da base de Spline_3_5

Transformada direta	transformada inversa
$\sqrt{2} * -5 / 512$	$\sqrt{2} * 1 / 8$
$\sqrt{2} * 15 / 512$	$\sqrt{2} * 3 / 8$
$\sqrt{2} * 19 / 512$	$\sqrt{2} * 3 / 8$
$\sqrt{2} * -97 / 512$	$\sqrt{2} * 1 / 8$
$\sqrt{2} * -26 / 512$	
$\sqrt{2} * 350 / 512$	
$\sqrt{2} * 350 / 512$	
$\sqrt{2} * -26 / 512$	
$\sqrt{2} * -97 / 512$	
$\sqrt{2} * 19 / 512$	
$\sqrt{2} * 15 / 512$	
$\sqrt{2} * -5 / 512$	

Quadro A.20. Filtros da base de Spline_3_7

Transformada direta	transformada inversa
$\sqrt{2} * -35 / 16384$	$\sqrt{2} * 1 / 8$
$\sqrt{2} * -105 / 16384$	$\sqrt{2} * 3 / 8$
$\sqrt{2} * -195 / 16384$	$\sqrt{2} * 3 / 8$
$\sqrt{2} * 865 / 16384$	$\sqrt{2} * 1 / 8$
$\sqrt{2} * 363 / 16384$	
$\sqrt{2} * -3489 / 16384$	
$\sqrt{2} * -307 / 16384$	
$\sqrt{2} * 11025 / 16384$	
$\sqrt{2} * 11025 / 16384$	
$\sqrt{2} * -307 / 16384$	
$\sqrt{2} * -3489 / 16384$	
$\sqrt{2} * 363 / 16384$	
$\sqrt{2} * 865 / 16384$	
$\sqrt{2} * -195 / 16384$	
$\sqrt{2} * -105 / 16384$	
$\sqrt{2} * -35 / 16384$	

Fonte: Daubechies, I., "Ten Lectures on Wavelets". SIAM, Philadelphia, 1992, Table 8.2.

Apêndice B

Glossário

Algoritmo:	Conjunto predeterminado e bem definido de regras e processos destinados à solução de um problema, com um número finito de etapas.
Byte:	Termo genérico usado para se referir a uma série de dígitos binários consecutivos que são operados como uma unidade.
Buffer:	Área de armazenamento temporário em um dispositivo onde dados em transito podem ser tratados. Frequentemente é usado para compensar diferenças em velocidades de processamento entre dispositivos da rede. Rajadas de dados podem ser armazenadas em <i>buffers</i> até que possam ser processadas por dispositivos de processamento mais lento.
Datagrama:	Um grupamento lógico de informação enviado como uma unidade da camada de rede sobre um meio de transmissão. Datagramas IP são as unidades primárias de informação na Internet. Os termos pacote, <i>frame</i> , segmento e mensagem são também usados para descrever grupamentos lógicos de informação nas várias camadas do modelo OSI.
Delay:	Tempo requerido para mover um pacote da fonte para o destino sobre um determinada rota (caminho).
Ethernet:	Tecnologia utilizada para comunicação entre computadores em redes locais. Redes ethernet operam a 10Mbps, usando cabo coaxial, e estão padronizadas através da IEEE 802.3.
Host:	É um sistema de computação em uma rede.
IEEE:	(<i>Institute of Electrical and Eletronic Engineers</i>). É uma organização profissional que define padrões de rede
IETF:	(<i>Internet Engineering Task Force</i>). É um órgão ligado ao ISOC (<i>Internet Society</i>) que tem como objetivo estabelecer grupos de trabalho em assuntos diversos da Internet para criar recomendações e padrões publicados através das RFCs.

- Interface:** Uma conexão entre dois sistemas ou dispositivos. Na rede, uma interface é o dispositivo que realiza a conexão a instalação local e a rede.
- ITU-T:** (*International Telecommunications Union – Telecommunications Standardization Sector*). É uma organização internacional que desenvolve padrões de comunicação, tais como a recomendação X.25.
- LAN:** (*Local Area Network*). É uma rede que cobre uma área geográfica pequena, usualmente a área de um edifício ou de um pavimento ou andar.
- Largura de Banda:** (*bandwidth*). É a capacidade de vazão calculada ou velocidade de transmissão de uma rede, meio ou protocolo.
- Melhor-esforço:** (*best-effort*). É uma característica de um sistema de rede que não usa reconhecimentos sofisticados para garantir a entrega confiável da informação.
- Multicast:** A maioria dos endereços IP referem-se a um único destinatário (*unicast*). Quando as conexões são para alguns *hosts*, isto é, se referem a um sub-conjunto de endereços da rede, tem-se a conexão *multicast*. Em contraste, nas conexões *broadcast*, os pacotes são enviados para todos dispositivos da rede.
- NIC (*adapter*):** Uma placa (ou cartão), usualmente instalada dentro de um sistema de computação, que provê comunicação do equipamento na rede.
- Nó ou Nodo:** Termo genérico usado para se referir a uma entidade ou dispositivo que pode acessar uma rede
- Pacote:** Um grupamento lógico de informação que inclui um cabeçalho (*header*) e dados.
- Porta:** Uma *interface* em um dispositivo (como um roteador) de interconexão de redes.
- Protocolo:** Uma descrição formal de um conjunto de regras e convenções que regem a troca de informações entre dispositivos de uma rede.

- Roteador:** Um dispositivo que trabalha com a camada de rede com o objetivo de decidir quais as melhores rotas para o tráfego baseado em alguma métrica. Também chamado de *gateway*, embora este termo seja obsoleto, os roteadores enviam pacotes de uma rede para outra baseados na informação da camada de rede
- Roteamento:** O processo de encontrar uma rota para os pacotes ou segmentos de pacotes até o equipamento destino.
- Switches:** Termo em inglês para designar equipamentos comutadores. *Switch* é um equipamento, usado nas redes locais e de longa distância, que realiza comutação entre as suas portas *ethernet*.
- TCP/IP:** Os dois protocolos mais conhecidos dentre os protocolos da Internet. O TCP corresponde a camada 4 (camada de transporte) do modelo OSI e provê transmissão confiável de dados. O IP corresponde a camada 3 (camada de rede) do modelo OSI e provê serviço de datagrama sem conexão.
- Transformada de Fourier:** Técnica usada para avaliar a importância dos ciclos de várias frequências em um padrão de séries temporais.
- WAN:** (Wide Area Network). É uma rede que abrange uma grande área geográfica.
- Working party:** Termo utilizado para definir grupos de trabalho de organizações que fazem padronizações e recomendações. Também referenciado como *working group*.